

ABC BLADET

NUMMER 4, 1990

ABC-KLUBBENS MEDLEMSBLAD FÖR BLANDAD INFORMATION TILL BÅDE NYTTA OCH NÖJE

```
*****
**
**                               **
**           Välkommen till      **
**   ABC-Klubbens Monitor ver 3.01  **
**           Stockholm           **
**                               **
**   Internet:   abc.se ( <mn>Émsg.abc.se ) **
**   Fidonetnod: 2:201/235.0      **
**   Fax:        08-80 15 22      **
**   Kansli:     08-80 17 25      **
**               säkrast tisdagskvällar  **
**                               **
*****
```

Erfarenheter av PCBACKUP

PCSHELL erfarenheter

Att ta tillbaka enstaka filer från backup med PCBACKUP har alltid varit knöligt. PCBACKUP förutsätter att du vill ta tillbaka alla filer och inte bara några enstaka. Jag har alltid haft problem med detta, men idag fick jag tillgripa sista utvägen för att klara det dvs titta i manualen. Jag vill här vidarbefordra mina erfarenheter.

Jag har en Tranfor AT (CAF AT) med en monokrom Hercules skärm. Första problemet är att när jag markerar filer med PCBACKUP så blir det mycket lite skillnad i intensitet på markerade filer och på andra. Det är nästan omöjligt att se. Jag letade efter switchar till DOS kommandot som startar PCSHELL, men hittade inga sådan i manualen. Gjorde då PCBACKUP /X i hopp om att X skulle vara en illegal switch och visa mig vilka som fanns. Ja, det stämde. Man kunde visst också skriva PCBACKUP /? för att få informationen.

Så här såg det ut:

PCBACKUP 6.0 (c)1988,1989,1990 Central Point Software. Apr 13 1990 16:28:14

Options are:

```
/BW      -- Suppress colors
/LCD     -- LCD display screen
/DOB     -- Use Deluxe Option Board for Format
/NO      -- Don't Use Overlapped I/O
/LE      -- Left Handed Mouse (swap left and right)
/PS2     -- Use this option if mouse disappears on PS/2s
/R       -- Automatically start restore mode
/?       -- Help (this display)
fname    -- Automatically start backup with this setup
d:       -- specify a start up drive letter to backup
/ADDR=base-i-d -- specify I/O addresses for tape drive
           -- where base=hex base address, i=irq, d=dma channel
```

Jag hade tidigare provat /BW för det kom jag ihåg, men det gjorde ingen verkan. Provade då /LCD och så, då löste det sig. Alla markerade filer skrevs så ut med understrykning och även om det såg fult så kunde jag nu enkelt se vilka filer och bibliotek som var markerade.

När detta var löst var det enkelt att följa manualen och läsa tillbaka några filer i ett biblioteket. Hur man gör framgår av backup-manualen från sid 74 och framåt. Jag redogör här i korthet för det viktigaste stegen.

När man har startat PCBACKUP skall man gå till "restore To entry" och välja den drive som man vill lägga tillbaka filerna på. Därefter väljer man "Chose directorie" och väljer sedan bland de backuper som man har gjort och vars register är lagrade på hårddisken.

Därefter har PCBACKUP markerat alla filer och nu är det viktigt att man går till root-volymen och avmarkerar alla filer

genom att trycka på musen eller på enter-tangenten!

Nu skall du markera det bibliotek som du vill kopiera. Observera att PCBACKUP automatiskt markerar även underliggande bibliotek så du får ta och avmarkera dom. Har du markerat ett bibliotek så har också PCBACKUP markerat alla filer i det biblioteket vilket väl inte är vad man vill. Enklarest avmarkerar du det biblioteket också och går in på filnamnen och markerar de som du vill lägga tillbaka. Det går så klart och göra tvärt om också dvs att gå till filerna och avmarkera de filer som man inte vill återställa.

Efter detta är det bara att välja "Start Restore" och sedan sätta in den diskett som PCBACKUP kräver.

Börje Jansson har i MSG meddelat att man kan få konstiga i fel om man använder APPEND när man kör PCBACKUP. Felen uppträder när man läser tillbaka filer och om man hade bibliotek med samma namn som är append-ade till varandra. Jag kan inte beskriva felen i detalj, men det är tillräckligt för att varna er för att använda APPEND i samband med PCBACKUP. Överhuvud taget så kan APPEND ställa till problem även i andra program.

<1789>

Bo Kullmar

LIST80

Bakgrund

- Hur är det nu, du har väl en ABC-dator?
- Javisst, en gammal 802 med skönskrivare.
- Skulle inte du kunna hjälpa mig att skriva ut några program som jag gjorde på en kurs en gång?
- Vad är det för program då?
- Vi hade ABC80 på kursen, och programmen är på kassett.
- Jaha, och hur gjorde du när du sparade dem?
- Ja, jag skrev SAVE och namnet som man brukar.
- Då är programmen alltså semikompilerade.
- ??

Problemet löstes så småningom genom att jag laddade ett program i sänder i en ABC80, bytte kassett och sparade på nytt med LIST. Från den nya kassetten kunde jag sedan hämta programmen till min ABC802, spara dem på diskett och lista dem på papper.

Skulle man kunna lista ABC80:s internkod direkt? Jag letade, men kunde inte hitta något program för det ändamålet. Däremot hittade jag ett program för ABC80, LISTAREN.BAS, som skrivits av Leopold Lundström <2694>. Programmet listade .BAC-filer genom att anropa BASIC-tolken.

LIST80

Programmet består av två filer. Huvudfilen, LIST80.BAS, är en BASIC2-kodning av de viktigaste delarna i Listaren. Datafilen LIST80.DAT innehåller maskinkod - ett utdrag ur listrutinen i ABC80 (checksumma 9913) och ett gränssnitt till huvudprogrammet.

Programmet startas med:

```
RUN LIST80
```

och frågar då efter infil och utfil. Om man besvarar frågan om utfil med Return

så listas filen på skärmen. Filtyp kan utelämnas, och sätts i så fall automatiskt till .BAC (respektive .BAS om utfil har angivits). Programmet kan dock även använda enheter där filtyp inte tillåts, t ex MEM: och PR:. Genom att bara ange CAS: som infil är det också möjligt att söka på kassettband.

Jag har medvetet undvikit att använda parametrar i startkommandot. Det finns en viss risk för förväxling, vilket skulle kunna leda till att filer raderades av misstag.

För ABC802 gäller att listning kan göras direkt från CAS: till MEM: trots att båda enheterna normalt använder samma buffert. Om datorn inte har någon diskettstation ansluten (alltså inget styrkort) måste dock argumentet i rad 510 ändras från 5 till 9 för att listning till och från MEM: skall vara möjlig.

Om man vill ha en korsreferens över ingående variabler så kan den delen av programmet Listaren användas separat. Det finns troligen också andra program för korsreferens som går att köra på datorer i 800-serien. Tänk dock på att det måste vara avsett för ABC80:s programformat.

Begränsningar

Den nuvarande versionen kan tyvärr inte lista "smartladdad" maskinkod. Det går inte heller att lista komprimerade filer. De vanliga .BAS och .BAC-filer som jag har provat går däremot utmärkt. Programmet har testats på ABC802 + ABC821 + ABC834.

<6057>

Jan-Olof Svensson

Medlemsorgan för ABC-klubben
Box 14143
161 14 Bromma
ISSN 0349-3652

Redaktör, ansvarig utgivare: Ulf Sjöstrand
I redaktionen: Claes Schibler och Sven Wickberg

ABC-klubbens postgiron:

Medlemsavgifter: 15 33 36-3

Publikationer: 62 93 00-5

Q-Zentralen: 43 51 74-8

Bankgiro: 216-25 43

Telefoner:

08-80 17 25 "Prattefon" till klubblokalen och kansliet. Kansliet träffas säkrast tisdagskvällar.

08-80 15 22 Telefax och telefonsvarare.

Monitorer:

08-80 64 40 Gruppnummer för allt upp till 2400.

08-80 64 44-46 Direktlinje upp till 2400 MNP 5.

08-80 64 47 Direktlinje upp till 2400 MNP 5.

08-80 15 23 Testnummer nya monitorn, max 2400.

ABC-Öst

013-11 49 30 Gruppnummer för 31 och 32.

013-11 49 31 Direktnummer allt upp till 1200.

013-11 49 32 Direktnummer 300 och 1200/75.

Tidningen ansvarar ej för att införda programlistningar är korrekta.

Upphovsrätt gäller för införda program om inget annat anges.

I tidningen uttalade åsikter står författarens räkning och är endast där så anges uttryck för ABC-klubbens mening.

Tryck: Svenskt Tryck AB Upplaga 2 200 ex.

Lämnad till tryck 10 december 1990

NUMMER 4, 1990

INNEHÅLL

Erfarenheter av PCBACKUP av Bo Kullmar	Oms 2
LIST 80 av Jan-Olov Svensson	
Redaktionssidan	1
Ledare av Ulf Sjöstrand	2
Ordreda, ordbehandling på ABC	3
av Karsten Husberg	
Årsmöteskallelse	
Den nya MONITORN av Bo Kullmar	4
Grunderna för kommunikation av Bo Kullmar	9
PC-diskett 12 av Bo Kullmar	10
PC-diskett 13	12
HACK III av Hugo Wikström	
Kallelse till HACK IV	
Radannons	
Minnen av Bo Michaelsson	13
Ett svårt problem av Sven Wickberg	18
DÖBELN	
av Sven Wickberg o Anders Umegård	20
Sextionio frågor och svar om	22
C-programmering	
SNUS och SWIPNET av Bo Kullmar	29
Modem för 9600 bps av Bo Kullmar	
The GNU:s Bulletin, June 1990	30
The ANSI Standard	37
Lokalsamtal till USA av Bo Kullmar	41
Videotex av Bo Kullmar	
Utdrag ur MSG av Lars Gjöring	42
PSPRÅK	
MONITOR	
Datakommunikation	
MS-DOS	
PCProg	
PChård	
Pascal	
Radannonser	48
Några ord om EDITOR av Claes Schibler	
Läsövning av finaste kanslisvenska	

Medlemsavgifter för 1990

Seniorer 190 SEK och juniorer 160 SEK

Junior räknas man t o m det kalenderår man fyller 18 år. Ange därför personnummer när Du betalar medlemsavgiften.

Medlemsskapet är personligt och avser fysisk person. Medlemsskapet räknas per kalenderår och Du får automatiskt det löpande årets förmåner retroaktivt om Du inte markerat annat årtal på talongen när medlemsavgiften betalas in.

ABC-klubbens styrelse för 1990

(enligt årsmötet 1990-03-17)

Ordförande: Bo Kullmar

Vice ordförande: Jaan Tombach

Ledamöter: Bengt Almén

Kjell Brealt

Jan Liebe-Harkort

Martin Lundberg

Tom Sjöberg

Ulf Sjöstrand

Suppleanter: Ulf Hedlund

Karsten Husberg

Per Sten

Ledare

Jubileumsår. ABC-klubben tio år. Har det då varit någon festyra under det gångna året? Tja, sådär. Och har vi något att festa?

Man måste dock först konstatera att ABC-konceptet var världsberömt i Sverige och hade nog inte någon chans att riktigt lyfta över, nåja, Skandinavien då. Man kan också konstatera att hade ABC-klubbens medlemmar som grävde djupt i hårdvaran inte haft någon möjlighet att sprida denna sin kunskap hade ABC-konceptet inte varit kvar så länge som det i alla fall har varit.

Man kan peka på den förändrade nischen för 800-serien som gjorde att klubbens medlemmar inte längre i samma stora utsträckning var "hackers" som gjorde sin egna program, utan mer fick inriktningen att man skulle ha "köpeprogram" som man på sin höjd anpassade till sin verklighet. Denna förändring var å andra sidan en nödvändighet för kontorssidan när IBM och de övriga kom med sina "de-facto-standarder" och som förutsatte en massupplaga ett program för att man skulle kunna utbilda människor att sköta "smör-och-bröd-rutinerna".

Man kan konstatera att ABC-klubben har klarat inlämandet av PC-världen relativt smärtfritt. Klubben fortsätter sin "spjut-spets-teknik" genom att i sina led ha medlemmar som bygger stora komplex av program.

Iden finns kvar inom klubben att den enskilde medlemmen skall kunna följa med vad som händer inom en sfär som intresserar allt fler.

Som ett lysande exempel på detta har den nya monitorn öppnats. Själva övergången skildras i detta nummer av Bo Kullmar som i sin artikel beskriver några av de kommandon som redan finns inlagda. Det förutskickas också en del nya möjligheter. Det är väl inte någon större hemlighet att ett sånt här system "aldrig" skall bli klart.

Vad avser MSG-utdragen denna gång har vi försökt att avsluta utdragen från den "gamla" monitorn och vi börjar alltså om med numreringen och det kanske blir några nya mötes-namn också. Lars G har beskrivit sin vända över de val som måste göras. Många av de första inläggen i den nya monitorn handlade mycket om svårigheter att få det

hela att fungera. Inläggen var nog mycket intressanta för den/de som skrev men var bäst glömda av en större krets när svårigheterna var bemästrade.

Förhoppningen att det kan vara intressant att veta hur ABC-disketterna "kommer till" har fått Bo K att beskriva det ofta oglamörösa arbetet (=slitet) bakom kulisserna.

Bo Michaelsson har skrivit en nästan lyrisk hyllning till ABC-klubbens verksamhet och det stöd som han kände. Men det känns så riktigt för oss som haft förmånen att få vara med när det begavs sig. Hur var och är det ute i landet? Bo skildrar ju det från en stockholmsk utsiktspunkt.

Sven W redovisar resultatet från förra numrets tävling och startar genast en ny. Här får man ju verkligen tillfälle att se olika sätt att angripa ett och samma problem.

Det får mig att osökt komma in på de olika "språken". Den här gången har vi tagit med en del texter om C-programmering samt om dialekter av Unix. Även ett discussionsunderlag om ANSI har tagits med.

Sedan har några artiklar kommit med som handlar om speciella program.

Sedan till något som man skulle kunna kalla ABC-bladets inre arbete. Hittills har vi framställt bladet "manuellt", med en gammal Diablo 1650 som trotjänare. Nu skall vi försöka att gå över till modern teknik, sk desk top. Som försäkring kommer vi att ha kvar Diablon, en maskin som vi kan lura att göra nästan vad som helst. MSG-utdragen kanske blir litet större :-), tidplanen skall vi försöka hålla bättre.

Mot slutet vill jag uppmana alla och envar att ransaka sitt minne om hur kom datorn in i era liv. Och skriv ned det!

Med en sådan här genomgång av de aktiviteter som klubben har på gång och inte minst för vad ABC-klubben står för, tycker jag att vi verkligen har någonting att fira. Varje dag.

<1208>

Ulf Sjöstrand

ORDREDA

ordbehandling på ABC

Årsmötes-

kallelse

ORD-reda för ABC800, 802 och 806 distribueras nu av ABC-klubben.

Programmet gjordes i början på 80-talet av Bengt Sagnell som hade samlat på sig en rad synpunkter på de ordbehandlingsprogram som då användes. Programmet gjordes i första hand för dem som författade texter, PM, rapporter och böcker direkt i datorn. De övergripande idéerna var följande:

- Ett snabbt system som också reagerar direkt på alla kommandon, även vid arbete med stora dokument.
- Kommandona kan ges i olika nivåer, varav den enklaste inte stör skribentens koncentration.
- Formateringsmöjligheter för laserutskrift och för reproducering med enklare tryckmetoder.

Administrativa funktioner

Vid produktion av många och långa dokument uppstår nästan omedelbart problem med att hålla reda på dokumentens innehåll, revisionsnummer och lagringsplats på disketter eller hårddisk. Systemet försågs därför med ett textbibliotek som automatiskt förtecknar och uppdaterar innehållet på 32 disketter/-bibliotek. Varje diskett rymmer 18 dokument a' cirka 100 A4-sidor. Varje dokument kan dessutom förses med en 45 tecken lång beskrivning och två datum.

Funktionstangenter

Som funktionstangenter på ABC80 användes den skiftade övre raden på tangentbordet, dvs tecknen ! till >. Detta visade sig vara en så praktisk och lättanvänd metod att den behölls i ABC800 versionen, parallellt med de "äka" funktionstangenterna.

Menyer

Ord-reda använder sig med ett par undantag av horisontella menyer. Från alla menyer kan alla funktioner anropas, utom kalender och skrivmaskinsfunktionerna. Ett defaultvärde står i markören och nås med return.

Formateringsystem

På skärmen syns texten med formateringskoder. De består av en punkt följt av vad den gör. Det finns 38 olika punktkoder. T ex ".vänster 10" flyttar vänstermarginalen tio steg till höger och ".vänster -10" flyttar tillbaka den. Dessutom finns 11 teckenkoder som fungerar på enstaka tecken såsom understrykning och specialkoder till printern.

Vissa koder är ovanliga även i nyare ordbehandlingsprogram t ex ".radav X/Y" som ändrar radavståndet i proportionen X/Y, eller ".rad -10" som backar papperet tio rader, om printern klarar det. Koderna kan skrivas med små eller stora bokstäver på svenska eller engelska.

Texthandbok

Standardtexter eller ofta återkommande fraser kan lagras i textmoduler på programskivan och sedan kopplas in i den aktuella texten.

Dokumentstruktur

För att underlätta förflyttning i texten och för att undvika stora förluster vid strömvabrott, delas texten in i textavsnitt. De lagras som strängar på 4000 tecken, vilka ABC-maskinerna hanterar mycket snabbt. Varje text kan bestå av upp till 65 textavsnitt, som i en innehållsförteckning visas tillsammans med första raden i varje avsnitt.

Sidindelning

Problemet med indelning i sidor är löst med hjälp av en särskild rutin som föreslår lämpligt läge för sidslut. Operatören väljer sedan själv vid vilken rad brytningen skall ske, detta för att eliminera det inte helt ovanliga felet att en rubrik kommer längst ner på en sida och dess understrykning på nästa.

Skrivmaskin

I de fall man inte behöver en fullständig ordbehandlare finns motsvarigheten till en elektrisk skrivmaskin, som kan använda alla initierade typsnitt.

Kalender

Systemet rymmer även en kalenderfunktion där man kan göra anteckningar för varje dag. Anteckningarna kan sedan skrivas ut för en månad i taget eller användas i texter.

Med en 806:a och en snabb kontroller överträffar ORD-reda många av PC:ns populära ordbehandlingsprogram.

Programmet beställs genom att sätta in 100 kronor på

postgiro 62 93 00-5 eller
bankgiro 216-25 43 och ange:

ORD-reda, format, namn och medlemsnummer.

Karsten Husberg <6574>

Kallelse

till ABC-dagen med årsmöte 1991 i ABC-klubben

**Tid: Lördagen den 16 mars 1991
med början kl 13.00**

**Plats: Brommasalen, kommunalhuset i Alvik
Gustavslundsvägen 168, 161 33 Bromma
T-banestation: ALVIK**

Förslag till dagordning för
ABC-klubbens årsmötesförhandlingar 1991

1. Val av mötesordförande
2. Val av mötessekreterare
3. Val av två justeringsmän, tillika röst-räknare att jämte mötesordförande justera årsmötesprotokollet
4. Fråga om mötet är behörigen utlyst
5. Fastställande av dagordning
6. Styrelsens redovisningshandlingar
7. Föredragning och godkännande av revisionsberättelse
8. Fråga om ansvarsfrihet för styrelsens medlemmar
9. Fastställande av balansräkning
10. Beslut med anledning av vinst eller förlust enligt balansräkningen
11. Fastställande av budget och medlemsavgift
12. Val av ordförande och vice ordförande samt övriga styrelseledamöter och suppleanter för ett år
13. Val av revisorer och en revisorssuppleant för ett år
14. Val av valberedning om minst två personer
15. Behandling av ärenden som styrelsen vill förelägga årsmötet
16. Behandling av motioner som medlemmarna inkommit med senast sex dagar före mötet
17. Övriga frågor

Eventuella frågor skall ha inkommit senast fredagen den 8 mars 1991 till

ABC-klubben
Vidängsvägen 1
161 33 Bromma

ABC-klubben/Styrelsen

Den nya MONITORN

Den nya monitorn

Den 14 oktober 1990 hade vårt gamla monitorsystem varit i drift i sex år. Vi valde detta datum för att öppna det nya systemet fullt ut. Vi skaffade ett nytt system därför att den gamla hårdvaran inte längre var tillräckligt snabb. Att det nya systemet är mycket snabbare än det gamla kan alla märka som provar det. Det gäller såväl konferenssystemet som filöverföringen.

Det nya systemet har varit under test-drift på 08-801522 och på några linjer på gruppnumret sedan någon dag under sommaren. Nya monitorn blev snabbt populär trots att programvaran inte var helt färdig. Många ville att vi skulle lägga ner det gamla systemet snabbt och köra det nya fullt ut, trots att alla funktioner inte var helt färdiga.

Vi beslöt att övergången till det nya systemet skulle ske den 14 oktober.

Det gamla ABC802-baserade systemet startade den 14 oktober 1984 och har sedan dess gått oavbrutet med mycket få driftstopp. När systemet startade fanns det bara två linjer med två 300 bps modem. Systemet baserades på ett CAT-NET nätverk och en Mica 40 MB hårddisk.

Några år senare fick vi ett LUX-NET och flera 802:or gratis från Luxor och till slut fanns det 7 st 802:or anslutna till systemet och 7 modem. Mot slutet klarade alla modem alla hastigheter upp till 2400 bps dvs 300, 1200/75, 1200 och 2400 bps.

Öppningen

Den 14 oktober 1990 öppnades det nya systemet fullt ut. Det innebär att alla modem på gruppnumret 08-806440 kopplades om till den nya DIAB-datorn. Därutöver finns också 08-801523 inkopplad till nya systemet.

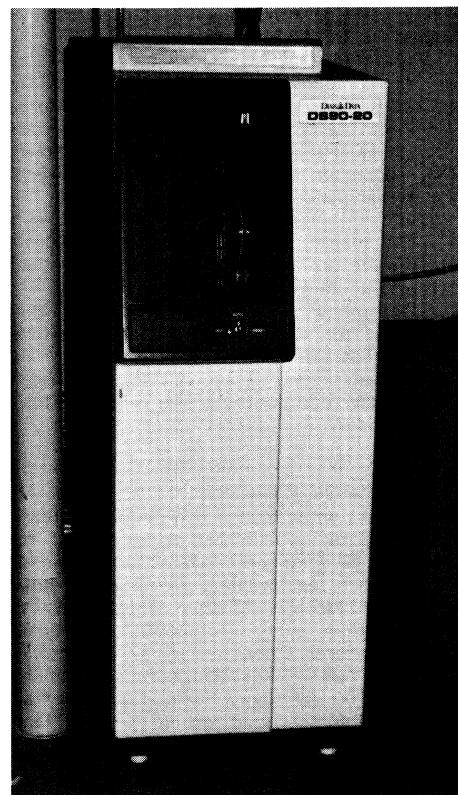
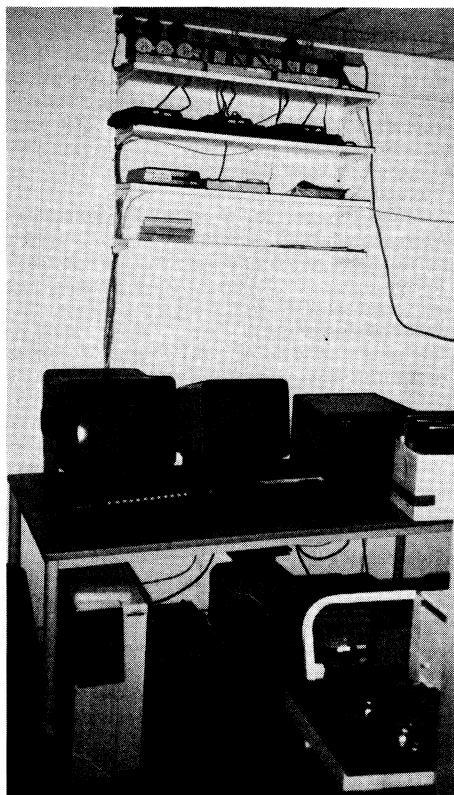
Benny Löfgren och Kjell Brealt höll ställningarna i klubblokalen medan jag provade lite hemmifrån.

I samband med öppningen flyttades alla modem till en ny modemhylla och alla kablar sattes upp i prydlig ordning. Benny Löfgren bjöd också in medlemmar strax före den 14:e genom att annonsera öppningen i MSG. Det kom en del medlemmar för att bevittna denna för ABC-klubben historiska händelse.

Det nya systemet

Så här hälsar systemet dig välkommen numera. Först kommer den vanliga Tryck RETURN. Har du ett MNP 5-modem så kan det hända att du missar texten Tryck RETURN om du ringer upp mot ett av våra modem som inte har MNP. Vi har dock lagt in en liten fördröjningar så att du inte skall missa någon text.

Det står versionsnummer 3.01 nedan och Benny har alltså redan hunnit uppdatera från 3.00 som var versionen på öppningsdagen till version 3.01. Troligen är det ett högre versionsnummer när du läser detta.

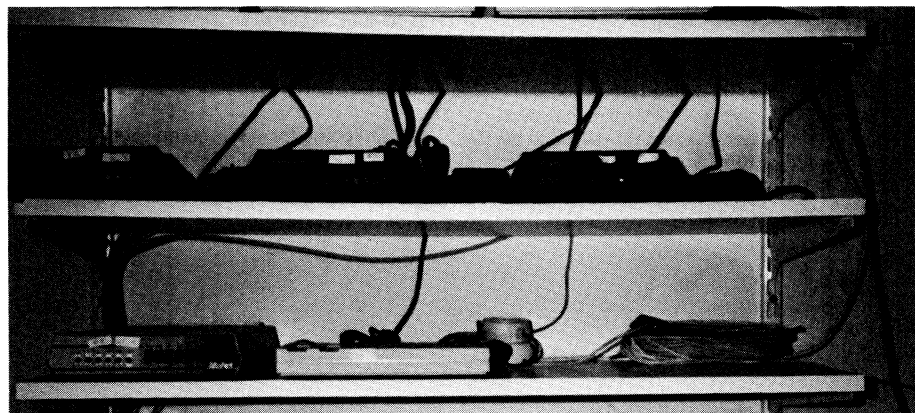


```
*****
**                               **
**           Välkommen till      **
**      ABC-Klubbens Monitor ver 3.01      **
**           Stockholm          **
**                               **
** Internet:  abc.se ( <mn>Emsg.abc.se ) **
** Fidonetnod: 2:201/235.0          **
** Fax:       08-80 15 22          **
** Kansli:    08-80 17 25          **
**           säkrast tisdagskvällar      **
**                               **
*****
```

1. Detta är den nya monitorn med DS90-20:n med dess skärm och modemen på hyllan

2. DS90-20 från Diab Data

3. Modemhyllan



Tryck CTRL-C för att avbryta inloggningen!

Medlemsnamn/nummer:

Numera kan du ange medlemsnummer eller namn när du loggar in. Observera att du inte kan ange både namn och medlemsnummer!

Som vanligt får du se nyhetstexten om den är uppdaterad sedan du körde sist. Är du ny användare får du dessutom se en text de första gångerna som du kör.

Tyvärr blinkar DS90:n lite med DTR. Det sker ungefär en gång per minut. Detta gör att du i olyckligaste fallet kan råka ut för att klubbens modem kopplar ner dig under själv uppringningen. Vi undersöker nu om det går att göra något åt detta.

Har du tappat bort ditt lösenord så ta kontakt med kansliet så hjälper vi dig. Se också nedan under rubriken password.

Kommandon i systemet

bye/quit/logout

Dessa tre kommandon avslutar din körning. De är alla synonymer.

help

Hjälpkommandot ger dig hjälptexter för alla kommandon. Använd detta kommando om du är osäker på ett kommando. Här beskrivs också alla switchar till kommandona. Allmänt gäller det att du kan ange switchar med komma som i gamla monitorn, ex: "dir,a" eller med ett minus före som: "dir -a".

Enbart "help" ger information om vilka hjälpfiler som finns och vilka extra hjälpfiler som finns och som inte är anknytna till ett kommando. Observera att du fn måste skriva ut hela parametern på det som du vill ha hjälpinformation om. Du kan alltså inte skriva bara "help view" utan måste skriva "help viewarch".

?

Ett frågetecken ger en kort lista med alla kommandon.

setpar

Setpar är ett av det viktigaste kommandona i det nya systemet. Det fanns inte alls i gamla monitorn. Några kommandon fanns dock i det gamla systemet som nu ingår i setpar som terminalhöjd och terminallängd.

När du skriver setpar kan du få upp en text liknande denna:

```
Medlemsnummer:      10
-----
Namn:                Bosse L Test
Telefon:
Adress:

Privilegier:         00008001
Personliga flaggor:  00000041
Terminal:            Typ vt100, storlek 24 x 80 tecken.
Senaste login:       900922 16.18, totalt 10 inloggningar.
Total tid:           1 timme, 34 minuter, 15 sekunder.
Filöverföring via:   ascii.
Filöverföring:       0 bytes hämtade, 0 bytes skickade.
```

```
1. Adressrad 1,      2. Adressrad 2,   3. Adressrad 3,
4. Telefonnummer,    5. Terminaltyp,   6. Terminalbredd,
7. Terminalhöjd,    8. Personliga flaggor,
9. Filöverföringsprotokoll,
s. spara ändringar, q. avsluta utan ändring.
```

Vad vill du göra/ändra på?

Svarar du 8 så får du upp följande. Inom parentes anges standardvärdet.

Tecken ekas från monitorn (J)?

Detta kallas ibland för full duplex eller echo. Det skall vara J här normalt så det är bara att trycka på return.

Hemliga adressuppgifter (N)?

Vill du hindra att andra medlemmar ser din adress här så anger du J. Observera dock att detta adressregister fn inte på något sätt är samordnat med medlemsregistret. Dessa namnuppgifter får du själv ändra om de blir fel.

Hemligt telefonnummer (N)?

Här kan du göra så att andra inte får se ditt telefonnummer.

Menysystem (ej implementerat) (N)?

Det är meningen att vi skall införa ett menysystem för monitorn som är likt vad som brukar förekomma i vanliga PC/BBS-system. Idag är systemet enbart kommando-styrt för att vara likt det gamla systemet.

Gör WHO vid inloggning (N)?

Du kan få en automatiskt "who" utförd vid inloggningen om du anger J här.

Visa när någon loggar in eller ut (N)?

Svarar du J är det tänkt att du skall se när någon loggar in eller ut när du befinner dig i monitorn. Detta är inte implementerat ännu.

Tryck RETURN-prompt (J)?

Vill du att systemet inte skall fråga efter tryck RETURN när du t ex gör type på en fil så anger du J här.

Visa aktuellt bibliotek i kommandoprompten (N)?

Ange J här om du vill se aktuellt bibliotek i prompten. Detta är att rekommendera.

Åter till setpar-menyn

När du har gått igenom alla frågorna kommer den aktuella informationen upp igen och du kan välja mellan att ändra något annat eller att gå ur genom att spara (kommando s) eller genom att gå ur utan att spara (kommando q).

Du kan inte ändra ditt namn här utan det måste systemansvariga göra. Skriv ett brev till mötet "Kansli" i MSG om du vill att ditt namn skall ändras. Vi förutsätter också att du då vill ändra ditt namn i medlemsregistret också.

Anger du 9 för att välja filöverföringsprotokoll får du upp följande meny med valmöjlighet:

1. ASCII, 2. Kermit, 3. Xmodem, 4. Ymodem, 5. Zmodem
Välj protokoll (l):

ASCII är standard och innebär bara att filen dumpas in eller ut enkelt utan flödeststyrning. Du väljer här normalt Kermit eller något av xyz-protokollen.

Binär (Image) eller text (I/T) (T):

Detta är en fråga om du vill ha binär eller textmode som standard vid filöverföringen. Binär innebär att filen i fråga ej förändras medan textfil innebär att filen lagras i vårt system som en Unix-textfil vilket skiljer sig lite från t ex PC-DOS-/MS-DOS-textfiler vad gäller radslut.

Slutligen kan du svara s för att spara dina ändringar och då får du följande text:

Uppdatering i databasen sker vid utloggning

Det innebär att skulle monitorprogramvaran råka ut för systemkrash under din körning så uppdateras inte systemet med dina nu inmatade uppgifter.

talk

Talk-kommandot är inte klart ännu. Det är avsett för att du on-line skall kunna tala med en annan medlem.

msg

Msg är kommandot för att starta MSG-systemet, dvs brev och konferenssystemet MSG.

MSG är sig likt, men en del förbättringar har införts. Nu finns det plats för drygt 100 möten och mötesnamnen kan vara upp till 14 tecken långa. Texterna kan också vara längre. Storleken är numer satt till 499 rader. Skickar du en textfil från din egen dator genom att bara dumpa ut den så kan du använda kommandot "InoWrap" först på en rad för att undvika radbrytning. Är det någon som har ett bra svenskt namn för detta?

Att nå stora världen

Peter Fässberg har nu tagit över underhållet av MSG. Peter har nu infört möjligheterna att skriva brev till yttervärlden genom att skriva brev till nr 1. Du kan på detta sätt nå personer inom Internet dvs Unix-världen och Fidonet.

Du skriver ett brev till nr 1 och sedan frågar MSG efter en adress och du skriver en vanlig unix-adress där. Om jag skriver brev till mig själv skriver jag brev till "bkEkullmar.se" och min avsändaradress blir då "l789Emsg.abc.se" i MSG. Besvarar jag brevet hemma kommer svaret att skickas tillbaka till mig i MSG.

Skriver du brev till Fidonet skriver du också brev till nummer 1 och anger adressen på Unix-vis. Ev kommer detta att ändras med hänsyn till att adressen kan bli för lång. Ändringen kan gå ut på att du i stället skall skriva brev till nr 2 och då kan du ange adressen på ett mera Fidonet-lik sätt.

Idag skriver du min Fidonet adress så här:

"Bo_KullmarEf235.n201.z2.fidonet"

Skall du skriva ett brev till en point, t ex point nr 1, skriver du "pl.f235.n201.fidonet". Min adress här från Fidonet är alltså "Bo Kullmar" 2:201/235. Det är denna senare varianten av adress som skall användas om Peter gör en speciell hantering av Fidonet. Den adressen blir kortare än den rena Unix-adressen och kan då bättre få plats i MSG:s interna register.

Ändrat MSG-kommandon

Kommandot "vilka" i MSG har ändrats lite och ser nu ut så här:

Läsa (inlägg) 2. - vi
Vilka (är närvarande)

tty13: Göran Edqvist är inloggad i monitorn
tty14: Paul Pries är inloggad i monitorn
tty15: Bosse L Test <10> befinner sig i Medforum
tty16: Benny Löfgren SYSOP <2615> är inloggad

Gamla MSG-texter

Samtliga gamla msg-texter finns i /text-/old_msg-biblioteket. De som tidigare har rensats ut ur gamla MSG finns packade i PC-ziparkiv. De som fanns i gamla MSG när det stängdes finns i textfiler i samma bibliotek. Är det någon som vill ha de äldre filerna och som inte kan packa upp dem så skriv till "Kansliet" så skall vi försöka hjälpa till.

Nytt MSG

Planerna är att vi senare skall skaffa ett nytt och bättre konferenssystem än MSG. MSG är därför bara en tillfällig lösning. Ändringarna i MSG kommer därför inte att bli så omfattande utan bara de som är nödvändiga. MSG är porterat från gamla ABC via BASIC på DS90 och BasC. BasC är Basic to C, vilket gör att det går att kompilera som C-program. BasC är dock ingen generell konverterare för DIAB BASIC utan programmen som konverteras kan enbart köras på DS90.

När allt annat är klart skall vi se över de alternativ som finns. Antingen kan vi köpa ett färdigt konferenssystem för Unix eller så kan vi skriva något själva. Bland färdiga system finns Ulf Hedlunds Unix TCL, KOMunitys PortaCOM och QZ:s SuperKOM.

password

Password är kommandot för att ändra lösenord. Vi fick tilldela nya lösenord som är åtta tecken långa för det nya systemet eftersom lösenordsalgoritmen har förändrats. På grund av envägsenkrypteringen kan vi inte överföra de gamla lösenorden. Lösenorden skrevs på adressetiketten till ABC-bladet nr 1 vilket tyvärr distribuerades i kuvert varför lösenordet i regel försvann. Har du inget lösenord så ta kontakt med kansliet på 08-80 17 25 eller genom brev så skall vi hjälpa dig.

Tar du kontakt med oss via brev eller fax för att få ett nytt lösenord så är det enklast att du anger ett nytt lösenord som du vill ha. Vi lägger så in det så snart vi kan och på så sätt vinner vi både portokostnaden och tid. Du kan själv kolla om du kommer in på det nya lösenordet.

När du ger kommandot "password" frågar systemet först efter ditt gamla lösenord och du måste ange det rätt för att kunna ge ett nytt. Du måste därefter upprepa det nya lösenordet en gång till för säkerhets skull.

time

Kommandot visar dagens datum och tid, hur länge du har varit inloggad, hur lång tid du har på dig innan du blir utloggad på grund av maxtid, hur många gånger du har loggat in och total tid för dina körningar. Så här kan det se ut:

upload>time

Idag är det lördag, och klockan är 20.03.19.

Dagens datum är 1990-11-10 (dag 314)

Du har varit inloggad i 17 sekunder.

Du blir automatiskt utloggad om 1 timme, 59 minuter, 44 sekunder.

Antal körningar: 11

Total tid: 1 timme, 36 minuter, 59 sekunder.

status

Kommandot ger status för en medlem. Allmänt gäller att denna typ av kommando kan ta medlemsnummer eller namn som parameter. Informationen som du får är samma som setpar-kommandot ger.

who

Kommandot who visar vilka som är inloggade i monitorn. Det finns inga switchar eller parametrar till kommandot. Så här kan det se ut när du ger kommandot:

Mnr	Login	Linje	Namn	>veri bo k
7381	901023 07.37	/dev/tty13	Göran Edqvist	Bo Klang <739>
5322	901023 09.19	/dev/tty14	Paul Pries	Bo Kullmar, SYSOP <1789>
10	901023 09.20	/dev/tty15	Bosse L Test	Bo Knutsson <7786>
2615	901023 09.15	/dev/tty16	Benny Löfgren	Bo Korsell <7885>

Här följer en lista på ttylinjer och motsvarande modem och modemtyper som finns uppkopplade:

/dev/tty12	08-80 15 23	SRT 8333
/dev/tty13	08-80 64 41	SRT 8333
/dev/tty14	08-80 64 42	SRT 8333
/dev/tty15	08-80 64 43	SRT 8333
/dev/tty16	08-80 64 44	HeathCOM AX/2424c
/dev/tty17	08-80 64 45	HeathCOM AX/2424c
/dev/tty18	08-80 64 46	HeathCOM AX/2424c
/dev/tty19	08-80 64 47	Alfa-NET ANC 8X224

HeathCOM-modemen har MNP 5 och körs mot vårt system i 9600 bps. Alfa-NET-modemet har också MNP men det är en lägre nivå. Normalt bör du ringa gruppnumret, men om du har ett MNP 5-modem bör du ringa direkt till något av våra HeathCOM-modem. Alla modem klarar alla hastigheter upp till 2400 bps, dvs 300, 1200/75, 1200 och 2400. Ring inte på 08-80 15 23 ty på den linjen har vi satt kort maxtid.

Vi diskuterar att skaffa 9600 bps-modem och dessa kommer troligen att sättas på ett nytt gruppnummer eftersom två nya linjer på 806440 kostar lika mycket som två nya linjer på ett nytt gruppnummer. Vi har ännu inte bestämt vilka modem vi skall skaffa, bara att det blir minst ett HST-modem. Problemet med HST-modemen är att de ännu inte är godkända för inkoppling på televerkets nät. Importören Dennis Bergström arbetar med detta så frågan kan nog lösas sig. Om vi skall skaffa enbart HST/DS-modem som förutom HST-standarden även har V.32 och V.42bis är främst en prisfråga. Tecken tyder nu på att vi kan få rena 9600 bps-modem mycket billigt i början på nästa år. Det skulle tala för flera modem av den typen och bara ett HST/DS-modem.

loglist

Kommandot visar loglistan. Det är en tillfällig lösning på kommandot och enbart en enkel listning av en logfil. Du måste ange "loglist -r" och får då se en textfil som just nu börjar när vi öppnade systemet.

```
>loglist -r
901014:113330:pk00 :2615 :LOGIN, User=Benny Löfgren, SYSOP, logins=301
901014:113359:pk00 :2615 :LOGOUT, User=Benny Löfgren, SYSOP, time=29 s
901014:113909:pk00 :2615 :LOGIN, User=Benny Löfgren, SYSOP, logins=302
901014:114508:pk00 :2615 :LOGOUT, User=Benny Löfgren, SYSOP, time=359 s
```

news

Kommandot visar nyhetstexten som du annars får se automatiskt när den har blivit ändrad.

info

Kommandot visar en allmän infofil med information om systemet. Det är samma textfil som du alltid får se när du loggar in. Vi har kommit överens med DIAB när vi köpte DS90-20:en till ett förmånligt pris att tala om att maskinen är en DIAB-maskin och det gör vi på detta sätt.

verify

Kommandot används för att visa vem som har ett viss medlemsnummer eller för att visa vilka medlemmar som matchar en viss förkortning av för och/eller efternamn. Vissa

andra kommandon t ex status kan använda en unik förkortning av medlemmens namn som parameter.

Som parameter anger du antingen medlemsnummer eller en namnförkortning. Så här kan det se ut:

>ver 1789

Medlem <1789> är Bo Kullmar, SYSOP.



4. Den är inte stor...

listuser

Kommandot används för att lista medlemmar. -n skall ge en lista sorterad på efternamn och -r skall ge en lista sorterad baklänges. Dessa båda två funktioner är ännu inte färdiga.

>listuser -a

Ger en lista på alla medlemmar som någon gång har loggat in. Så här kan listan se ut:

Mnr	Senaste login	Antal	Total tid	Namn
3	900811 20.29	4	42:44	Per Testare
10	901023 09.20	10	1:36:42	Bosse L Test
22	901109 14.25	20	1:48:20	Olof Backing

>listuser -t

Ger en lista över hur mycket medlemmar har skickat in till monitorn och hämtat från monitorn i KBytes. Den grundläggande funktionen som lagrar uppgifter om detta är inte klar och därför finns det alltid noll här och på andra liknande ställen som skall innehålla samma uppgifter.

Mnr	Senaste login	K t. mon	K fr. mon	Namn
3	900811 20.29	0	0	Per Testare
10	901023 09.20	0	0	Bosse L Test
22	901109 14.25	0	0	Olof Backing
92	-	0	0	Jinge Flücht (demokonto)

>listuser -s

Detta ger alla som har beteckningen SYSOP. I nya systemet ges denna beteckning automatiskt av systemet till medlemmar som har extra behörighet. Därför finns det fler medlemmar som har detta än i det gamla systemet.

Mnr	Senaste login	Antal	Total tid	Namn
283	901110 09.21	52	16:26:05	Kjell Breal, SYSOP
441	901110 16.39	225	19:49:11	Peter Fälsberg, SYSOP
837	901109 16.54	61	21:47:43	Karl Lindström, SYSOP
1789	901110 19.47	453	87:42:53	Bo Kullmar, SYSOP
2615	901109 22.44	458	56:56:23	Benny Löfgren, SYSOP
4283	901110 17.08	72	15:52:24	Jaen Tombach, SYSOP
4401	901029 20.53	17	2:42:47	Martin Lundberg, SYSOP
5581	901110 10.02	81	13:27:37	Per Andersson, SYSOP
6366	901110 14.04	41	6:35:36	Per Sten, SYSOP
26150	900905 05.15	178	16:05:04	Bennys Testkonto, SYSOP
6366	901110 14.04	41	6:35:36	Per Sten, SYSOP
26150	900905 05.15	178	16:05:04	Bennys Testkonto, SYSOP

>listuser -c

Denna variant ger en sammanräkning av det totala antalet inloggningar. Det är också tänkt att kommandot skall räkna ut hur många aktiva medlemmar som finns i systemet, dvs vilka som har kört inom 30 respektive 7 dagar. Detta kommando innebär en total genomläsning av användarfilen och därför tar det lite tid varför texten "Vänta" kommer upp. Så här kan det se ut:

>listuser -c

Vänta, beräkning pågår...
337 personer har någon gång loggat in.
Totalt 9902 inloggningar har gjorts.
Total onlinetid 1758 timmar, 11 minuter, 49 sekunder.

unix

Detta kommando är tänkt att ge tillgång till en begränsad unix så att de som inte har en egen c-kompilator kan prova på att kompilera ett litet program. Kommandot är ännu inte klart.

oldmon

Kommandot skall vara ett tillfälligt kommando för att nå det gamla systemet. Det är i första hand tänkt att användas mot gamla MSG. Kommandot är ännu inte klart och kommer när det införs bara att finnas en kortare tid och sedan tas bort.

Vi kommer dock att behålla det gamla systemet för att från detta kunna kopiera ut ABC-program, men det kommer inte att på sikt vara möjligt att nå det från nya systemet.

send/download

Kommandot används för att hämta en eller flera filer med ett valfritt filöverföringsprotokoll. Download är en synonym.

I gamla systemet fanns det två kommandon för filöverföring. Kermit och SendABC.

Det senare finns det inte stöd för i det nya systemet. SendABC var ett enkelt filöverföringsprotokoll som enbart kunde användas på ABC om man körde vissa terminalprogram. Idag finns det Kermit för både ABC80 och ABC800 så vi har valt att inte skriva ett program som kan överföra filer enligt den enkla FILTRANS-metoden.

Det blir enklast om du använder kommandot setpar för att ange det filöverföringsprotokoll som du normalt vill använda och om du vill ha text eller binärfiler som standard. Hämtar du filer i ett arkiv, t ex PC-filer som är zipade måste du ange binärmod.

Du kan välja mellan Kermit, Xmodem, Ymodem och Zmodem. Egentligen kan du välja Ascii också, men det är egentligen inte något filöverföringsprotokoll utan bara överföring av rådata utan flödeskontroll.

Även om du har valt ett annat filöverföringsprotokoll så kan du ange ett annat protokoll när du använder kommandot download eller send.

-k Kermit
-x Xmodem
-y Ymodem
-z Zmodem

-i Image mode/binär mode
-t Textfil mode

Använder du ABC bör du använda Kermit med hjälp av ett Kermitprogram till ABC80 eller ABC800. Det finns inte något program för överföring med hjälp av Xmodem, Ymodem eller Zmodem till ABC.

Använder du PC eller en annan modern dator bör du enbart använda Kermit om du har en snabb och modern Kermit som

MS-Kermit eller C-Kermit (på unix eller andra operativsystem). De flesta Kermit-versioner som finns i populära PC-program såsom Procomm och Telix är ganska dåliga och bör undvikas.

Anledningen till att Kermit gick långsamt på gamla systemet var att hårdvaran satte sina begränsningar vad gäller både blockstorlek och snabbheten i allmänhet.

Använder du program typ Procomm eller Telix bör du använda Zmodem eller Ymodem. Xmodem kan du använda också, men det förekommer i olika varianter och de kan vara svåra att få att gå bra. Procomm har inte Zmodem, men du kan använda Zmodem som externt protokoll.

Försöker du skicka upp filer med ascii så kommer filen att hamna i en enkel editor. En tområd gör att du kommer till kommandoläge i den editorn varför du inte får skicka tomrader. MS-Kermit och C-Kermit kan ställas in för att skicka en blankrad i stället för en tområd med "set transmit fill space" resp "set transmit fill 32".

För att hämta en fil ser du till att du finns i det bibliotek där filen finns. Kolla det med "dir -s". Sedan skriver du ditt downloadkommando med filens namn som parameter. Kom ihåg att du måste skriva filnamn med små bokstäver eftersom systemet skiljer på små och stora bokstäver.

Säg att du har Kermit som standardprotokoll och binärfiler som standard och vill hämta filen lsupdate.zip. Då skriver du så här:

download lsupdate.zip

Efter detta går du över till ditt eget Kermitprogram. Är det MS-Kermit så gör du ALT-X och skriver "r" samt trycker på return. Du kan föra över flera filer på en gång och då anger du ett wildcard som filnamnet du vill hämta. Det kan se ut så här:

download ls*.zip

Xmodem klarar inte flera filer, men de andra protokollen gör det.

Det är mycket vanligt att binärfiler hämtas som textfiler och därmed blir de skadade. Du kan inte fixa dem med PKZIPFIX utan du måste hämta filerna igen. Samma sak gäller när filer skickas in. Felaktiga arkiv tas bort direkt men du kan inte se vem som har skickat in en fil varför vi inte kan meddela vederbörande att filen är felaktig.

Glöm inte att skriva filnamnen med små bokstäver eller stora bokstäver beroende på vad som gäller för den aktuella filen. Filen Fil.txt är en annan fil än filen fil.TXT.

receive/upload

Kommandot används för att skicka upp en fil till monitorn med ett valfritt protokoll. Upload är en synonym. Filer som du skickar in hamnar i biblioteket /upload. Kontrollera gärna med kommandot viewarch ifall arkivet är läsbart om du har skickat upp ett sådant.

Skickar du in PC-program så skicka in dem i ziparkiv med en zipkommentar. Du skriver en kort kommentar på en fil helst med två blankrader i början och lägg in det i arkivet så här i DOS:

PKZIP -Z ARKIV < TEXT.TXT

Kontrollera gärna resultat med pkzip/pkunzip -v. Skulle du använda svenska tecken och det ser konstigt ut så prova att lägga till -Q när du lägger in arkivkommentaren. Du skall använda vanlig IBM/PC ASCII.

server

Server är kommandot om du vill köra mot monitor med Kermits serverkommando. Det innebär att du ger alla kommandon till Kermit från ditt eget Kermitprogram. Om du inte har ett Kermitprogram som kan "prata" Kermit serverkommando bör du inte ge detta kommando för då kan du få intrycket av att systemet har hängt sig. Du bör dock kunna ta dig ur det genom CTRL-C.

Du kan fn enbart hämta filer med server-mode från det bibliotek som du befinner dig i. Du kan alltså inte göra cd från ditt eget Kermitprogram. Använd detta kommando främst om du kör med MS-Kermit eller C-Kermit. Du kan inte heller skicka upp filer med serverkommandot eftersom vi har stängt av det. Det skulle nämligen innebära att du skulle kunna skicka upp filer till vilket bibliotek som helst sedan du har gjort "remote cd" och det är inte meningen.

calctime

Kan användas för att räkna ut ungefärlig överföringstid för en fil.

Det är meningen att du skall kunna göra beräkningen utifrån alla olika filöverföringsprotokoll och då skall väljas det som är ditt standardprotokoll. Detta är dock ännu inte helt klart och fn ges alltid beräkning med ascii som "protokoll". Du kan ange en eller flera filer som parameter eller ange antal bytes som parameter om du ger switchen -b.

Så här kan det se ut:

```
/upload>calc sun news.txt
Överföringstid beräknad på 15917 bytes i 1 filer:
Bps      Ascii (ungefärlig tid i minuter:sekunder)
9600     0:19
2400     1:09
1200     2:15
300      8:53
```

dir/lib

Listar innehållet i det aktuella biblioteket. För att du skall få fram storleken måste du ange switchar. Lib är samma sak som Dir. Lib är ABC-terminologi.

```
-s Lista filernas storlek i bytes
-k Lista filernas storlek i Kilobytes
-t Lista datum och tid då filen ändrades senast
-o Sortera. Om inte -o anges kommer filerna att listas i den ordning som de ligger råkar ligga i biblioteket. Efter -o anger du på vad du skall sortera. Ex: Sortera på storlek, därefter i tidsordning: -ost eller -okt om du vill ha presentation i Kbytes. Bokstavsordning kan anges med '.', t. ex. -ot. dvs. sortera i kronologisk ordning, därefter i bokstavsordning.
-r Visa filerna i omvänd sorteringsordning.
-f Visa enbart filer.
-d Visa enbart underliggande bibliotek.
-a "Allt". -a är en kortform som ger samma effekt som "-ort.s".
```

/pc/comm>dir -a

Bibliotek /pc/comm, antal filer/bibliotek 34
Underbibliotek:

	kermit	901105	17.55	telix	901103	19.35
	procomm	900922	18.21	com-and	900920	16.46
	windows	900830	18.35	zmodem	900826	17.22

Filer:

	envoy100.zip	901029	20.24	95373	pipeln10.zip	900916	11.40	19867
	charcvt.zip	900906	20.27	6737	tf.zip	900830	18.23	23068
	zcopy.zip	900830	18.23	26911	minitel.zip	900830	18.23	33852
	com34.zip	900830	18.23	6605	texfree.zip	900830	18.23	23092

/pc/comm>dir -k

Bibliotek /pc/comm, antal filer/bibliotek 34

Underbibliotek:

com-and	kermit	procomm
telix	windows	zmodem

Filer: 34

	instal.zip	4k	asyncepec.zip	5k	backmail.zip	121k
	boac12.zip	8k	charevt.zip	6k	com34.zip	6k
	comsend.zip	24k	dosgat16.zip	114k	elseware.zip	32k
	envoy100.zip	93k	hotkz-47.zip	44k	minitel.zip	33k

Observera att alla filer inte finns med i exemplen ovan utan jag har kortat av det.

type/cat

Visa innehållet i en fil på skärmen. Du anger filnamn som parameter och ev någon switch enligt nedan. Kom ihåg att ange filnamn med små bokstäver eller i förekommande fall med en blandning av små och stora bokstäver. Det finns en fil i rooten som innehåller information om ändringar i systemet som har ett filnamn som består av både små och stora bokstäver. För att titta på den eller hämta måste du ange filnamnet rätt.

```
-a ASCII. Detta är default kommandot och det visar en normal textfil.
-h Hex. Visa som en unix hexdump.
-x Hex. Samma som -h.
-b "Binärt", på samma sätt som i gamla monitorn. Icke skrivbara tecken skrivs som en punkt.
-p "PC". Visa en PC-textfil på en 7-bitars terminal. Svenska tecken konverteras till 7-bits ASCII.
```

viewarch

Viewarch visar innehållet i en arkivfil.

```
-v Visa en utförlig lista (verbose)
-t Visa innehållet i en unix tarfil.
-c Visa innehållet i en compressad unix tarfil.
-a ARC eller PKPAK arkiv.
-h LHARC arkiv. Inte implmenterat ännu.
-z PKZIP arkiv. Alla PC-filer i programbanken är lagrade i denna typ av arkiv.
-o ZOO arkiv.
```

Så här kan det se ut när man tittar i ett ziparkiv. I den senaste versionen av unix unzip som används finns det två kommando för att lista innehållet i ett arkiv. Dels -l som ger en kortare lista och dels -v vars resultat syns nedan. Det är möjligt att den kortare versionen kommer att införas som standard och att du måste ange -v för att få den längre versionen.

```
/pc/comm/kermit>viewarch -z msk302t.zip
Arkiv: msk302t.zip
```

Åmsk302t.zipÅ comment: MS-Kermit testversion daterad 25 oktober

Length	Method	Size	Ratio	Date	Time	CRC-32	Name
140368	Implode	79461	43%	10-31-90	21:37	c370ea41	kermit.exe
44890	Implode	18771	58%	10-26-90	05:09	7be4915b	msr302.upd
185258		98232	47%				2

Detta är ett exempel på hur man kan se innehållet i en unix tarfil som är komprimerad. Innehållet är förkortat här.

```
/unix/kermit>viearch -c cku5A162.tar.Z
Arkiv: cku5A162.tar.Z
```

```
tar: blockfactor = 1
./kermit/
./kermit/ckcfn3.c
./kermit/ckuhdr.mss
./kermit/ckuker.bwr
./kermit/ckuins.doc
./kermit/ckuker.ann
./kermit/ckcmai.c
```

Viewarch kan ännu inte visa IBM:s åäö på ett rätt sätt. De försvinner helt och hållet nu.

Terminal

Skall du köra mot en annan dator använder du traditionellt en terminal. Det är en speciell utrustning med bildskärm och tangentbord samt med lite elektronik. Det är ingen dator och den är inte speciellt intelligent. Den kan i princip bara ta emot tecken från tangentbordet och skriva tecken på bildskärmen samt kommunicera med en större dator.

En terminal kan vara ansluten direkt via en kabel eller via ett telefonmodem. En känd terminaltyp är Digitals VT100. En annan är IBM:s 3270-terminal. IBM:s terminaltyp kan normalt bara användas mot större IBM-datorer medan Digitals terminaltyp används mot många typer av datorer bl a vårt monitorsystem.

Persondator

En personlig dator, t ex en PC eller en ABC-dator är ingen terminal även om de ibland kallas för terminaler. De är små datorer för en användare. En sådan liten dator kan efterlikna, emulera, en terminal.

Den vanligaste terminaltypen som man efterliknar är Digitals VT100-terminal. Program som efterliknar terminaler på bordsdatorer kan i regel också lagra på disk den information som kommer på skärmen eller skriva ut den på en skrivare. Den kan också ofta hämta upp en redan lagrad fil från skivminnet och skicka ut den som om du skrev in den på tangentbordet. Detta kallar man normalt inte för filöverföring utan det är bara en loggning av information och en ersättning för vad du annars skulle ha skrivit på tangentbordet.

Program som är gjorda för att föra över filer har ibland namn efter själva metoden att föra över filer. Kermit är ett sådant exempel. För att du skall kunna använda ett Kermitprogram kan de flesta sådana program också efterlikna en terminal. Detta innebär att du kan använda ett Kermitprogram som terminalprogram utan att föra över filer med Kermit.

Filöverföring

Vill du föra över en fil från din lilla dator till den stora eller omvänt använder vi begreppet filöverföring. Då använder du ett program som flyttar filen mellan den lilla datorn och den stora. Ofta kan också ett program som efterliknar en terminal användas för filöverföring, men det kan också vara ett separat program som anropas från ditt terminalprogram.

Nya monitorn

Har du använt ett gammalt program som har kunnat föra över filer enligt den gamla FILTRANS-metoden för ABC så kan du mycket väl fortsätta att använda detta program idag om du vill. Du kan dock bara använda det för att efterlikna en terminal och du kan inte använda det för att

write

Kommandot skall användas för att skriva till en fil när det införs.

cd/chdir

cd används för att byta bibliotek. Observera att du med kommandot setpar, val 8 - Personliga flaggor, kan få aktuellt bibliotek utskrivit i prompten.

Kom ihåg att du måste skriva biblioteksnamnen med små bokstäver eftersom systemet skiljer på små och stora bokstäver. Fn finns det inga bibliotek i systemet som har versaler eller en blandning av versaler och gemena.

pwd

Visar aktuellt bibliotek. Se även ovan ang. cd och setpar.

find Find är ett kommando för att söka efter en fil. Det är implementerat provisoriskt.

fileid

Fileid skall användas för att visa och ändra status på en fil. Kommandot är ännu inte infört.

rm/del

rm är ett kommando för att ta bort en egen fil. Det är inte infört ännu.

name/ren/mv

name är ett kommando för att byta namn på en fil som du äger. Kommandot finns ännu inte.

<1789>

Bo Kullmar

Grunderna för kommunikation

föra över filer eftersom vi inte har stöd för FILTRANS-metoden i nya monitorn.

Skall du bara köra som en terminal med din ABC-dator kan du alltså använda vilket program som helst som kan efterlikna en terminal. Skall du föra över filer måste du använda ett Kermitprogram som då också kan användas som terminalprogram. Du kan med ett program som inte har stöd för Kermitöverföringar ofta lagra inkommande text på fil, men detta är inte vad vi kallar filöverföring utan enbart loggning av text som kommer på din skärm.

Synkron kommunikation

Synkron kommunikation innebär att kommunikationen körs i takt hela tiden över linjen även om du inte skickar något tecken på linjen. Synkron kommunikation används huvudsakligen mot stora IBM-datorer och den är också normalt kombinerad med ett sidorienterat användargränssnitt, dvs användaren skickar hela sidor till datorn på en gång. Först när du skickar sidan till datorn tar den emot det. IBM:s terminaltyp 3270 är en synkron fullskärmsorienterad terminaltyp.

Asynkron kommunikation

Asynkron kommunikation innebär att du bara skickar ut de tecken på linjen som du vill sända. Skickas inga tecken på linjen är den "tyst". Asynkron kommunikation används till många olika typer av datorer, bl a vårt monitorsystem. Asynkron kommunikation är normalt teckenorienterad dvs varje tecken du skriver tas direkt hand om datorn och datorn som du kör emot ekar också tillbaka varje tecken till dig så att tecken som skrivs på din bildskärm kommer från den centrala datorn. Digitals terminaltyp VT100 är en asynkron teckenorienterad terminaltyp.

Modem

I regel så används olika typer av modem för asynkron och synkron kommunikation. Det gäller också kommunikationsprogram. I de flesta fall är det ett asynkront modem och ett asynkront terminalprogram som du behöver till din persondator. Enbart om du vill köra mot en stor IBM-dator kan ett synkront modem med ett synkront terminalprogram bli aktuellt. Det finns några asynkrona modem på marknaden som också kan användas som synkrona modem.

När det gäller asynkrona modem bör du normalt använda dig av snabba modem som kan prata i hastigheten 2400 bps, dvs bitar

per sekund. Enbart om du kör mycket sällan och kort tid bör lägre hastigheter bli aktuella. Med lägre hastigheter kan du råka ut för höga telefonräkningar. Till den nya monitorn skall skall ABC-klubben nu börja satsa på höghastighetsmodem av typen som kan köra 9600 bps. Enbart om du kör modem mycket och vill minska dina telefonräkningar kan ett så snabbt och kostsamt modem bli aktuellt. Börja i regel inte med ett så snabbt och dyrt modem.

Modemkabel

Du behöver en kabel mellan ditt modem och din dator om inte modemet sitter i själva datorn. Vad gäller PC så kan du köpa en sådan på många ställen, men kablar är inte billiga över allt. Vissa företag säljer dem billigt. Du kan också göra en egen kabel.

När det gäller ABC så kan du i regel inte köpa en färdig kabel idag utan du måste göra en sådan själv eller med hjälp av någon annan. Köper du en begagnad ABC-utrustning så är det möjligt att du får med en fungerande modemkabel.

Observera att du inte kan använda en skrivarkabel som modemkabel! Du kan inte heller använda en modemkabel avsedd för ABC till en PC även om det är samma typ av kontakter. I regel så är det standard i modemets kontakt men det förekommer även modem med udda typer av kontakter. Det är definitivt inte standard i persondatorns kontakt för modem även om många använder samma sorts kontakt, dvs stiften i kontakten är olika.

Monitorsystem

Monitorsystemet kallas för monitorsystemet av historiska skäl. Det går tillbaka till det första system på ABC80 som Gunnar Tidner ställde till ABC-klubbens förfogande för ca 8-9 år sedan. Monitor används ibland för att beteckna det användargränssnittet som man kör emot när man "pratar" med en dator. På den tiden var detta begrepp vanligt på DEC-10 som var en stor dator på QZ som Gunnar Tidner då hade kontakt med. Sedan dess har vi fortsatt att använda detta begrepp och det har alltså inget alls med bildskärm att göra.

<1789>

Bo Kullmar

PC-diskett 12

Framställning av diskett nr 12

PC-prenumerationsdiskett nr 12 har just skickats ut. Detta är berättelse om tillkomsten av just den disketten. Själva framställningen har också fotograferad av undertecknad.

Vilka program skall vara med

Den första fasen vid framställningen av disketten är att bestämma vilka program som skall finnas med på disketten. För detta ändamål finns det numera ett möte i MSG som heter PC-pren.disk där förslag kan diskuteras. Tanken är att göra en diskett med så brett innehåll som möjligt dvs så att så många som möjligt skall kunna använda programmen.

Denna gången borde vi ha med ett kalkylprogram eftersom det är ett allmän och populär programtyp. Sigvard Nilsson hade skickat in ett sådant till monitorn. Per Sten som kan Lotus 1-2-3 testade det och alla andra kalkylprogram som jag hittade på SIG:s CD-ROM skiva som fick plats på en diskett.

AEASY visade sig bara ett bra program så vi valde det.

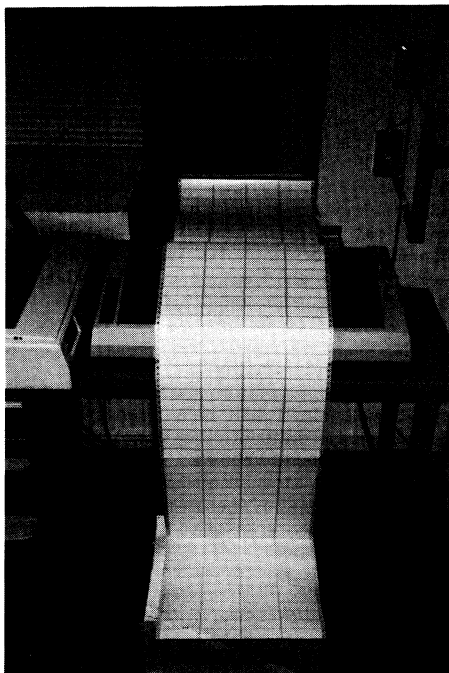
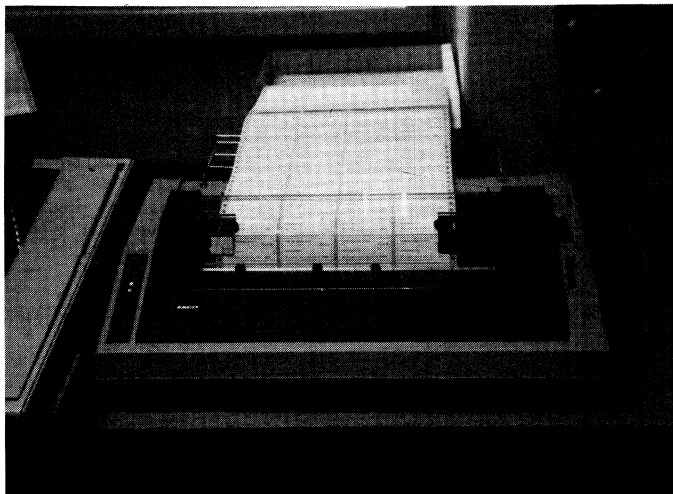
Två mastrar

Första fasen i framställningen av själva disketterna är att göra iordning två mastrar. En för 5 1/4"- och en för 3.5"-disketter. En infofil skrivs och den finns nedan. Nytt på denna diskett var att vi skickade med ett helt fritt program för att packa upp ZIP-arkiv. Det fungerar dock inte helt på samma sätt som PKUNZIP eftersom det inte kan packa upp filer till annat bibliotek än det aktuella.

Hur många disketter skall göras?

När mastern är klar skall vi ta reda på hur många disketter som skall skickas ut. Detta görs genom att ett program körs i kanslidatorn mot medlemsregistret. Distributionsposter skrivs i ett register för varje medlem skall ha en diskett och en fil med antalsuppgifter skapa också. Kontroll kan också göras mot en statistiklista från medlemsregistret som visar hur många diskett prenumeranter som finns. Denna gången var det 1102 st 5 1/4"-disketter. Antalet 3.5" var ungefär drygt 300. 3.5"-disketter har ökat märkbart sedan vi började med prenumerationsdisketter för PC.

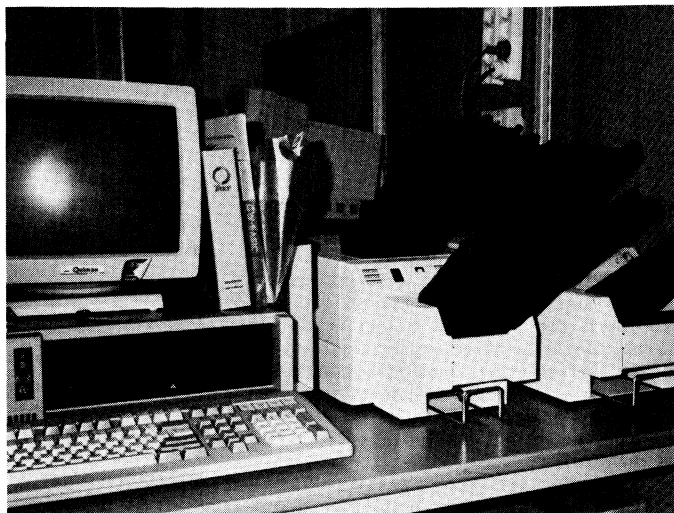
Från distributionsregistret och medlemsregistret skrivs sedan adressetiketter ut på kanslidatorn med hjälp av REGGEN. Därefter körs etiketter för disketterna ut. Vi använder samma typ av etikett för båda typerna av disketter numera eftersom den passar på både 5" och 3" disketter. Etikett-utskriften sker från REGGEN genom att vi flera gånger skriver ut varje post. Ettikett-layouten finns alltså i ett register och vi skriver ut en och samma post många gånger. Vi kan på detta sätt köra ut 200 etiketter åt gången eftersom det är lite begränsat hur många utskrifter från samma post som man kan göra. Det gör inget för det är ganska enkelt att skriva ut flera gånger.



Kopieringsförfarandet

Själva kopieringen sker numera på en Trace kopieringsutrustning. Den består av en PC med en 20 MB hårddisk och två sk autoloader. Det finns en autoloader för 3.5" och en för 5 1/4". Man läser in ett original med hjälp av en av autoloaderna och detta lagras på hårddisken och sedan körs kopiering från originalet på hårddisken. Nya disketter läggs i det övre schaktet och de kopierade disketterna kommer ut i det nedre facket. Längst ner finns ett fack där felaktiga disketter kommer ut. 5 1/4"-disketterna kopieras på cirka 15 sekunder och 3.5"-disketterna kopieras på drygt 30 sekunder.

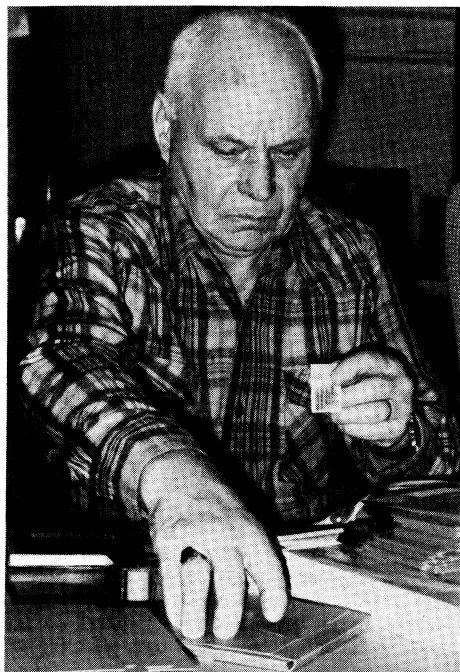
Många av de disketter som vi köper är inte bulk, dvs de förpackas i kartonger om tio disketter med påsar och dessa måste rivas upp och disketterna plockas fram och läggas i maskinen.



1. Etiketterna "trycks" på en Facit 4509..
2. fyra etiketter i bredd
3. PC:n med hårddisk och sina två autolodars



4. Löpande band vid packning under gemytliga former..
5. Koncentration behövs när man ser den stora packen.
6. Försändelserna skall föras med klubbens nya box-adress.
7. Resultatet av en kvälls möda, postsäckarna.



Packning

När alla etiketter och disketter är kopierade återstår bara packning av disketterna. Packningen sker ofta på tisdagskvällar då det ändå är folk i klubblokalen. Fram till nu har många 5 1/4"-disketter packats i pappkartonger som viks ihop och sedan tejpas. De måste också stämplas och avsändaradressen måste sättas på. I fortsättningen kommer vi att enbart packa i kuvert med pappskivor. Det blir något dyrare, men är mycket enklare och snabbare att hantera eftersom vi använder förtryckta kuvert som inte behöver stämplas eller föras med etikett med avsändaradress.



Oläsbara disketter

Vi har noterat ett ovanligt stort antal disketter som har varit oläsbara hos medlemmarna. Vi har fått tillbaka ett 10-tal och skickat ut nya ersättningsdisketter. Har du en oläsbar diskett så meddela kansliet så får du en ny. Du behöver inte skicka tillbaka den gamla.

Vi har fått besked från Fromat Data att det troligen är fel på vår kopiator och de kommer att fixa den inom garantin. De hävdar också att våra ABC-disketter är dåliga men att de kan användas för 48 tpi. De säger också att de är "vandrande", dvs att de efter en tid kan bli oläsbara vilket delvis kan förklara vad som har hänt.

PC-012.INF:

PC-012.INF

Denna informationsfil

UNZIP.EXE

Unzip v 3.98, för att packa upp arkiven på disketten. OBServera att detta inte är PKUNZIP utan ett helt fritt program för att packa upp ZIP-arkiv. Främsta skillnaden är att programmet inte kan packa upp arkivet i annat bibliotek än det aktuella. Se nedan.

DOSHELP ZIP

Ett litet trevligt program som visar hjälp för olika DOS kommandon.

TETRISS ZIP

Det kända spelet Tetris för Windows 3.0!

TEXFREE ZIP

Ett fritt Videotexprogram. Glöm inte att lägga in dina egna uppgifter som entre-nummer, lösenord och telefonnummer i programmet innan du testat det!

DOSFIXA ZIP

Ett svensktillverkat hjälpprogram för DOS-användaren.

ASEASY ZIP

Ett Lotus 1-2-3 liknade kalkylprogram.

ICONDRW ZIP

En ikoneditor för Windows 3.0 med vilken du kan göra ikoner med.

För att använda programmen på denna diskett måste du först packa upp dem! OBServera att flertalet program är användarstödda och användaren förväntar sig en slant om du regelbundet använder programmet. Det är dock helt gratis att en kortare tid prova programmet.

För att packa upp arkiven kan du använda UNZIP.EXE som finns med på disketten eller PKUNZIP.EXE som du kanske redan har. Använder du UNZIP.EXE så ställer du dig i det bibliotek där du vill ha dina filer uppackade och skriver:

A:UNZIP DOSFIXA

om du vill packa upp DOSFIXA.ZIP och om du har disketten i fack A: samt befinner dig på någon annan disk än C: för annars blir A: full.

90-10-26

ABC-klubben, Bo Kullmar

PC-diskett 13

Här presenterar vi fem olika programpaket av vilka hela tre är svenska och gjorda av medlemmar! Vi ser gärna att vi får tillgång till flera svensktvecklade PC-program!

PC-013.INF

Denna informationsfil.

UNZIP.EXE

Unzip v 3.99, för att packa upp arkiven på disketten. OBS! Servera att detta inte är PKUNZIP utan ett helt fritt program för att packa upp ZIP-arkiv. Främsta skillnaden är att programmet inte kan packa upp arkivet i annat bibliotek än det aktuella. Se nedan.

BOOT12.ZIP

Innehåller ett program som gör det möjligt för dig att ha olika AUTOEXEC.BAT och CONFIG.SYS. Du kan välja mellan dessa vid uppstart.

CDISK450.ZIP

Innehåller ett programpaket för att katalogisera disketter.

TID.ZIP

Är ett enkelt program som skriver ut tiden. Det är tänkt att användas i BAT-filer för att skapa loggar. Av Bo Bo Michaelsson, <913>. Se även ABC-Bladet nr 3, 1990 sid 3.

WPMENU.ZIP

Ett meny, skalprogram, av Anders Umegård, <4396>. Initieringsfilen måste finnas i rotbiblioteket. (Ö).

XMAS.ZIP

En julkhälsing från Sten Öhman, <6127>.

För att använda programmen på denna diskett måste du först packa upp dem! OBS! Servera att flertalet program är användarstödda och användaren förväntar sig en slant om du regelbundet använder programmet. Det är dock helt gratis att en kortare tid prova programmet.

För att packa upp arkiven kan du använda UNZIP.EXE som finns med på disketten eller PKUNZIP.EXE som du kanske redan har. Använder du UNZIP.EXE så ställer du dig i det bibliotek där du vill ha dina filer upppackade och skriver:

A:UNZIP XMAS

om du vill packa upp XMAS.ZIP och om du har disketten i fack A: samt befinner dig på någon annan disk än C: för annars blir A: full.

90-12-01 ABC-klubben, Bo Kullmar

HACK III

Hackerence III 2-4 november

En sammanfattning av vad som hände under Hackerence III 2-4 november 1990

Mellan den 2 och 4 november arrangerade Unga Forskare Härnösand en sk Hackerence, dvs en konferens för hackers där datorintresserade träffas och byter erfarenheter med varandra. Inträdet var 30 SEK eller en full PD-disk.

Fredagen den 2 november kl 18.00 startade Hackerencen men deltagare hade redan börjat strömma in kl 16.00. Deras ursäkt var att man måste ju få en bra plats. Vi hade riggat upp ett "kansli" som sålde disketter, CC, Snickers, Chips (ätbara), kaffe etc etc (och annat nödvändigt). På plats fanns ett stort PD-bibliotek på 38.400 filer(!) som alla kunde ta del av.

På fredagskväll/natt var den sk "kulmen". Då stod det skärmar överallt. Totalt var det ca 80 st datorer av typerna Atari, PC och en DS90-20 (Benny Löfgrens "lilla leksak") och dessutom 220 st deltagare. Vårt (eller mitt) största problem var att ge ström åt alla datorer. Säkringarna gick nämligen på löpande band.

På lördagsmorgonen kl 00.00 började programmeringstävlingen, där de tävlande fick 24 timmar på sig att lösa vissa programmerings- och matematiska problem. Vid 4.00 började en SIMCITY-tävling.

Söndag morgon kl 00.00 började en tävling i CROBOTS. CROBOTS går ut på att man skall programmera en tank i ett C-liknande språk. Med sin radar och kanon skall man försöka vinna över de andra programmen. Detta intresserade många!

De priser som fanns till tävlingarna var 10 st helårsprenumerationer på svenska Hemdatoren, 2 st analoga joystick till PC:n, diskettbox, musmatta och en läskande frukt-korg.

Naturligtvis utbyttes en massa erfarenheter under Hackerencen mellan deltagarna. AV en annan hjälpsam hacker fick jag de första insikterna i C-programmering. Kopiering från vårt PD-bibliotek var också en syssla som en del roade sig med.

Under Hackerencen hade vi sålt 1500 st 3.5"-disketter, 42 st backar CC, 170 st Snicker, 60 st 200 g Chips-påsar samt ca 200 pizzor och 75 st mackor. Detta gav ett uppsving på ca 6.000 SEK i UFH:s ekonomi. För dessa pengar skall vi köpa oss ett V.42bis-modem.

Kallelse till HACK IV

Hackerence IV som skall hållas i Härnösand den 10-12 maj 1990.

Vi satsar på att

- bli 400 deltagare
- ha tillgång till det gigantiska PD-biblioteket på 38.400 filer
- ha tävlingar i programmering
- ha tävlingar i spel
- ha tävlingar i demos
- ha försäljning av CC
- mm mm

Som du säkert förstår måste du vara med om detta någon gång under ditt liv som "hacker"

Intresserade av HACK IV kan enklast höra av sig för fribiljetter genom en telefon-signal eller ett brev i monitorn.

PS Glöm inte att uppgge namn och fullständig adress

<5523>

Hugo Wikström
0611-230 55

Radannons

Säljes

Parallell LapLink 3.0 kabel säljes

Pris: 175:- + frakt och ev pf

Vid skriftlig förfrågan kom ihåg att ange telefonnummer

<5523>

Hugo Wikström
0611-250 55
nätbrev 2:202/603

Med hjälp av parallell LapLink kan man överföra ca 3.0 MB/min. Detta är särskilt bra vid backup och överföring av stora datamängder mellan två PC-maskiner. Kabeln är tre meter lång och kan för en extra slant göras längre

Minnen

Under min skoltid fick man använda ett då vanligt räkneredskap, nämligen räknestickan. Jag blev mycket intresserad av sådana och hade flera olika, bl a japanska av bambuträ, några tyska av olika system och plaster samt en rund.

På universitetet arbetade vi mycket med räknescrutar. Vi räknade många saker på samma sätt och jag började dunkelt ana att man kanske kunde använda datorer för det. Institutionen hade dock inga och saken fick vara. Jag skaffade ett begagnat piggsvin för 275 kr (Odhner-snurra) och räknade i årtal med denna pålitliga apparat. Piggsvinet krånglade aldrig och man kunde göra mycket knepiga saker med dess hjälp. Konsum ordnade för ett antal år sedan en insamling av piggsvin för att skänka dem till behövande länder i Afrika. Därmed har vårt land egentligen gjort sig av med en viktig reserv och gjort medborgarna beroende av högteknologi även för enklare beräkningar.

Under 1973 och 1974 kom de första fickräknarna. De var alldeles för dyra men lovade att bli överkomliga om den vanliga transistortrenden upprepades. Min första kostade 175 kronor på PUB år 1975 och var en Hugin. Den hade stora gröna siffror och pep lite olika beroende på vad som gjordes. Jag åkte buss en gång och fick en fråga av en dam bredvid om "man kunde räkna procent och sådana där svåra saker med den". Jo, räknaren hade en tangent för procent men den har jag aldrig använt, varken då eller senare. Minnesfunktionen hade jag stor nytta av. Jag började fundera på hur det skulle vara att ha flera minnen.

Första gången jag satt vid en dator var på AMS samma år. Under en tid gjorde jag en juridisk utredning om olika stödformer och vilade mellan varven med att titta på olika offentliga saker i den centrala datorn. Jag minns att det var en jättelik terminal som mera påminde om tandläkarmottagning. Efteråt har jag förstått att det var system-konsolen jag satt vid. Man kom in i systemet efter ett nästan omöjligt inloggningsförfarande.

Sedan hjälpte jag till med ett föreningsregister som låg på hålkort. Jag stansade flera kort någonstans på Teknis.

En släkting arbetade med en dator vid en ingenjörskontor. Jag fick starta den med RUN och kände mig mycket stolt samt tyckte "att det här var ju enkelt". Maskinen lagrade på hållremsor vilket kändes ganska primitivt.

Min första egna programmerbara utrustning köpte jag efter nyåret år 1977. Det var en Texas TI 57, en trevlig programmerbar räknare man kunde göra märkliga saker

med. Jag minns att jag gjorde program som skulle beräkna exponeringstiden vid förlängt utdrag på storformatkameror och ett liknande för att beräkna belysningstiden i en förstöringsapparat som funktion av den belysta pappersytan. Jag fann dock att räknaren förlorade sitt mödosamt inknyttade program då den stängdes av och det därför inte var någon ide att använda räknaren då jag skulle fotografera eller förstora. Det var enklare att göra en tabell eller ett koordinatsystem. I stället gjorde jag ett program som avslöjade räknarens interna fel genom att jämföra kvadraten på räknarens beräknade rot med den av mig kända, exakta kvadraten. Anders Franzen, på den tiden lika framåt som idag, gjorde ett startrek-spel på räknaren. Det var en prestation med de 50 programstegen. Efter ett par år gick det inte att ladda ackumulatören längre och dessutom föll tangentbordet samman. Sedan blev det inte så mycket programmerande på TI 57.

Det fanns andra räknare från Texas. TI 58 hade nog bortåt 500 programsteg men tydligen ingen möjlighet att lagra programmen. Jag båkade vid tanken på att behöva skriva in alla stegen varje gång man skulle arbeta med maskinen. Det skulle nog bli så att man aldrig stängde av räknaren. En större modell var TI 59. Den kunde lagra sina program på små magnetmanschetter. Idén tycktes jättepraktisk. Tyvärr var räknaren mycket dyr. Jag skulle inte heller ha någon nytta av en apparat som inte kunde hantera bokstäver.

Då TI 57 hade gått sönder tittade jag efter räknare med bättre tangentbord. HP hade flera men den omvända polska notationen gjorde mig trött redan från början. Jag köpte aldrig någon.

År 1980 dök det upp en liten dator som hette Sinclair ZX80. Leverantören hade sin agentur i porten där jag bodde och jag beställde en maskin som lät vänta på sig länge. En av de anställda sekreterarna tyckte att jag var besvärlig som ringde på och tjatade så ofta.

Efter en tid kom så äntligen byggsatsen och jag började löda ihop den. Det var ett ganska petigt göra och jag råkade vända kretshållarna fel. Det gjorde inget om man bara satte in kretsarna tvärs emot markeringarna. Efter visst stöd hos leverantören fungerade faktiskt maskinen. Jag skaffade en billig kassetbandspelare till den och hade ett sekundärminne. Det skulle vara en riktigt enkel bandspelare, ju enklare desto bättre. Maskinen hade ett primärminne om något 1000-tal bytes, tror jag. Sedan byggde jag ut primärminnet med en kloss som man satte fast baktill. Det var en

mycket ranglig anordning och ofta, alltför ofta, vickade minnet till och allt jag hade matat in försvann.

Jag gjorde ett demonstrationsprogram att köras i en av museets salar. Detta blev inte någon succé hos ledningen. Man förstod inte vitsen med att ha en dator i salarna och som skulle kunna berätta om föremålen. Längre har man inte kommit idag, tio år senare. På det pedagogiska området finns en hel del att göra med datorer. Museifolk är ett konservativt släkte.

Min ZX80 dumpade hela minnet med data och program på bandet. Det hade både fördelar och nackdelar. Man kunde visst avbryta en bearbetning och spara den som den var på bandet. Genom att senare ladda maskinen med samma band kunde man fortsätta processen. Maskinen spelade in ett fullständigt mardrömslikt skrikande ljud på bandet. Det blev för övrigt ofta fel på bandet och då kunde man vara tvingad att knappa in alltihop igen på det hårda, plastiga tangentbordet.

Jag gick en del kurser på KomVux för att få tillgång till maskintid, som det hette då. Vi hade en del små Pet Commodore, mest kända för sina mikroskopiska tangentbord och, för all del, sina krånglande bandstationer. Med dessa som erfarenhet förstod jag att jag måste ha ordentliga grejor. Pet-en var dock inte dum och jag tänkte köpa en begagnad hos en liten källarfirma på kanske Frejgatan. Firman hette i alla fall Datorisering och har sedan vuxit ordentligt i likhet med hela branschen. Datorn hann bli såld innan jag kunde bestämma mig och det kanske var tur för mig. I en radio- och TV-broschyr från Luxor och som jag fick från min pappa stod det nämligen något om en ny dator som hette ABC80. Den verkade lovande.

Jag började besöka databutiker och hamnade ofta på Kungsgatan hos Computerland. Det var en massa ungar där, minns jag. Vad de gjorde fick jag aldrig kläm på men att maskinerna var dyra förstod jag snabbt. Maskinerna var nog mest Apple men det fanns även en ABC80. Att få tag på en försäljare var omöjligt. Dessa var alltid ute på uppdrag någonstans och hade inte tid med kunder som inte hade någon dator. Ett antal andra butiker besökte jag också men man var oftast fruktansvärt ointresserad. Så länge det gällde Luxor var det mest radio- och TV-butiker och för deras personal var datorn lika ny som för kunderna. Personalen stod mest och hängde över maskinerna, uppträdande som om kunderna

var ett otyg och som störde i försäljarnas viktiga arbete med butikens datorer.

Sedan ringde jag Luxor och ställde en massa frågor. Jag funderade på att skapa ett register över negativ och undrade om man kunde lagra en bild i datorn. Jag menade textinformationen och tyckte att de 16 kilobytes som datorn var utrustad med inte skulle räcka för så många av mina negativ. Att datorn hade sekundärminne och att data kunde lagras där och läsas därifrån förstod jag inte ännu. Jag insåg att jag måste läsa böcker och gick igenom allt som fanns på biblioteket.

Filer hade jag inte kommit i kontakt med ännu. Sinclair ZX80 kunde inte hantera filer och KomVux sysslade mest med problemlösning, loopar och flödesschemor. Men inget ont om KomVux, jag hade stor nytta och glädje av den skolförmen.

Det kostade i mesta laget att köpa en ny ABC80. Den kostade begagnad i allmänhet 5500 kr. I en annons fann jag en som kostade 5000 kr. Jag besåg den och ringde nästa dag och sade att jag ville köpa den. Ingen hade lyckligtvis bestämt sig även om flera hade kommit till säljaren för att höra sig för.

Anders Franzen och jag tog pendeltåget från Centralen. Då jag, som råkade komma först, stod och väntade, kom en konstig typ gående i gången, drog fram en stor, svart pistol ur jackans innerficka samt gjorde en mantelrörelse, tror jag det kallas. Vi satte oss för säkerhets skull i en annan vagn. Vem visste vad den sortens galning kunde ta sig för? Det är verkligen inte alla människor som sysslar med datorer.

Säljaren hade tänkt använda maskinen för att styra verktygsmaskiner men fann att datorn var för långsam. Själv tänkte jag använda den för redaktionellt arbete och säljarens hustru utropade förvånat "Kan man använda den till det också!?"

Vi kånkade hem datorn. Bildskärmen var av den äldre typen med vit kåpa. Bruksanvisningen (det hette fortfarande så då) var lösa fotostatkopior och behandlade bl a hur man skulle ansluta ABC80 till en kassettradio från Luxor för att på det viset få ett kassetminne. Min maskin måste ha varit ett av de första exemplaren. Den hade 16 kB minne och en skärm med 40 tecken på bredden. Sekundärminnet var en apparat som tydligt var en ombyggd bandspelare för ljud. En liknande tingest kunde jag se inbyggd i en stereoanläggning som verkligen hade allt i samma hölje.

Med datorn följde också en blå bok som handlade om de första stegen med ABC80. Boken hette "ABC om BASIC" eller något liknande. Den var mycket lärorik och lättfattlig samt skriven på svenska språket. Denna egenskap uppskattade jag mycket.

Sedan började hackandet. Jag satt uppe sent, sent på kvällarna och lät min skrivbordslampa lysa något lite på tangentbordet. Det mesta av ljuset fick lysa ut genom fönstret.

En dag ringde det på dörren. Jag hörde en kvinna klaga över att min skrivbordslampa lyste in på dem i deras sovrum på andra sidan gatan. "Den lyser hela nätterna och många i huset undrar över den" sade hon. Jag blev sur och brydde mig inte om

att förklara genom min fortfarande stängda ytterdörr.

Goda vänner gav mig rådet att dra ner rullgardinen.

Senare fick jag ett tackkort från mina belysta grannar på andra sidan den faktiskt ovanligt breda Wollmar Yxkullsgatan. Än i dag vet jag inte hur kvinnan såg ut som ringde på. Om hon känner igen sig får hon gärna höra av sig.

Det här var hösten år 1980.

Någonstans hade jag läst att en klubb som hette ABC-klubben hade bildats. Efter många försök lyckades jag så småningom få tag på någon som kunde göra mig till medlem. Det var hemskt svårt att få tag på en sådan person.

Under sommaren 1981 började jag skriva ett program för hantering av museets data. Programmet skulle bara användas för inmatning och förberedande rättning samt utskrift. Att jag ville ha kompensationsledigt för mitt arbete förstod man först inte. Det var ju en hobby, ansåg man, men bladet vände snart.

Mitt program kördes på ABC80 men konverterades och utvidgades för drift på ABC800 då den datorn kom. Programmet är fortfarande i drift vid några arbetsplatser i Refugen.

Så småningom tyckte jag att det kunde vara kul att använda datorn för kommunikation. Jag hittade ett billigt - 1450 kr - akustiskt, 300 bauds modem. Det kom på posten från DataLab tillsammans med en terminalrutin på kassett. Företaget hade fått tag på ett parti Luxor-modem som man målat i blått i stället för sandgult och man sålde dem förhållandevis förmånligt.

Mitt modem kvittrade och tjöt beroende av vad man gjorde med det. Man hörde när den uppringda datorn svarade och då fick man kvicka sig på att passa in telefonluren i modemets skumplastklädda fack och försiktigt men ändå snabbt lägga igen luckan. Lamporna lyste trevligt och proffsigt. Medan man kommunicerade fick man inte tala för högt i rummet för modemmet hörde även en del främmande ljud. Ljuden störde funktionen och kunde spolia en filhämtning. Att oförsiktigt lägga något på bordet där modemmet stod var också farligt liksom att slänga igen luckan hårt.

Jag körde mot Datatronics konferensträd och ABC-klubben med den tidnerska monitorn. Man fick verkligen vara på hugget för att komma in. Det var ofta upptaget. I den tidnerska monitorn kom jag på ett knep att få vara inloggad längre. Om man skrev "login" eller motsvarande medan man var inloggad så nollställdes klockan och man fick mera tid tillgodo. Jag använde inte denna egenhet annat än vid konstiga tider då ingen annan normalt loggade in. Jag slapp att ringa upp igen och knappast någon blev lidande.

Trädet skrev man inlägg i. Det grenade ut sig genom sina otaliga kommentarer inte bara på djupet utan också på bredden. Detta var en lösning som jag aldrig har sett senare. Tekniskt rullade det nog på en Pet med hårddisk. Hårddisk var ett ganska okänt begrepp då och priserna var hutlösa medan kapaciteten var låg. Trädet var skrivet i Forth vill jag minnas.

Jag läste många inlägg i alla möjliga frågor. Jag skrev många, en del kloka, en

del antagligen mindre kloka. Jag gjorde reklam för en förening genom att beskriva föreningen i ett inlägg och sedan i en underordnad kommentar skriva in föreningens program för terminen samt fick beröm av Sysop för detta sätt att använda Trädet. Vi diskuterade allt, från matrecept till medicin. Trädet blev verkligen ett sätt att leva i nuet. Man kunde associera fritt och kommentarerna avspeglade de fria tankarna.

Så en dag var trädet plötsligt borta. Jag ringde en man hos Datatronic och han sade att det inte fanns längre.

Jag hade då grubblat över hur jag skulle kunna bevara inläggen jag ville spara, jag skrev t o m till Sysop och bad honom göra ett SAFT-protokoll i Trädet så att jag kunde använda det. SAFT (Simple Ascii File Transfer) hade visst utvecklats hos QZ och var tänkt att bli en standard för filöverföring. Jag kunde aldrig få fram en dokumentation över SAFT vilket var en förutsättning för att Sysop skulle kunna implementera detta protokoll. SAFT var aldrig särskilt bra. Det ersattes av andra metoder. Det fanns heller inga skärmdumpprogram till ABC80 och VT100-emulatorn hade inte kommit ännu.

Trädet var verkligen ett kulturellt lyft, en innovation. Jag sörjer uppriktigt över att det försvann och att inget i det är bevarat. Denna erfarenhet har gjort att jag föreslagit ABC-klubben att spara allt från de egna konferenssystemen. Vilket material att forska i!

Den tidnerska monitorn var något annat. Man kunde hämta filer och program sedan man laddat den egna datorn med en hopp-lös terminalrutin från Scandia Metric, T80PRT eller något liknande hette den. Ovanpå den hade man ytterligare ett program försett med nödvändiga filhanterare m m. Monitorn hade passande rutiner. Vad jag vet finns det fortfarande stöd för dessa metoder i klubbens ABC802-baserade monitor. Jag tror dock att det är få personer som använder dessa föråldrade tillvägagångssätt.

Det fanns inget MSG i monitorn utan man fick skriva filer och lagra dem. Sedan kunde andra med hjälp av filnamnen tänkas avgöra om innehållet var intressant.

Som terminalrutin användes T80PRT och den hade jag fått av en programmerare på Karolinska institutet. Hur jag fick kontakt med honom minns jag inte. Han hade försett rutinen med några små ändringar men det betydde inget för mig. Programmet användes både för kommunikation och som stöd för seriell skrivare. Det stödde visst också Teledata eller Datavision som sedan kom att heta Videotex.

Jag började lägga upp ett eget programarkiv på kassett. Filerna hämtade jag med mitt akustiska 300-modem och jag lagrade dem fortfarande på mitt kassetminne - förlåt - bandstation. Det tog lång tid, minns jag. Sedan kunde man få en drop-out på bandet och så fick man göra om alltihop, om man kom in förstås.

Man kunde också hämta från ELFAs datasystem. Funktionen liknade klubbens system. En del program var användbara (virus var ett okänt begrepp) medan kvaliteten på texterna oftast var sämre, antagligen på grund av att systemet var öppet för alla. En gång var det ur funktion och då ringde jag och frågade. Jo, den vänliga damen

skulle be någon sätta igång det. Att ha det gående ständigt hade man tydligen inte tänkt sig.

Jag köpte 16 kB extra minne från - ja, hette han GeJo? - och lödde in det. Det fungerade men jag var genomsnittligt efter operationen. Det extra minnet gjorde att jag kunde hantera ett riktigt adressregister som jag hade funnit i "Mikrodatorn", en skrift som finns kvar idag men har förlorat sin karaktär av nydanare och pionjärblad. Nummer efter nummer var fyllt av nya agenturer och representerade fram till dess okända tillverkare. Maskinen skulle snabbt ta halva marknaden, lovade de glada slipsförsedda figurerna. Registret hette REGPROG och var riktigt användbart. Upphovsmannen hette nog Istvan Toth. Jag har arbetat med REGPROG i nästan tio år, faktiskt. Man kan ju tycka vad man vill om det, men REGPROG och ett av mina egna mästerverk, EDLIST, gjorde att jag kunde hantera informationen och ta ut valfria fält m m. EDLIST rymdes i primärminnet och har alltid rymt vad som skulle rymmas.

Att jag körde systemet i tio år kanske väcker förundran. Men det finns försvar för detta: Så länge ett system fungerar och gör vad det ska, finns det inte anledning att byta. Men om kunnandet om en utrustning börjar gå förlorat kan man en dag bli stående med en trasig maskin och kan inte flytta sina data till den nya man måste skaffa. Det är den verkligt svåra förlusten.

ABC-klubbens monitor hade en klocka som alltid gick fel. Det irriterade mig och jag undrade om man inte kunde få klockan att visa hur länge till man fick vara inloggad i stället. Det hade ingen tänkt på förut och jag fick beröm för idén. Det blev aldrig så ändå, för monitorn försvann och MSG kom så småningom i stället. Bo Kullmar fick fart på programmerandet och gjorde sig känd för sitt kunnande, sina lösningar, sin hjälpsamhet och sin entusiasm.

Ulf Sjöstrand lånade mig ett extra kassetminne och en märklig växel som gjorde att man kunde ha två sådana minnen kopplade till samma ABC80. Genom en styrd omkopplare skulle man kunna läsa på det ena bandet och skriva på det andra. Jag lovade att skriva om apparaten i ABC-bladet. Trots mina tappra försök med det lånade omkopplarelandet fungerade det inte. Jag lämnade tillbaka sakerna, blev arg och köpte en floppy i stället hos Anders Björnhem på Upplandsgatan. Jag hade på något vis snokat reda på att Luxor hade ett restlager av den gamla stationen FD2. Denna hanterade flexskivor om 70 KB. Jag föreslog Björnhem att han skulle sälja de här utgångna stationerna och jag köpte en av honom efter en tid. Burken, kallad "skokartongen", kostade 5000 kronor. Att jag själv kunde ha köpt partiet och sålt det till klubbens medlemmar föll mig aldrig in.

Flexskivan var ett verkligt lyft. Jag köpte flera paket Verbatim i en jättelägenhet på Strandvägen. De var billigare där än på andra ställen, märkligt nog. Mina disketter kostade 250 kr för tio stycken, idag ett skamlöst högt pris i kronor även

med dagens löner och penningvärde. Disketterna rullar dock fortfarande, nu i PC-format.

Allt gick mycket snabbare med flexskivan. Man kunde ha flera filer i gång samtidigt, man kunde sortera större material än vad som rymdes i maskinens primärminne o s v. Det fanns också speciella program med vilka man kunde gå in på skivan och radda filer som råkat ut för oläsliga tecken. Det hände då och då med FD2. Med en nyare version av "Disk Operating System" försvann de flesta problemen. Antagligen hade det med formateringen att göra.

Som det brukar gå då det kommer en ny sak med nya möjligheter satte det spår i min verksamhet. Jag kunde börja laborera med direktfiler. Dessa var en omöjlighet på kassetbandet. Nu kunde man gå in i en post utan att läsa alla data och utan att behöva få in alla data i primärminnet. Filerna kunde vara mycket större och programmen kunde göras effektivare. Detta konstaterade jag men fortsatte sedan med mitt gamla registerprogram. Det var trots allt ganska krävande med direktfiler och registren var inte så stora heller.

Björnhem blev min huvudsaklige leverantör av dyrare saker. Han, hans fru och Håkan var alltid hjälpsamma och vänliga. Håkan kunde fixa allt som krånglade och bygga lustiga saker. Det fanns för övrigt gott om databutiker för åtta år sedan. Bara i trakterna kring Slussen - Mariatorget i Stockholm fanns det tre stycken. En del sålde ABC-er, någon hade specialiserat sig på speldatorer och en hade en märklig maskin som bara kunde programmeras i Forth. Detta språk prövade jag på - genom ABC-klubbens försorg. Det var ett nytt sätt att programmera och ganska roligt också. Det blev dock snabbt obegripliga program av koden, även för programmeraren. Som jag inte ville skriva om programmet då jag ville ändra i det fick Forth ligga på sin diskett.

År 1982 betalade museet en 80-teckens-tillsats som ersättning för arbete jag hade gjort på min maskin hemma. Denna monterades av Turn-Key Data vid Kungsbroplan. Museet köpte också en dator där - det var en av den nyintroducerade typen ABC800M. Hos Turn-Key hjälpte man oss genom att svara på många frågor. Vi hade tänkt köpa flera maskiner där, men ÖB hade gått ut med ett påbud att alla försvarsmyndigheter borde köpa från Försvarets Datacentral för att komma i åtnjutande av service och statliga rabatter. Därmed gick Turn-Key miste om ordern på två maskiner och jag har fortfarande lite dåligt samvete för det. Sedan gick det dåligt för Turn-Key och företaget försvann. Men det kan vi knappast lastas för.

I samband med monteringen av 80-tecken började min dator att krångla. Den tappade sin information och långa texter försvann ur minnet om jag inte hade sparat dem. Jag misstänkte den nya tillsatsen och den var också en av felkällorna. Jag tryckte fast kortet ordentligt men en del störningar fortsatte att uppträda. Sedan jag några gånger hade lossat alla kretsar i hela datorn ur deras kretshållare någon millimeter samt tryckt ner dem igen fungerade ABC80

som den skulle. Under åren har det måst upprepas någon gång.

Att museet plötsligt kunde köpa tre ABC800M hade sina egna poänger. Genom kontakter med andra museer kom vi i kontakt med något som hette DFI - Delegationen För vetenskaplig och teknisk Informationsförsörjning. Det var en statlig myndighet som tycktes ha gott om pengar. Jag fick resa till Düsseldorf och besöka stadens mäs-sa. Denna mäs-sa gav en försmak av stora världen - där var folk välklädda medan folk hemma ännu gick klädda i 1970-talets trasmode. Spåravnarna hade antagligen digitalt ljud i sina högtalaranläggningar, ljudet var tydligt och klart.

Mässan i Düsseldorf har fortfarande sina efterdyningar. Sedan dess plågar mig nämligen ideliga försändelser från en avsändare i Holland med sina erbjudanden om dyra seminarier och mässor. De går direkt i retur-papperet.

DFI-projektet gick ut på att vi skulle lagra museets bilder på videoband. Tekniken skulle sedan användas av "liten och medelstor industri" för att lagra bilder. Att arbeta med videoband för det ändamålet trodde ingen på utom DFI, men vi kunde köpa datorer, skriva program, anställa en tro-tjänare som fortfarande strukturerar och uniformerar våra inmatade data och göra mycket annat. Jag skrev en stor skrift som beskrev projektet och som sedan efter bearbetning skulle publiceras i en rapport. Rapporten syntes aldrig till och sedan DFI upphört lär ingen kräva oss på någon vidare redovisning. Projektet gav oss i alla fall en mängd erfarenheter och gjorde oss kända i museernas speciella värld. Många, många besökte oss. Vi tillhörde de yppersta i museivärlden. Sedan har CD-ROM kommit och gjort bandet meningslöst. Försvaret har fått sina anslag minskade och museerna inom organisationen har inte prioritet.

Våra tre ABC800M hade flexskivestationer som hette ABC338. De hade skivor i storleken 8 tum och som rymde 1 MB. På våra datorerna kunde man köra CP/M. Detta operativsystem ansågs hopplöst långsamt i Sverige och hatades av alla för detta. Jag tyckte dock att det var rätt så bra, särskilt som programmen till det var så mycket kraftfullare än till ABC-en. På andra maskiner än just ABC var CP/M inte märkbart långsamt. Alltså måste detta irritationsmoment bero på en olycklig kombination av hård- och mjukvaror.

Under CP/M körde jag dBASE II och Wordstar (också hatat), gick på en dyr, fin kurs med utsökta luncher och hade trevliga lärare och kamrater. Dataverksamheten ansågs - och anses nog fortfarande - som alltför privilegierad på många ställen. I museivärlden är den också dåligt betald, men det är en annan historia.

Vårt CP/M råkade ut för något som antagligen var ett spontant virus. Jag hade ingen kommunikation från CP/M annat än med skivor. Efter en tid försvann allting från systemskivans bibliotek. Om jag använde de konstiga skivorna för att skapa nya systemskivor kunde de nyas data också försvinna med tiden. Trots att jag formaterade nya skivor med systemspår på försvann material. Jag var tvungen att gå tillbaka

till originalskivorna för att göra nya system- och brukskivor. Hur många källkodsfiler för bild databasen som måste skrivas om vågar jag inte tänka på.

MyAB var leverantör av CP/M för ABC-datorerna. De hade kontor i Solna - eller var det Sundbyberg? - men också i Göteborg. Efter lång, orofylld väntan fick vi det program som skulle flytta data från ABC till CP/M. Först då kunde vi flytta data från museet i Stockholm till stordatorn hos Medicindata i Göteborg.

Senare kom också Luxor med CP/M till sina maskiner. Skivorna var inte kompatibla med varandra. Ännu senare lade Luxor ner sin CP/M-verksamhet och överlät den till MyAB som gjorde rutiner med vilka man kunde hantera både de egna och Luxors skivor.

MyAB tillverkade också någon egen dator och en tillsats till Luxors ABC80 och ABC800. Dessa tillsatser förvandlade ABC-erna till PC-datorer. Tillsatserna var så billiga att de kunde konkurrera med nya PC men då priserna på datorerna från olika asiatiska länder började falla, kunde man inte fortsätta med sin produkt.

Vårt rykte spred sig. Jag fick besök av en kvinnlig forskare som ämnade doktorera på olika namn på filmjolk inom det nordiska språkområdet. Hon arbetade med gamla uppgifter men med datateknik ville hon bearbeta materialet till dittills oanade vetenskapliga höjder. Hon ämnade arbeta med dBASE II på en CP/M-dator. Vi resonerade länge och jag fick lära mig mer om dBASE. Många år senare - strax innan jag skrev den här artikeln - åhörde jag ett föredrag där hon presenterade sin forskning. Hon arbetade nu med dBASE IV på en PC. Hennes avhandling var klar och jag kunde studera hur hon gjort.

I museet fick biblioteket en programvara från Statskontoret för utskrift av katalogkort och nyförvärvslistor. Systemet används ännu. Det fungerar utmärkt och vållar inga andra problem än att jag måste formatera nya skivor eftersom de gamla blir fulla ibland. Programmets upphovsman är medlem i ABC-klubben.

Skrivare är ett viktigt tillbehör för många. En sådan tyckte jag att jag själv skulle ha nytta av och jag letade länge efter en billig. Jag fann en gammal Metric hos ELFA för 1500 kr med parallellkort. Skrivaren hade några mycket speciella g-bokstäver; g brukar normalt gå nedanför raden men Metric kortade av bokstaven så den rymdes ovan raden. Det deformerade g-et liknade i stället ett s vilket vållade mycket ilska hos en tidningsredaktion där någon sattes att för hand förtydliga mina g-en på utskriften.

Med TV-editorn, en texteditor för ABC80, skrev jag långa artiklar för en förenings-tidskrift. Editorn hade mycket som tog bort skrivmaskinens då ständiga problem. Man kunde rätta utan att kladda, man kunde lägga in textrader och radera utan att ta till kopian. Den enkla och idag antagligen tämligen trögrbetade ordbehandlaren gjorde arbetet till en fröjd och jag upplevde här ett skrivande där skrivmaskinens fasor var långt borta. Det medförde att jag såg möj-

ligheter att uttrycka mig i skrift på ett helt nytt, kreativt, snyggt och roligt sätt.

TV-editorn ersattes, eller kanske riktigare kompletterades, snart av TED, skriven av Anders Franzen. Han tyckte att TV-editorn var långsam och själv behövde han en editor för sin programmering. Jag tänkte att en möjlighet att ta in text till TED direkt från modem skulle vara bra och så kom TEDTERM också.

Under 1984 köpte jag en större flexskive-enhet till min ABC80. Den hette ABC830 och var en bamse tillsammans med kortlådan som hörde till. Skivorna rymde nu 150 kB. Den stora lådan gjorde att skrivarens kabel blev för kort. I MSG:s annonser fann jag ett billigt parallellkort med löstagbar och lång kabel. Säljaren, Stefan Berg, var mycket vänlig och hjälpte mig in i Unga Forskare någonstans. De hade konto i Digitalis kunddator på kvällarna. Tillsammans med VT100-emulatoren TERM100 lärde jag mig EMACS och började sammanställa verkligt långa dokument samt arbetade med fönster i ordbehandlaren för första gången. Jag kom också i kontakt med ett företag i Stockholm, Paralog, som skrev en av de kraftigare informationssystemen för DEC10 (Digitalis äldre stordator) och VAX (nyare). Dessa programvaror, kallade 3RIP respektive TRIP, kunde söka igenom en museidata-bas om 70000 poster på någon sekund. Det skulle vi ha! Det blev aldrig så. Problemet var och är stordatorn och kostnaden.

Jag skaffade ett konto i Datavisionen genom att låtsas vara näringsidkare i en affär som min pappa drev eftersom inte privatpersoner fick egna konton. De tekniska frågorna ställde jag från arbetet och därför fick jag som kontonummer telefonnumret till mitt arbete.

Datavisionen var främst dyr. Det kostade att ha möjligheten och det kostade att koppla upp sig. Det kostade också att betala de vanliga markeringarna. Innehållet var sämre och grafiken meningslös samtidigt som kommandospråket var unikt och besvärligt. Man kunde visserligen se att det fanns hus till salu på höglandet i Småland och se en grov fyrkant som beskrivning samt få veta hur vädret i Stockholm skulle bli igår men det tyckte jag inte var roligt. Att ha hela landets telefonkatalog till hands var nyttigare och även att kunna se aktuell portotabell. Att glänsa hos någon som hade premieobligationer med datum för de kommande dragningarna samt visa dagens börs-kurser kunde också vara kul.

Datavisionen kom att administreras från två system. Ett fanns i Göteborg och ett i Stockholm. När man sedan bytte system och användarna fick nya kontonummer med fruktsamt många tecken - omöjliga att memorera - gav jag upp. Jag sade upp kontot.

I en Cyber kunde jag också logga in. Det var en gräsligt omständlig procedur men jag kunde publicera några artiklar. Ofta glömde jag hur man skulle göra och måste ringa och fråga hur det skulle gå till.

Till min utökade ABC80 kunde man hantera operativsystemet CP/M och jag kopierade någonstans i Försvaret dBASE II till små skivor. Det fanns verkligen ABC800-

system med både små och stora skivor men de var sällsynta. Att skaffa ett själv för hobbyändamål var otänkbart. Man fick leta reda på andra som hade sådana resurser.

På mitt CP/M-system körde jag sedan i årtal mitt bokregister. Registret innehåller nu mera än 850 volymer och det kommer att bli större. Data kunde jag flytta till PC med program som jag funnit i ABC-klubbens programbank.

Naturligtvis har jag i ABC80-mod kört KERM, TERM100, ASS och andra fina produkter ur klubbens rikhaltiga programbank. En lustig sak var ett program som gav ABC80 ett antal toner. De hördes då man tryckte på olika tangenter på tangentbordet. Ljuden kunde skapas först då man kopplat samman några stift i den så kallade V24-porten. Ett böjt gem fick göra tjänst som brygga.

Ett nytt program kunde så småningom ersätta T80PRT på ABC80. Klubben hade fått en terminalemulator med namnet TERM100. Det hade den trevliga egenskapen att medge loggning på fil av inkommande och även utgående text. Man kunde sända en lokalt skriven text vilket minskade kostnaderna avsevärt. Om man kopplade upp sig mot en VAX kunde man få full skärmhantering med fönster och annat skoj. I olika VAX-system kunde jag konferera med många andra samtidigt. Var och en hade ett skikt av skärmen och alla kunde se vad alla skrev.

Att köra sådant som skärmhantering blir olidligt i 300 Baud. Den nya hastigheten split speed (1200 vid egen mottagning och 75 vid egen sändning) gjorde det hela njutbart. Sådana modem kom i marknaden till anständiga priser och jag köpte flera Selic modell 16 som jag fortfarande använder.

Ett annat uppriggande system fanns i Åkersberga och hette Permoba. Det drevs av Permobil, ett företag som tillverkar motoriserade rullstolar m m. I Permoba deltog jag i långa reseromang i olika frågor av alla slag och underhöll en omfattande korrespondens med någon som kallade sig Cilla Shinobu, senare Cilla Shadow. Denna kvinna var den enda som fick fortsätta vara anonym när Permoba sedan ville ha betalt för sina tjänster. Många skällde på henne, men jag fann hennes tankar väl genomtänkta och alltid intressanta. Jag fick lära mig om klärvoajans (från franskans ord clairvoyance, inre klarsyn, fjärrsyn) och annat som knappast någon annan modern människa vill tro på.

Varför ska man egentligen motsätta sig konstiga tankar som inte skadar? Jag har den åsikten att man inte bör avskära sig från kunskap i kontroversiella frågor - det kan finnas viktig, riktig kunskap och färdigheter där. Det är ju snarare inom de okända områdena som man kan göra framsteg.

Vid den tiden gjorde jag också ett program som skulle kontrollera om man viljemässigt kunde styra slumptalen i en Epson HX20-maskin. Jag fann att jag inte kunde styra talen men att jag ganska bra kunde gissa de tal som redan hade beräknats i maskinen. Det var ett märkligt resultat som gav vid handen att det möjligen kunde finnas en viss klärvoajans. Inom denna diskutabla vetenskap lär också finnas resultat som

tyder på att det klarskådade materialet flyttar sig till en annan "dimension" - i stället för det nyss framslumpade, dolda materialet träffar klarsynen de dessförinnan framslumpade talen och så vidare.

HX20 tillverkades av Epson och var en portföljdator. Jag fann en billig i en butik och kompletterade med en liten bandspelare som monterades fast vid datorn. Skärmen var liten med fyra rader och 20 teckens längd samt var av typen flytande kristall. Datorn hade MS-BASIC med ett antal egna instruktioner. Datorn skapade och underhöll ett bibliotek i början av mikrokassetten vilket gjorde det lätt att hitta filer på bandet. En annan lustig sak var att man kunde lägga en instruktion som slog av datorn i ett program. Jag fann också en tysk assembler som jag översatte och lekte lite med. Jag skaffade ett kommunikationsprogram för modemtrafik. Även ett program för registrering av museiföremål samt överföring av data till ABC800 skrev jag. Detta paket kom dock aldrig till användning. HX20 var annars intressant och hade en dokumentation som var större än datorn. En tjock A4-pärm fick jag från Luxor.

Senare försvann Permobas också. Våra Login lämnades över till en annan dator, jag minns inte vad den hette - var det CompuText? Jag tyckte aldrig om det systemet. Det var på något vis en annan stämning där - om man nu kan tala om stämning i ett datasystem. Jag körde bara några gånger och slutade sedan. Att lämna över användarnas identiteter var en åtgärd som väckte stort rabalder. Man använde databaserade uppgifter i ett annat sammanhang än det som uppgifterna insamlats för och det torde förbli en tveksam åtgärd så länge man inte har tillstånd för det.

MSG i klubbens datasystem betydde och betyder mycket för mig. I MSG kom jag i kontakt med andra likasinnade hackers. I MSG påträffades en ny arbetsmarknad. I MSG fick jag veta att en statsminister hade mördats - först senare läste jag tidningen, egentligen för att bara kontrollera uppgiften i klubbens konferenssystem. Jag trodde mera att det var utslag av dåligt skämtlynn. I MSG blev jag senare beskydd för "att på sätt och vis ha mördat Olof Palme".

Många andra saker diskuterades, en del triviala, en del väsentliga. Heter det "formatera" eller "formattera"? Den och frågan om det heter "ett modem" eller "en modem" diskuterades utförligt och länge på den tiden. I dag väcker frågorna bara löje men då var de viktiga.

När det första MSG konstruerades fick jag förtroendet att testa det. Genom en oväntad manöver gav jag ett kommando som inte var fullt implementerat. Systemet reagerade med en hängning och några dagar senare, då systemet var uppe igen, fick jag ett brev från Bo Kullmars som undrade vad jag hade gjort. Det mindes jag mycket väl, och så blev den buggen fixad.

Sedan fick jag testa Bo Kullmars DS90 och den lyckades jag också sänka under mina övningar med UNIX. Hur jag gjorde minns jag inte, men jag tror att jag ville ha ut något på ägarens radskrivare. Jag

kopierade därför till en, som jag trodde, enhet. Denna hette "printer" men visade sig nog vara ett program. I Bo Kullmars trevliga maskin körde jag sedan dess ABC-liknande BASIC. Vårt registerprogram konverterades och gick sedan utmärkt att köra på denna dator. Tanken var naturligtvis att museet skulle skaffa sig en egen DS90. Det kom aldrig längre än så. Basdatorer är dyra och investeringen måste slås ut på flera år. Detta förhållande gör nog att statsförvaltningen ofta blir utan stordatorstöd. En förvaltning som mycket väl behöver fleranvändarsystem blir utan eftersom pengar inte kan flyttas från ett budgetår till ett annat.

I MSG hade vi heta resonemang om ABC-klubben skulle behandla PC-maskiner i sin programbank. I dag väcker väl den frågan inte något intresse. Klubben skulle inte ha överlevt utan PC:n eftersom knappast någon numera utvecklar något för en ABC. Jag tror ändå att många behåller sina ABC-er och kör dem som terminaler från någon annan plats eller i någon speciell uppgift. Själv har jag en ABC80 just som terminal.

ABC80 var en mycket duglig maskin. Den slog många andra i hastighet, dess dokumentation var riklig och kunskandet om den var stort och spritt. Men man måste inse att det har kommit andra utrustningar som kan mer, har bättre program, kan hantera mera data och allt annat som är viktigt. ABC80 var en svensk maskin, väl känd av svenskar. PC-familjen är känd i hela världen, låt vara att allt skrivs på engelska språket. Egentligen har vi vunnit på bytet. Hackerns värld har blivit större samtidigt som den har blivit mindre genom att många fler förstår varandra.

Jag saknar inte ABC80 men kanske tiden som den representerade. Det var en tid av småpulsande, av begriplighet och småskalighet. Maskinen hade som sagt ett primärminne om 16 kB, nu har väl alla åtminstone 640 kB. Bandet har bytts ut mot diskett redan från början. Skärmen har minst 80 tecken och är ofta i färg. Om ABC-n skulle gå sönder så bytte man en krets. Numera byter man hela kortet eller hela datorn.

Programmen går inte att jämföra. Det finns allting till PC. ABC80 besvärades av litet RAM-minne, sekundärminnen som rymde för litet och var långsamma, frånvaron av en riktig V24-kontakt och ingen parallellport m m. Datorn hade en skärm om 40 tecken, hade måttlig grafik samt saknade funktionstangenter. Datorn var ändå trevlig och begriplig för dem som kom att arbeta med den. Dess begränsade storlek men kraftfulla programmeringsmöjligheter och rikhaltiga litteratur gjorde att många lyckades krama nästan det omöjliga ur den.

Det är svårt att tänka sig hur min verksamhet skulle ha kunnat utformas utan ABC-klubben. Den ofta dagliga kontakten med andra och alla program, ideer och tips som strömmar fram och tillbaka - det kan nog bara en hacker förstå.

Det var också genom ABC-klubben som Refugen uppstod - eller kunde börja växa, för många av ideerna hade funnits förut.

För dem som inte var med på den tiden kan jag berätta att det handlar om museiregistrering med hjälp av gravt handikappade ungdomar som annars skulle vara utestängda från arbetsmarknaden. Huvudmän är Frälsningarmen och Armémuseum. Det här hände strax före midsommar 1985. Refugen har beskrivits i ABC-bladet nr 4 år 1986. Datorerna kom att spela stor roll för de handikappade som nu kunde skriva brev, uttrycka sig, arbeta. För museet betyder det att ett viktigt arbete kan bli utfört. Samhällets vårdkostnader blir mindre eftersom långvården annars skulle vara det enda alternativet.

De senaste åren har jag varit upptagen med att få igång Refugen, bygga datasystem med Clipper-kompilatorn och att sätta mig in i Stockholms Läns Landsting och dess omsorgsverksamhet. Denna min politiska verksamhet är också en direkt förlängning av ABC80. Det var nämligen i Refugen jag kom i kontakt med en politiker och han föreslog att jag skulle arbeta vidare politiskt. Även i den verksamheten arbetar man med fördel med datorer. Datorerna är perfekta för att sammanställa material från olika håll och från olika tidpunkter. Politik kräver eftertanke och långa förberedelser samt noggrannhet. Texter kan byggas upp genom att skilda dokument sammanföres. Flera politiker kan arbeta hemma och ta med materialet till gemensamma träffar där det sammanställs.

Varje dag kommer jag i kontakt med datorer. Efter de här tio åren kan jag se att datorn har kommit in på alla områden och det är inte lika krångligt och förenat med samma rädsla hos medmänniskorna. Det går mycket lättare att få saker genomförda, mycket beroende på ett vidgat synsätt hos användarna men också på att systemen numera är bättre, kan mera och är utformade för att passa för dem som ska arbeta med datorn. Vi som ska hantera dem har bättre verktyg och driftsäkrare utrustning, längre erfarenhet och möjligheter att i konferenssystemen snabbt öka våra kunskaper om det vi behöver kunna.

ABC-klubben har betytt mycket för mitt arbete, min fritid och mina kunskaper om datorer, program och tillbehör. Med den här lilla återblicken vill jag tacka ABC-klubben och alla dess dåvarande och nuvarande medlemmar samt önska dem och kommande datorintresserade lycka till i fortsättningen.

<913>

Bo Michaelsson

Ny jättetävling!

Ett svårt

Under semestern hade jag långtråkigt. Visst var det fint väder och visst var det så mycket trivsamma saker man kunde göra (och gjorde) tillsammans med familj och vänner - men: jag har i många år varit van vid att ägna en del roliga timmar åt något programproblem i min dator.

Det är verkligen ödets ironi! Förr (på ABC80-tiden) fordrades det åtskillig övertalning för att förmå omgivningen att acceptera att utrymme skulle prioriteras för minst fyra ganska skrymmande plåtlådor med tillhörande nystan av kablar - både i bilen och i sommarhemmet. Grejerna måste ju stå framme hela tiden, när man väl med stort besvär kopplat ihop dem och fått allt att fungera. Men nådens sol lyste inte alla dagar. Skulle vi ha främmande eller extra nattgäster, då var det bara för mig att plocka ned alltsammans igen och gömma undan någonstans. Att jag hann med att skriva någonting alls under dessa förhållanden är en gåta, men ABC-bladets höstspalter genom åren bär omisskännligt vittnesbörd om att det faktiskt lyckades.

Nu är hela datoranordningen en enda liten burk, inte så mycket större än ABC80s tangentbord. Alla tillbehör, inklusive manualer och inspirationslitteratur, utan vidare rum i en ordinär portfölj som är mycket lättstuvad och knappast märks bland allt det övriga. Att plocka upp hela härligheten och komma i gång med något jobb är en bagatell - det som tar tid är att vänta medan datorn räknar sina tår och frågar sig vad den heter...ja, ni vet, den där "bootningen"...

Men nu har jag alltså ingenting att skriva på eller göra! Delvis är väl detta ett utslag av Alexander-syndromet (ganska målade beskrivet i ett tidigare ABC-blad (1987:1, 1988:4-5) Alla uppdämda behov, allt ackumulerad nyfikenhet, all upptäckarglädje och äventyrlusta har genom åren fått sitt utlopp och det finns inga nya världar att erövra - i varje fall inga nya verk att tillräckligt lockande för att löna mödan.

Väl är jag icke den förste att uppleva detta. Alexander den store har redan apotroferats. Under några sköna veckor i Dalom var jag på jubileumsutställningen i Zornmuseet i Mora. Utställningskatalogen (395kr!) är en diger sak, drygt A4-format och en halv decimeter tjock. Den innehåller inte bara förträffliga reproduktioner av alla Zorns centrala verk, utan också åtskilliga recensioner och citat ur korrespondens och tidningsartiklar.

Zorn:

"Jag lider av oro för att det är slut med mig som målare, att jag inte kan åstadkomma något, som är verkligt bra. Och det värsta för mig är, att när jag ser tillbaka på mitt verk, så är jag rädd för att jag som inbillat mig vara maskulin, i själva verket varit neutral, att jag flutit på lättköpta effekter som gjort sig hos den följsamma massan."

Nå, hmmm, "maskulin", tja, man skall väl inte dra jämförelsen för långt. Men jag är tydligen i gott sällskap...

Zorn föreföll att ha sökt en nytändning i att avbilda nakna damer i (för kritikerna) allt mer utmanande poser. Den tidens "vänner av ordning" anmärkte på att hans modeller var för nakna, för detaljerat avbildade, för feta och för många... Hade han inte förresten målat tillräckligt med naket redan tidigare? Var det kanske andra saker som låg bakom hans nyuppflyttade intresse för aktnålverket?

En kritiker skrev:

"Zorns popularitet har på ett farligt sätt gått hans konst i blodet, slappat självkritiken och underminerat den ideella grund all konst bör byggas på. Den vårdslösa nonchalans och kärlekslösa likgiltighet, som stöter oss i många av hans dukar från de senare åren, är barn av hans samvetslöshet med den breda publiken..."

Det som framför allt tycktes reta etablissemanget var att den stora allmänheten visade ett stort intresse för Zorns tavlor och att vykort reproduktionerna av nakenmotiven hade en strykande åtgång... Ajabaja. I Tyskland gjordes rent av ett beslag av sådana vykort, med motiveringen att ett konstnärligt verk kunde få pornografiska kvaliteter när den togs ned från galleriväggen och spreds i massupplaga bland folket...

Men - utan några jämförelser för övrigt - jag fattade mod och tänkte: Det som dög åt Zorn kan väl duga för mig. Så tog jag fram ett gammalt problem, som jag redan behandlat många gånger och försökte hitta en ny infallsvinkel.

PROBLEMET

I all korthet kan problemet beskrivas så: Hur får man datorn att lägga patiens?

Detta är inte så tråkigt som det kanske låter. Det är inte meningen att skapa ett automatspel som datorn spelar med sig själv. Det finns ett antal patiens ingalunda är i förväg bestämda, utan där operatören kan välja mellan olika strategier och olika drag inom varje strategi.

Jag tänker särskilt på två av dem: XXX och YYY - den förra en gång beskriven i Scientific American som ett intrikat problem, det andra något enklare, men med vissa individuella inslag. För den som inte känner till dem kommer här en kort beskrivning.

I båda fallen lägger man ut korten i 8 kolumner, med korten litet förskjutna så att man kan se färg och valör på alla korten:

1	2	3	4	5	6	7	8	H	S	R	K
9	10	11	12	13	14	15	16				
17	18	19	20	21	22	23	24				
25	26	27	28	29	30	31	32				
osv										
31	32	33	34	35	36	37	38				
49	50	51	52					A	B	C	D

I XXX har man ytterligare 4 högar med högst ett kort i varje, där man kan lägga udda kort som man tillfälligt vill bli av med: kalla dem A-D. De fyra högarna H S K R är givetvis plats för hjärter, spader, ruter och klöver. Det gäller att bygga de fyra senast nämnda högarna från ess till kung. De kort man kan flytta skall antingen ligga nederst i en kolumn, eller finnas i A-D. Nederst i kolumnerna kan man vidare bygga bakvända sviter i samma färg - t ex kan man lägga en klöver fyra på en klöver fem osv.

YYY fungerar litet annorlunda. A-D finns inte med och de fyra översta korten i de fyra första kolumnerna är från början med baksidan upp. Sviter byggs längst ned i kolumnerna. Om sista kortet i en kolumn är klöver fem, och om klöver fyra finns bland de synliga korten, lägger man fyra på femman, men samtidigt följer alla korten nedanför fyran med till den nya högen. De omvända korten får vändas och kommer med i spelet först när de blir fria i slutet på resp kolumn. Om en kolumn blir tom får man flytta dit en kung, tillsammans med ev nedanföriggande kort i kungens gamla kolumn.

problem

UPPGIFTEN

för denna övning är att:

- skapa en kortlek
- blanda den
- lägga ut den i 8 kolumner som ovan

Om då något av essen ligger nederst, skall det kortet flytta ut till sluthögen, och det kan möjligen datorn få göra automatiskt. Om sedan tvåan i samma färg ligger nederst kan manövern upprepas. Men när inga ytterligare drag är självskrivna, då skall operatören-spelaren in och "peka på" vilket kort som skall flyttas och vart det skall ta vägen. Det är nämligen inte säkert att man alltid skall bygga de sviter som går att bygga. i XXX kan det löna sig bättre att lägga kort på sidohögarna A-D. I YYY kan man få olika resultat beroende på vilken kung man väljer att lägga i en tom kolumn osv.

Alltså (för XXX):

- OM nästa kort på H-R-högarna finns nederst på kolumn, flytta den till rätt sluthög, ANNARS begär uppgift om vilket kort som skall flyttas och vart
- utför flyttningen och gå till d) så länge det finns kort kvar.

ÄR DET SVÅRT?

Att skapa en kortlek är inte så svårt. Det har beskrivits ett antal gånger i litteraturen och även figurerat i Bladet (1981:4-5, 1982:2).

I modernare programspråk som Pascal kan man definiera variablerna så att de kan se ut hur som helst

```
Var    färg = H,S,R,K
       nummer = ess,1-10,Kn,D,K
```

varefter man sätter ihop korten med färg + nummer (ordet VALör kanske kan användas i pascal, men i basic finns ett reserverat ord VAL, som bekant).

En annan metod är att börja med att skapa korten som numeriska enheter. Låt 100 betyda hjärter. Då finns 101-113 i hjärter. På samma sätt kan 200, 300 resp 400 vara spader, ruter och klöver. Detta har fördelen att datorn lätt håller reda på kortordningen.

(Om man - å ytterligare en annan sida - inte låter datorn göra några som helst kortflyttning på egen hand, liksom man avstår från att låta datorn kontrollera att dragen utförs korrekt enligt reglerna, så spelar den logiska ordningen bland korten inte någon roll.)

För att hålla reda på den samlade korthögen, kan man lägga dem i en vektor (array) A(1..52).

ATT BLANDA

är inte heller svårt när man väl har A(52). Dra ett slumpstal x mellan 1 och 52 och lägg A(x) i B(1). För att bli av med A(x) kan man flytta A(52) till läge a(x). Nästa gång drar man ett slumpstal y mellan 1 och 51 och gör om manövern med A(y) och B(2) tills alla korten ligger i B(1..52).

Så där, nu var den lätta delen klar! För den som inte vill bry sig med de små problemen finns här intill (eller på annan plats i detta nummer (Eller nästa! sätt:s anm)) ett förslag till programkod i basic för ABC-datorerna. Där finns också ett förslag till rutin som översätter siffer-korten till litet mera konventionella beteckningar t ex H 6 i stället för 106.

ATT LÄGGA UT

Men nu börjar det svåra. Om uppgiften bara vore att lägga ut korten skulle man lätt kunna göra det genom att ta dem ut B(52) och placera ut dem en rad i taget. Efter vart åttonde kort gör man radbyte. Ett förslag till programskrivning finns med i ovannämnda programbit.

Nu börjar problemen hopa sig.

Hur "pekar" man på det kort man vill flytta? Hur vet datorn vilket kort det gäller? På något sätt måste datorn minnas hur korten låg.

Det finns säkert flera lösningar på detta. En metod är att placera korten i en matris C(Rad,Kolumn). Det går bra för den 6 första raderna, men hur gör man med de följande? Vid första utläggningen finns kort bara i de fyra första kolumnerna i rad 7. Genom uppbyggnad av sviter kan, i varje fall i princip, 51 kort i samma färg ligga på varandra nederst i en hög. Hur stor matris skall man reservera? Och hur gör man med alla tomma platser i matrisen?

När man benar upp ett problem börjar man förstås med att tänka sig (i vanligt språk) vad det är man vill göra. Sedan kan det hjälpa att skriva ett förenklat språk som påminner mera om datorns sätt att arbeta, s k pseudoprogram.

Ett första försök till pseudoprogram kunde se ut så här:

Procedur MinnsSlutet !ta reda på vilket kort som ligger nederst

For Rad=1 to 8

For Kolumn=1 to 100 !100 är kanske litet högt

IF C(Rad,Kolumn)=0 !tom plats
THEN Slutkort(Kolumn)=C(Rad-1,
Kolumn)

Next Kolumn

Next Rad

därmed har vi tillgängliga slutkort i Slutkort(Kolumn). där Kolumn=1-8.

Procedur Autoflytt

Om något slutkort hör hemma på H,S,R,K,
så flytta dit kortet och låt nästa kort i kolumnen varas slutkort

Jaha, här kan man fundera över om möjligen HSRK-raden lämpligen vore numerisk så att man kunde avsöka den med en FOR-NEXT-rutin. Innan man bestämt det kommer man inte längre med pseudorutinen.

Och så här långt kom jag innan semestern tog slut. Varsågod nu, alla Stora och små programmerare: Bladet emotser en mängd smarta lösningar på patiensproblemet i alla tänkbara programspråk. Som belöning kommer de bästa lösningarna att kommenteras i Bladet. Vem vet - kanske kommer de rent av ut på någon av klubbens förträffliga programskivor. Vilken högre ära kan man eftertrakta i ABC-världen?

Sven Wickberg <1384>

DÖBELN

Resultat av tävlingen i 1990:3 "Är världen punktförmig?"

Ett enda bidrag strömmade in till redaktionen. Det kom från Anders Umegård och presenteras härintill i Turbo Pascal. Anders egen kommentar medföljer.

Programpaketet, inkluderande en version för färgskärm, finns i monitorn under namnet HOTCHPOT.ZIP, vilket skall påminna om ursprungliga beteckningen i Scientific Amerikan "The Hotch-potch Machine".

I det lilla format som den aktiva rutan i programmet har (24*24) får man ingen uppfattning om de mönster som utlovades. Anders själv säger att det blir mycket bättre om man ökar x och y i början av programmet till 40. Helst borde det nog vara ännu mer, men då går programmet troligen mycket långsamt på de flesta maskiner. Det blir väl inte bättre av att de ursprungliga författarna tänkte sig ett betydligt större antal möjliga värden än 3,1,1 som Anders rekommenderar.

Kanske programmet inte kommer till sin rätt om man inte har skärmar med betydligt mindre pixels så att upplösningen blir finare.

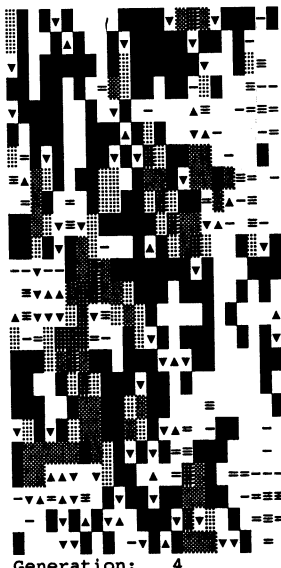
Det var i alla fall ett gott försök, som onekligen på nytt avslöjar det förbluffande i att vilket huller-om-buller-mönster man än börjar med, tycks det övergå i en allt högre grad av ordning.

Anders Umegård koras härmed - i brist på konkurrens :-)- till obestridd mästare i huller-om-buller-programmering.

Vi publicerar en av pascalversionerna och väljer då DOEBELN.PAS eftersom den inte förutsätter färgskärm och Anders egna kommentarer.

<1384>

Sven Wickberg



Generation: 4

```

program Doeblen; (* Version för PC *)
uses crt;

const x=24; (* Antal kolumner *)
      y=24; (* Antal rader *)
      n=9;  (* Antal sjukdomsstadier på cell *)
      sak : array(.0..n.) of (* Hur de olika nivåerna skall se ut *)
      char=(' ','█','▒','░',chr(31),#30,'=','-',',',' ');

var   xygrej,old : array(.0..x+1,0..y+1.) of real;
      g : real;    (* Här är tre *)
      k1,k2 : byte; (* Konstanter *)
      i,j : integer; (* Common-variabler *)
      varv : longint; (* Räkna hur många cykler datorn kört *)

procedure init;
var
  slask : string;
begin
  textmode(2+font8x8); (* Detta ger EGA 43 ch VGA 50 rader *)
  randomize;
  clrscr;
  writeln('DÖBELN eller Epidemispelet. Anders Umegård .. ABC-Klubben');
  write('Ange konstant k1 (',k1,')'); readln(slask);
  if slask<>' ' then val(slask,k1,i);
  write('Ange konstant k2 (',k2,')'); readln(slask);
  if slask<>' ' then val(slask,k2,i);
  write('Ange konstant g (',g,')'); readln(slask); (* g=0 avslutar *)
  if slask<>' ' then val(slask,g,i);
  (* Slumpa fram *)
  for i:=1 to x do
    for j:=1 to y do
      xygrej(i,j):=random(n+1);
  clrscr;
end;

procedure display(k,r : integer);
var nr : integer;
begin
  gotoxy(k,r);
  nr:=trunc(xygrej(k,r)); (* Här kan man ju skriva direkt i bild- *)
  write(sak(nr)); (* minnet om man vill öka farten *)
end;

procedure satta; (* I vilken noll-koordinater sätts *)
begin
  for i:=1 to y do
    begin
      old(.0,i):=old(x,i);
      old(x+1,i):=old(1,i);
    end;
  for i:=1 to x do
    begin
      old(i,0):=old(i,y);
      old(i,y+1):=old(i,1);
    end;
end;

procedure checka;
var k : integer;
  A : byte; (* Antalet smittade grannar *)
  B : byte; (* Antalet sjuka grannar *)
  S : real; (* Summan av tillståndsvärden *)
  iq : array(1..4.) of real; (* Värden på grannar *)

```

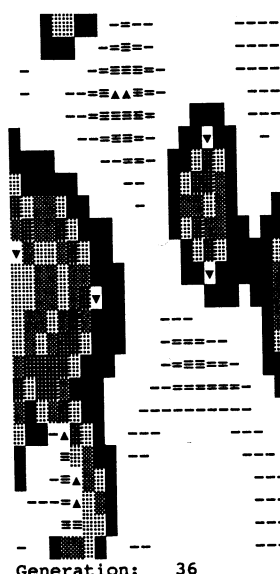
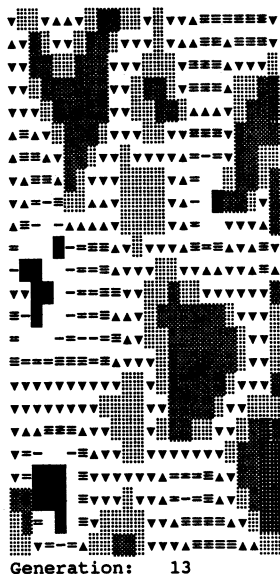
```

begin
  satta;
  for i:=1 to x do
    for j:=1 to y do
      begin
        iq(.1.):=old(.i-1,j.);
        iq(.2.):=old(.i+1,j.);
        iq(.3.):=old(.i,j-1.);
        iq(.4.):=old(.i,j+1.);
        A:=0;
        B:=0;
        S:=xygrej(.i,j.);
        for k:=1 to 4 do
          begin
            if iq(.k.)>n-1 then B:=B+1;
            if (iq(.k.)>0) or (iq(.k.)<n) then A:=A+1;
            S:=S+iq(.k.);
          end;
        if trunc(old(.i,j.))<1
          then xygrej(.i,j.):=trunc((A/k1)+(B/k2)) else
        if trunc(old(.i,j.))>n-1
          then xygrej(.i,j.):=0 else
        xygrej(.i,j.):=(S/A)+g;
        display(i,j);
      end;
    end;
  end;

  procedure kopiera;
  begin
    move(xygrej,old,sizeof(xygrej)); (* Tag "backup" på värden *)
  end;

  begin
    g:=1; k1:=3; k2:=1;
    while g>0 do
      begin
        textbackground(0);
        init;
        varv:=1;
        gotoxy(1,y+1); write('Generation:'); textcolor(15);
        if g>0 then
          repeat
            kopiera;
            checka;
            varv:=varv+1;
            textbackground(0); textcolor(15);
            gotoxy(15,y+1); write(varv);
            until keypressed
          end;
        textmode(2);
      end;
    end.

```



Västerås 1990-10-19

Hej!

Jag sänder här över DÖBELN som jag kallar programmet. Programmet ifråga är alltså det som står beskrivet i ABC-bladet §3 1990 i artikeln "Är världen punktförmig?"

DOEBELN.EXE

Programmet i kompilerad form.

DOEBELN.PAS

Programmets källkod.

DOEBELC.EXE

Samma fast i färg.

DOEBELC.PAS

Källkod till färgversion.

Programiden är ju faktiskt suverän! De mönster, eller avskilda områden utstrålar ibland en sällan skådad symmetri. Ibland uppträder sk. "demoner" det vill säga sådana demoner som vi känner igen från DEMON-programmen, bara det att de förra är mycket "rundare". Övriga jämförelser med DEMON-programmen är att DÖBELN tar mycket längre tid eftersom de matematiska beräkningarna är avsevärt mer avancerade.

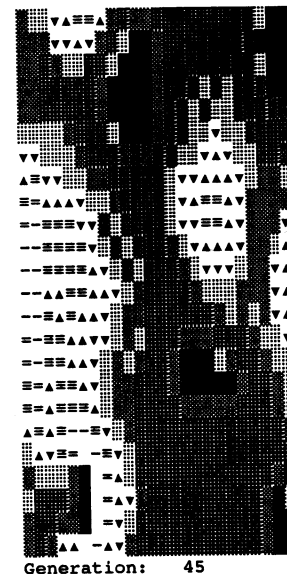
Ett litet tips är att (även om det tar längre tid) ha så stora värden på konstanterna X och Y (de sättes i programkoden). Har du EGA (eller VGA) så ta t.ex en matris på 40x40, då kommer programmet bäst till sin rätt.

Egna körningar har visat att bästa resultat inträffar med värden på 'k1','k2' och 'g' i storleksordningen runt 3,1 och 1.

Med vänlig hälsning

<4396>

Anders Umegård



Sextionio frågor och svar om C-programmering

Detta är en text som Steve Summit har publicerat i USENET om frågor och svar kring C-programmerig. /Bo Kullmar

Certain topics come up again and again on this newsgroup. They are good questions, and the answers may not be immediately obvious, but each time they recur, much net bandwidth and reader time is wasted on repetitive responses, and on tedious corrections to the incorrect answers which are inevitably posted.

This article, which will be reposted periodically, attempts to answer these common questions definitively and succinctly, so that net discussion can move on to more constructive topics without continual regression to first principles.

This article does not, and cannot, provide an exhaustive discussion of all of the subtle points and counterarguments which could be mentioned with respect to these topics. Cross-references to standard C publications have been provided, for further study by the interested and dedicated reader. A few of the more perplexing and pervasive topics may be further explored in some in-depth minitreatises posted in conjunction with this article.

No mere newsgroup article can substitute for thoughtful perusal of a full-length language reference manual. Anyone interested enough in C to be following this newsgroup should also be interested enough to read and study one or more such manuals, preferably several times. Some vendors' compiler manuals are unfortunately inadequate; a few even perpetuate some of the myths which this article attempts to debunk. Two invaluable references, which are an excellent addition to any serious programmer's library, are:

The C Programming Language, by Brian W. Kernighan and Dennis M. Ritchie.

C: A Reference Manual, by Samuel P. Harbison and Guy L. Steele, Jr.

Both exist in several editions. Andrew Koenig's book C Traps and Pitfalls also covers many of the difficulties frequently discussed here.

If you have a question about C which is not answered in this article, please try to answer it by referring to these or other books, or to knowledgeable colleagues, before posing your question to the net at large. There are many people on the net who are happy to answer questions, but the volume of repetitive answers posted to one question, as well as the growing numbers of questions as the net attracts more readers, can become oppressive. If you have questions or comments prompted by this article, please reply by mail rather than following up -- this article is meant to decrease net traffic, not increase it.

This article is always being improved. Your input is welcomed. Send your comments to scs@adam.mit.edu and/or scs@adam.mit.edu; this article's From: line may be unuseable.

Herewith, some frequently-asked questions and their answers:

Null Pointers

1. What is this infamous null pointer, anyway?

A: The language definition states that for each pointer type, there is a special value -- the "null pointer" -- which is distinguishable from all other pointer values and which is not the address of any object. That is, the address-of operator & will never "return" a null pointer, nor will a successful call to malloc. (malloc returns a null pointer when it fails, and this is a typical use of null pointers: as a "special" pointer value with some other meaning, usually "not allocated" or "not pointing anywhere yet.")

A null pointer is different from an uninitialized pointer. A null pointer is known not to point to any object; an uninitialized pointer might point anywhere (that is, at some random object, or at a garbage or unallocated address). See also question 34.

As mentioned in the definition above, there is a null pointer for each pointer type, and the internal values of null pointers for different types may be different. Although programmers need not know the internal values, the compiler must always be informed which null pointer is required, so it can make the distinction if necessary (see below).

References: K&R I Sec. 5.4 pp. 97-8; K&R II Sec. 5.4 p. 102; H&S Sec. 5.3 p. 91; ANSI X3.159-1989 Sec. 3.2.2.3.

2. How do I "get" a null pointer in my programs?

A: According to the language definition, a constant 0 in a pointer context is converted into a null pointer at compile time. That is, in an initialization, assignment, or comparison when one side is a variable or expression of pointer type, the compiler can tell that a constant 0 on the other side requests a null pointer, and generate the correctly-typed null pointer value. Therefore, the following fragments are perfectly legal:

```
char *p = 0;
if(p != 0)
```

However, an argument being passed to a function is not necessarily recognizable as a pointer context, and the compiler may not be able to tell that an unadorned 0 "means" a null pointer. For instance, the Unix system call "execl" takes a variable-length, null pointer-terminated list of character pointer arguments. To generate a null pointer in a function call context, an explicit cast is typically required:

```
execl("/bin/sh", "sh", "-c", "ls", (char *)0);
```

If the (char *) cast were omitted, the compiler would not know to pass a null pointer, and would pass an integer 0 instead. (Note that many Unix manuals get this example wrong.)

When function prototypes are in scope, argument passing becomes an "assignment context," and casts may safely be omitted, since the prototype tells the compiler that a pointer is required, and of which type, enabling it to correctly cast unadorned 0's. Function prototypes cannot provide the types for variable arguments in variable-length argument lists, however, so explicit casts are still required for those arguments. It is safest always to cast null pointer function arguments, to guard against varargs functions or those without prototypes, to allow interim use of non-ANSI compilers, and to demonstrate that you know what you are doing.

Summary:

unadorned 0 okay:	explicit cast required:
initialization	function call, no prototype in scope
assignments	variable argument to varargs function
comparisons	
function call, prototype in scope, fixed argument	

References: K&R I Sec. A7.7 p. 190, Sec. A7.14 p. 192; K&R II Sec. A7.10 p. 207, Sec. A7.17 p. 209; H&S Sec. 4.6.3 p. 72; ANSI X3.159-1989 Sec. 3.2.2.3.

3. But aren't pointers the same as ints?

A: Not since the early days. Attempting to push pointers into integers, or build pointers out of integers, has always been machine-dependent and unportable, and doing so is strongly discouraged. (Any object pointer may be cast to the "universal" pointer type void *, or char * under a pre-ANSI compiler, when heterogeneous pointers must be passed around.)

References: K&R I Sec. 5.6 pp. 102-3; ANSI X3.159-1989 Sec. 3.3.4.

4. What is NULL and how is it defined?

A: As a stylistic convention, many people prefer not to have unadorned 0's scattered throughout their programs. For this reason, the preprocessor macro NULL is defined (by `stdio.h` or `stddef.h`), with value 0 (or (void *)0, about which more later). A programmer who wishes to make explicit the distinction between 0 the integer and 0 the null pointer can then use NULL whenever a null pointer is required. This is a stylistic convention only; the preprocessor turns NULL back to 0 which is then recognized by the compiler (in pointer contexts) as before. In particular, a cast may still be necessary before NULL (as before 0) in a function call argument. (The table under question 2 above applies for NULL as well as 0.)

NULL should only be used for pointers. It should not be used when another kind of 0 is required, even though it might work, because doing so sends the wrong stylistic message. (ANSI allows the definition of NULL to be (void *)0, which will not work in non-pointer contexts.) In particular, do not use NULL when the ASCII null character (NUL) is desired. Provide your own definition

```
#define NUL '\0'
```

if you must.

References: K&R I Sec. 5.4 pp. 97-8; K&R II Sec. 5.4 p. 102; H&S Sec. 13.1 p. 283; ANSI X3.159-1989 Sec. 4.1.5 p. 99, Sec. 3.2.2.3 p. 38, Rationale Sec. 4.1.5 p. 74.

5. How should NULL be defined on a machine which uses a nonzero bit pattern as the internal representation of a null pointer?

A: Programmers should never need to know the internal representation(s) of null pointers, because they are normally taken care of by the compiler. If a machine uses a nonzero bit pattern for null pointers, it is the compiler's responsibility to generate it when the programmer requests, by writing "0" or "NULL," a null pointer. Therefore defining NULL as 0 on a machine for which internal null pointers are nonzero is as valid as on any other, because the compiler must (and can) still generate the machine's correct null pointers in response to unadorned 0's seen in pointer contexts.

6. If NULL were defined as follows:
`#define NULL (char *)0`

wouldn't that make function calls which pass an uncast NULL work?

- A: Not in general. The problem is that there are machines which use different internal representations for pointers to different types of data. The suggested definition would make uncast NULL arguments to functions expecting pointers to characters to work correctly, but pointer arguments to other types would still be problematical, and legal constructions such as

```
FILE *fp = NULL;
```

could fail.

Nevertheless, ANSI C allows the alternate

```
#define NULL (void *)0
```

definition for NULL. Besides helping incorrect programs to work (but only on machines with all pointers the same, thus questionably valid assistance) this definition may catch programs which use NULL incorrectly (e.g. when the ASCII nul character was really intended).

7. Is the abbreviated pointer comparison "if(p)" to test for non-null pointers valid? What if the internal representation for null pointers is nonzero?

- A: When C requires the boolean value of an expression (in the if, while, for, and do statements, and with the &&, &&, !, and ?: operators), a false value is produced when the expression compares equal to zero, and a true value otherwise. That is, whenever one writes

```
if(expr)
```

where "expr" is any expression at all, the compiler essentially acts as if it had been written as

```
if(expr != 0)
```

Substituting the trivial pointer expression "p" for "expr," we have

```
if(p)      is equivalent to      if(p != 0)
```

and this is a comparison context, so the compiler can tell that the (implicit) 0 is a null pointer, and use the correct value. There is no trickery involved here; compilers do work this way, and generate identical code for both statements. The internal representation of a pointer does not matter.

The boolean negation operator, !, can be described as follows:

```
!expr      is essentially equivalent to      expr?0:1
```

It is left as an exercise for the reader to show that

```
if(!p)     is equivalent to      if(p == 0)
```

See also question 48.

References: K&R II Sec. A7.4.7 p. 204; H&S Sec. 5.3 p. 91; ANSI X3.159-1989 Secs. 3.3.3.3, 3.3.9, 3.3.13, 3.3.14, 3.3.15, 3.6.4.1, and 3.6.5.

8. If "NULL" and "0" are equivalent, which should I use?

- A: Many programmers believe that "NULL" should be used in all pointer contexts, as a reminder that the value is to be thought of as a pointer. Others feel that the confusion surrounding "NULL" and "0" is only compounded by hiding "0" behind a definition, and prefer to use unadorned "0" instead. There is no one right answer. C programmers must understand that "NULL" and "0" are interchangeable and that an uncast "0" is perfectly acceptable in initialization, assignment, and comparison contexts. Any usage of "NULL" (as opposed to "0") should be considered a gentle reminder that a pointer is involved; programmers should not depend on it (either for their own understanding or the compiler's) for distinguishing pointer 0's from integer 0's. Again, NULL should not be used for other than pointers.

References: K&R II Sec. 5.4 p. 102.

9. But wouldn't it be better to use NULL (rather than 0) in case the value of NULL changes, perhaps on a machine with nonzero null pointers?

- A: No. Although preprocessor macros are often used in place of numbers because the numbers might change, this is not the reason that NULL is used in place of 0. The language guarantees that source-code 0's (in pointer contexts) generate null pointers. NULL is used only as a stylistic convention.

10. But I once used a compiler that wouldn't work unless NULL was used.

- A: This compiler was broken. In general, making decisions about a language based on the behavior of one particular compiler is likely to be counterproductive.

11. I'm confused. NULL is guaranteed to be 0, but the null pointer is not?

- A: A "null pointer" (written in lower case in this article) is a language concept whose particular internal value does not matter. (On some machines the internal value is 0; on others it is not.) A "null pointer" is requested in source code with the character "0". "NULL" (always in capital letters) is a preprocessor macro, which is always defined as 0 (or (void *)0).

When the term "null" or "NULL" is casually used, one of several things may be meant:

1. The conceptual null pointer, the abstract language concept defined in question 1. It is implemented with...
2. The internal (or run-time) representation of a null pointer, which may be different for different pointer types. The actual values should be of concern only to compiler writers. Authors of C programs never see them, since they use...
3. The source code syntax for null pointers, which is the single character "0". It is often hidden behind...
4. The NULL macro, which is defined to be "0" or "(void *)0". Finally, as a red herring, we have
5. The ASCII null character (NUL), which does have all bits zero, but has no relation to the null pointer except in name.

This article always uses the phrase "null pointer" for sense 1, the character "0" for sense 3, and the capitalized word "NULL" for sense 4.

12. Why is there so much confusion surrounding null pointers? Why do these questions come up so often?

- A: C programmers traditionally like to know more than they need to about the underlying machine implementation. The construct "if(p == 0)" is easily misread as calling for conversion of p to an integral type, rather than 0 to a pointer type, before the comparison. The fact that null pointers are represented both in source code, and internally to most machines, as zero invites unwarranted assumptions. The fact that a preprocessor macro (NULL) is often used suggests that this is done because the value might change later, or on some weird machine. Finally, the distinction between the several uses of the term "null" (listed above) is often overlooked.

One good way to wade out of the confusion is to imagine that C had a keyword (perhaps "nil", like Pascal) with which null pointers were requested. The compiler could either turn "nil" into the correct type of null pointer, when it could determine the type from the source code (as it does with 0's in reality), or complain when it could not. Now, in fact, in C the keyword for a null pointer is not "nil" but "0", which works almost as well, except that an uncast "0" in a non-pointer context generates an integer zero. If the null pointer keyword were "nil" the compiler could emit an error message for an ambiguous usage, but since it is "0" the compiler may end up emitting incorrect code.

13. I'm still confused. I just can't understand all this null pointer stuff.

- A: Follow these two simple rules:

1. When you want to refer to a null pointer in source code, use "0" or "NULL".
2. If the usage of "0" or "NULL" is in a function call, cast it to the pointer type expected by the function being called.

The rest of the discussion has to do with other people's misunderstandings, or with the internal representation of null pointers, which you shouldn't need to know.

Arrays and Pointers

14. I had the declaration `char a[5A]` in one source file, and in another I declared `extern char *a`. Why didn't it work?

- A: The declaration `extern char *a` simply does not match the actual definition. The type "pointer-to-type-T" is not the same as "array-of-type-T." Use `extern char a[5A]`.

15. But I heard that `char a[5A]` was identical to `char *a`.

- A: This identity (that a pointer declaration is interchangeable with an array declaration, usually unsized) holds only for formal parameters to functions. This identity is related to the fact that arrays "turn into" pointers in expressions. That is, when an array name is mentioned in an expression, it is converted immediately into a pointer to the array's first element. Therefore, an array is never passed to a function; rather a pointer to its first element is passed instead. Allowing pointer parameters to be declared as arrays is a simply a way of making it look as though the array was actually being passed. Some programmers prefer, as a matter of style, to use this syntax to indicate that the pointer parameter is expected to point to the start of an array rather than to a single value.

Since functions can never receive arrays as parameters, any parameter declarations which "look like" arrays, e.g.

```
f(a)
char a[5A];
```

are treated as if they were pointers, since that is what the function will receive if an array is passed:

```
f(a)
char *a;
```

To repeat, however, this conversion holds only within function formal parameter declarations, nowhere else. If this conversion confuses you, don't use it; many people have concluded that the confusion it causes outweighs the small advantage of having the declaration "look like" the call and/or the uses within the function.

References: K&R I Sec. 5.3 p. 95, Sec. A10.1 p. 205; K&R II Sec. 5.3 p. 100, Sec. A8.6.3 p. 218, Sec. A10.1 p. 226; H&S Sec. 5.4.3 p. 96; ANSI X3.159-1989 Sec. 3.5.4.3, Sec. 3.7.1.

16. So what is meant by the "equivalence of pointers and arrays" in C?

A: Perhaps no aspect of C is more confusing than pointers, and the confusion is compounded by statements like the one above. Saying that arrays and pointers are "equivalent" does not by any means imply that they are interchangeable. (The fact that, as formal parameters to functions, array-style and pointer-style declarations are in fact interchangeable does nothing to reduce the confusion.)

"Equivalence" refers to the fact (mentioned above) that arrays decay into pointers within expressions, and that pointers and arrays can both be dereferenced using array-like subscript notation. That is, if we have

```
char a[10];
char *p;
int i;
```

we can refer to `a[i]` and `p[i]`. (That pointers can be subscripted like arrays is hardly surprising, since arrays have decayed into pointers by the time they are subscripted.)

References: K&R I Sec. 5.3 pp. 93-6; K&R II Sec. 5.3 p. 99; H&S Sec. 5.4.1 p. 93; ANSI X3.159-1989 Sec. 3.3.2.1, Sec. 3.3.6.

17. My compiler complained when I passed a two-dimensional array to a routine expecting a pointer to a pointer.

A: The rule by which arrays decay into pointers is not applied recursively. An array of arrays (i.e. a two-dimensional array in C) decays into a pointer to an array, not a pointer to a pointer. Pointers to arrays are confusing, and it is best to avoid them. (The confusion is heightened by incorrect compilers, including some versions of pcc and pcc-derived lint's, which incorrectly accept assignments of multi-dimensional arrays to multi-level pointers.) If you are passing a two-dimensional array to a function:

```
int array[XSIZE][YSIZE];
f(array);
```

the function's declaration should match:

```
f(int a[XSIZE][YSIZE]) ...
```

or

```
f(int (*a)[YSIZE]) ...
```

In the first declaration, the compiler performs the usual implicit rewriting of "array of array" to "pointer to array;" in the second form the pointer declaration is explicit. The called function does not care how big the array is, but it must know its shape, so the "column" dimension `YSIZE` must be included. In both cases the number of "rows" is irrelevant, and omitted.

If a function is already declared as accepting a pointer to a pointer, an intermediate pointer would need to be used when attempting to call it with a two-dimensional array:

```
int *ip = &a[0][0];
g(ip);
...
g(int **ipp) ...
```

Note that this usage is liable to be misleading (if not incorrect), since the array has been "flattened" (its shape has been lost).

18. How do I declare a pointer to an array?

A: Usually, you don't want one. Think about using a pointer to one of the array's elements instead. Arrays of type `T` decay into pointers to type `T`, which is convenient; subscripting or incrementing the resultant pointer accesses the individual members of the array. True pointers to arrays, when subscripted or incremented, step over entire arrays, and are generally only useful when operating on multidimensional arrays. (See the question above.)

19. How can I dynamically allocate a multidimensional array?

A: It is usually best to allocate an array of pointers, and then initialize each pointer to a dynamically-allocated "row." The resulting "ragged" array often saves space, although it may not be contiguous in memory as a real array would be.

```
int **array = (int **)malloc(nrows * ncolums * sizeof(int *));
for(i = 0; i < nrows; i++)
    array[i] = (int *)malloc(ncolums * sizeof(int));
```

(In "real" code, of course, `malloc`'s return value should be checked.)

You can keep the array's contents contiguous, while losing the ability to have rows of varying and different lengths, with a bit of explicit pointer arithmetic:

```
int **array = (int **)malloc(nrows * ncolums * sizeof(int *));
array[0] = (int *)malloc(nrows * ncolums * sizeof(int));
for(i = 1; i < nrows; i++)
    array[i] = array[0] + i * ncolums;
```

In either case, the elements of the dynamic array can be accessed with normal-looking array subscripts: `array[i][j]`.

If the double indirection implied by the above scheme is for some reason unacceptable, you can simulate a two-dimensional array with a single, dynamically-allocated one-dimensional array:

```
int *array = (int *)malloc(nrows * ncolums * sizeof(int));
```

However, you must now perform subscript calculations manually, accessing `array[i, j]` with `array[i * ncolums + j]`. (A macro can hide the explicit calculation, but invoking it then requires parentheses and commas which don't look exactly like multidimensional array subscripts.)

Order of Evaluation

20. Under my compiler, the code

```
int i = 7;
printf("%d\n", i++ * i++);
```

prints 49. Regardless of the order of evaluation, shouldn't it print 56?

A: Although the postincrement and postdecrement operators `++` and `--` perform the operations after yielding the former value, many people misunderstand the implication of "after." It is not guaranteed that the operation is performed immediately after giving up the previous value and before any other part of the expression is evaluated. It is merely guaranteed that the update will be performed sometime before the expression is considered "finished" (before the next "sequence point," in ANSI C's terminology).

In the example, the compiler chose to multiply the previous value by itself and to perform both increments afterwards.

The order of other embedded side effects is similarly undefined. For example, the expression `i + (i = 2)` may or may not have the value 4. ANSI allows compilers to reject code which contains such ambiguous or undefined side effects.

References: K&R I Sec. 2.12 p. 50; K&R II Sec. 2.12 p. 54; ANSI X3.159-1989 Sec. 3.3.

21. But what about the `&&`, `&&`, `?:`, and comma operators?

I see code like "if(`c = getchar()`) == EOF `&& c == '0n'`)" ...

A: There is a special exception for those operators; each of them does imply a sequence point (i.e. left-to-right evaluation is guaranteed).

References: ANSI X3.159-1989 Secs. 3.3.2.2, 3.3.13, 3.3.14, 3.3.15.

ANSI C

22. What is the "ANSI C Standard?"

A: In 1983, the American National Standards Institute commissioned a committee, X3J11, to standardize the C language. After a long and arduous process, this C standard was finally ratified as an American National Standard, X3.159-1989, on December 14, 1989, and published in the spring of 1990. For the most part, ANSI C standardizes existing practice, with a few additions from C++ (most notably function prototypes) and support for multinational character sets (including the much-lambasted trigraph sequences for transfer of source code between machines with deficient or multinational character sets). The ANSI C standard also formalizes the C run-time library support routines, an unprecedented effort.

23. How can I get a copy of the ANSI C standard?

A: Copies are available from

American National Standards Institute
1430 Broadway
New York, NY 10018
(212) 642-4900

or

Global Engineering Documents
2805 McGaw Avenue
Irvine, CA 92714
(714) 261-1455

The cost is approximately \$50.00, plus \$6.00 shipping. Quantity discounts are available.

24. Does anyone have a tool for converting old-style C programs to ANSI C, or for automatically generating prototypes?

A: There are several such programs, many in the public domain. Check your nearest comp.sources archive. (See also questions 61 and 62.)

25. My ANSI compiler complains about a mismatch when it sees

```
extern int func(float);
int func(x)
float x;
...
```

A: You have mixed the new-style declaration "extern int func(float);" with the old-style definition "int func(x) float x;". Old C (and ANSI C, in the absence of prototypes) silently promotes floats to doubles when passing them as arguments, and makes a corresponding silent change to formal parameter declarations, so the old-style definition actually says that `func` takes a double.

The problem can be fixed either by using new-style syntax consistently in the definition:

```
int func(float x) { ... }
```

or by changing the new-style prototype declaration to match the old-style definition:

```
extern int func(double);
```

(In this case, it would be clearest to change the old-style definition to use double as well.)

References: ANSI X3.159-1989 Sec. 3.3.2.2.

C Preprocessor

26. How can I write a macro to swap two values?

A: There is no good answer to this question. If the values are integers, a well-known trick using exclusive-OR could perhaps be used, but it will not work for floating-point values or pointers. If the macro is intended to be used on values of arbitrary type (the usual goal), it cannot use a temporary, since it doesn't know what type of temporary it needs, and standard C does not provide a typeof operator. (GNU C does.)

The best all-around solution is probably to forget about using a macro. If you're worried about the use of an ugly temporary, and know that your machine provides an exchange instruction, convince your compiler vendor to recognize the standard three-assignment swap idiom in the optimization phase. Alternatively, use a language which supports multiple, parallel assignment (a,b := b,a).

27. I'm getting strange syntax errors inside code which I've #ifdeffed out.

A: Under ANSI C, the text inside a "turned off" #if, #ifdef, or #ifndef must still consist of "valid preprocessing tokens." This means that there must be no unterminated comments or quotes (note particularly that an apostrophe within a contracted word looks like the beginning of a character constant) and no newlines inside quotes. Therefore, natural-language comments should always be written between the "official" comment delimiters /* and */.

28. How can I write a cpp macro which takes a variable number of arguments?

One popular trick is to define the macro with a single argument, and call it with a double set of parentheses, which appear to the compiler to indicate a single argument:

```
#define DEBUG(args) fprintf("DEBUG: "); printf args;

if(n != 0) DEBUG(("n is %d\n", n));
```

The obvious disadvantage to this trick is that the caller must always remember to use the extra parentheses. (It is often best to use a bona-fide function, which can take a variable number of arguments in a well-defined way, rather than a macro. See questions 29 and 30 below.)

Variable-Length Argument Lists

29. How can I write a function that takes a variable number of arguments?

A: Use varargs or stdarg.

Here is a function which concatenates an arbitrary number of strings into malloc'ed memory, using stdarg:

```
#include <stddef.h>      /* for NULL */
#include <stdarg.h>      /* for va_stuff */
#include <string.h>      /* for strcat et al */
#include <stdlib.h>      /* for malloc */

extern char *malloc();   /* redundant */

/* VARARGS1 */

char *
vstreat(char *first, ...)
{
    int len = 0;
    char *retbuf;
    va_list argp;
    char *p;

    if(first == NULL)
        return NULL;

    len = strlen(first);

    va_start(argp, first);

    while((p = va_arg(argp, char *)) != NULL)
        len += strlen(p);

    va_end(argp);

    retbuf = malloc(len + 1); /* +1 for trailing 00 */

    if(retbuf == NULL)
        return NULL;

    (void)strcpy(retbuf, first);

    va_start(argp, first);

    while((p = va_arg(argp, char *)) != NULL)
        (void)strcat(retbuf, p);

    va_end(argp);

    return retbuf;
}
```

Usage is something like

```
char *str = vstreat("Hello, ", "world!", (char *)NULL);
```

Note the cast on the last argument. (Also note that the caller must free the returned, malloc'ed storage.)

Using the older varargs package, rather than stdarg, requires a few changes which are not discussed here, in the interests of brevity. See the next question for hints.

References: K&R II Sec. 7.3 p. 155, Sec. B7 p. 254; H&S Sec. 13.4 pp. 286-9; ANSI X3.159-1989 Secs. 4.8 through 4.8.1.3.

30. How can I write a function that takes a format string and a variable number of arguments, like printf, and passes them to printf to do most of the work?

A: Use vprintf, vfprintf, or vsprintf.

Here is an "error" routine which prints an error message, preceded by the string "error: " and terminated with a newline:

```
#include <stdio.h>
#include <stdarg.h>

void
error(char *fmt, ...)
{
    va_list argp;
    fprintf(stderr, "error: ");
    va_start(argp, fmt);
    vfprintf(stderr, fmt, argp);
    va_end(argp);
    fprintf(stderr, "\n");
}
```

To use varargs, instead of stdarg, change the function header to:

```
void error(va_alist)
va_dcl
{
    char *fmt;
```

change the va_start line to

```
va_start(argp);
```

and add the line

```
fmt = va_arg(argp, char *);
```

between the calls to va_start and vfprintf. (Note that there is no semicolon after va_dcl.)

References: K&R II Sec. 8.3 p. 174, Sec. B1.2 p. 245; H&S Sec. 17.12 p. 337; ANSI X3.159-1989 Secs. 4.9.6.7, 4.9.6.8, 4.9.6.9.

31. How can I write a function analogous to scanf?

A: Unfortunately, vscanf and the like are not standard. You're on your own.

32. How can I discover how many arguments a function was actually called with?

A: This information is not available to a portable program. Some systems have a nonstandard nargs() function available, but its use is questionable, since it typically returns the number of words pushed, not the number of arguments. (Floating point values and structures are usually passed as several words.)

Any function which takes a variable number of arguments must be able to determine from the arguments themselves how many of them there are. printf-like functions do this by looking for formatting specifiers (%d and the like) in the format string (which is why these functions fail badly if the format string does not match the argument list). Another common technique (useful when the arguments are all of the same type) is to use a sentinel value (often 0, -1, or an appropriately-cast null pointer) at the end of the list (see the vstreat and excol examples under questions 29 and 2 above).

33. How can I write a function which takes a variable number of arguments and passes them to some other function (which takes a variable number of arguments)?

A: In general, you cannot. You must provide a version of that other function which accepts a va_list pointer, as does vfprintf in the example above. If the arguments must be passed directly as actual arguments (not indirectly through a va_list pointer) to another function which is itself variadic (for which you do not have the option of creating an alternate, va_list-accepting version) no portable solution is possible. (The problem can often be solved by resorting to machine-specific assembly language.)

Memory Allocation

34. Why doesn't this program work?

```
main()
{
    char *answer;
    printf("Type something:\n");
    gets(answer);
    printf("You typed '%s'\n", answer);
}
```

A: The pointer variable "answer," which is handed to the gets function as the location into which the response should be stored, has not been set to point to any valid storage. It is an uninitialized variable, just as is the variable i in this example:

```
main()
{
    int i;
    printf("i = %d\n", i);
}
```

That is, we cannot say where the pointer "answer" points. (Since local variables are not initialized, and typically contain garbage, it is not even guaranteed that "answer" starts out as a null pointer.)

The simplest way to correct the question-asking program is to use a local array, instead of a pointer, and let the compiler worry about allocation:

```
#include <stdio.h>
main()
{
    char answer[100];
    printf("Type something:\n");
    fgets(answer, 100, stdin);
    printf("You typed '%s'\n", answer);
}
```

Note that this example also uses fgets instead of gets (always a good idea), so that the size of the array can be specified, so that fgets will not overwrite the end of the array if the user types an overly-long line. (Unfortunately, gets and fgets differ in their treatment of the trailing \n.) It would also be possible to use malloc to allocate the answer buffer, and/or to parameterize its size (#define ANSWERSIZE 100).

35. You can't use dynamically-allocated memory after you free it, can you?

A: No. Some early man pages for malloc stated that the contents of freed memory was "left undisturbed;" this ill-advised guarantee is not universal and is not required by ANSI.

Few programmers would use the contents of freed memory deliberately, but it is easy to do so accidentally. Consider the following (correct) code for freeing a singly-linked list:

```
struct list *listp, *nextp;
for(listp = base; listp != NULL; listp = nextp) {
    nextp = listp->next;
    free((char *)listp);
}
```

and notice what would happen if the more-obvious loop iteration expression listp = listp->next were used, without the temporary nextp pointer.

36. What is alloca and why is its use discouraged?

A: alloca allocates memory which is automatically freed when the function from which alloca was called returns. That is, memory allocated with alloca is local to a particular function's "stack frame" or context.

alloca cannot be written portably, and is difficult to implement on machines without a stack. Its use is problematical (and the obvious implementation on a stack-based machine fails) when its return value is passed directly to another function, as in fgets(alloca(100), stdin, 100).

For these reasons, alloca cannot be used in programs which must be widely portable, no matter how useful it might be.

Structures

37. I heard that structures could be assigned to variables and passed to and from functions, but K&R I says not.

A: What K&R I said was that the restrictions on struct operations would be lifted in a forthcoming version of the compiler, and in fact struct assignment and passing were fully functional in Ritchie's compiler even as K&R I was being published. Although a few early C compilers lacked struct assignment, all modern compilers support it, and it is part of the ANSI C standard, so there should be no reluctance to use it.

References: K&R I Sec. 6.2 p. 121; K&R II Sec. 6.2 p. 129; H&S Sec. 5.6.2 p. 103; ANSI X3.159-1989 Secs. 3.1.2.5, 3.2.2.1, 3.3.16.

38. How does struct passing and returning work?

A: When structures are passed as arguments to functions, the entire struct is pushed on the stack, which may involve significant overhead for large structures. It may be preferable in such cases to pass a pointer to the structure instead.

Structures are returned from functions either in a special, static place (which may make struct-valued functions nonreentrant) or in a location pointed to by an extra, "hidden" argument to the function.

39. The following program works correctly, but it dumps core after it finishes. Why?

```
struct list
{
    char *item;
    struct list *next;
};

/* Here is the main program. */

main(argc, argv)
{
    ...
}
```

A: A missing semicolon causes the compiler to believe that main returns a struct list. (The connection is hard to see because of the intervening comment.) When struct-valued functions are implemented by adding a hidden return pointer, the generated code tries to store a struct with respect to a pointer which was not actually passed (in this case, by the C start-up code). Attempting to store a structure into memory pointed to by the argc or argv value on the stack (where the compiler expected to find the hidden return pointer) causes the core dump.

40. Why can't you compare structs?

A: There is no reasonable way for a compiler to implement struct comparison which is consistent with C's low-level flavor. A byte-by-byte comparison could be invalidated by random bits present in unused "holes" in the structure (such padding is used to keep the alignment of later fields correct). A field-by-field comparison would require unacceptable amounts of repetitive, in-line code for large structures. Either method would not necessarily "do the right thing" with pointer fields: oftentimes, equality should be judged by equality of the things pointed to rather than strict equality of the pointers themselves.

If you want to compare two structures, you must write your own function to do so. C++ (among other languages) would let you arrange for the == operator to map to your function.

References: K&R II Sec. 6.2 p. 129; H&S Sec. 5.6.2 p. 103.

41. How can I determine the byte offset of a field within a structure?

A: ANSI C defines the offsetof macro, which should be used if available. If you don't have it, a suggested implementation is

```
#define offsetof(type, mem) ((size_t) 0
((char *)(&((type *) 0)->mem) - (char *)(&((type *) 0)))
```

This implementation is not 100% portable; some compilers may legitimately refuse to accept it.

See the next question for a usage hint.

References: ANSI X3.159-1989 Sec. 4.1.5.

42. How can I access structure fields by name at run time?

A: Build a table of names and offsets, using the offsetof() macro. The offset of field b in struct a is

```
offsetof(struct a, b)
```

If structp is a pointer to an instance of this structure, and b is an int field with offset as computed above, b's value can be set indirectly with

```
*(int *)(&((char *)structp + offset) = value;
```

Declarations

43. I can't seem to define a linked list node which contains a pointer to itself. I tried

```
typedef struct
{
    char *item;
    NODEPTR next;
} NODE, *NODEPTR;
```

but the compiler gave me error messages. Can't a struct in C contain a pointer to itself?

A: Structs in C can certainly contain pointers to themselves; the discussion and example in section 6.5 of K&R make this clear. The problem is that the example above attempts to hide the struct pointer behind a typedef, which is not complete at the time it is used. First, rewrite it without a typedef:

```
struct node
{
    char *item;
    struct node *next;
};
```

Then, if you feel you must use typedefs, define them after the fact:

```
typedef struct node NODE, *NODEPTR;
```

Alternatively, define the typedefs first (using the line just above) and follow it with the full definition of struct node, which can then use the NODEPTR typedef for the "next" field.

References: K&R I Sec. 6.5 p. 101; K&R II Sec. 6.5 p. 139; H&S Sec. 5.6.1 p. 102; ANSI X3.159-1989 Sec. 3.5.2.3.

44. How can I define a pair of mutually referential structures? I tried

```
typedef struct
{
    int structafield;
    STRUCTB *bpointer;
    A STRUCTA;
}

typedef struct
{
    int structbfield;
    STRUCTA *apointer;
    A STRUCTB;
}
```

but the compiler doesn't know about STRUCTB when it is used in struct a.

A: Again, the problem is not the pointers but the typedefs. First, define the two structures without using typedefs:

```
struct a
{
    int structafield;
    struct b *bpointer;
    A;
}

struct b
{
    int structbfield;
    struct a *apointer;
    A;
}
```

The compiler can accept the field declaration struct b *bpointer within struct a, even though it has not yet heard of struct b. Occasionally it is necessary to precede this couplet with the empty declaration

```
struct b;
```

to mask the declaration (if in an inner scope) from a different struct b in an outer scope.

Again, the typedefs could also be defined before, and then used within, the definitions for struct a and struct b. Problems arise only when an attempt is made to define and use a typedef within the same declaration.

References: H&S Sec. 5.6.1 p. 102; ANSI X3.159-1989 Sec. 3.5.2.3 .

45. How do I declare a pointer to a function returning a pointer to a double?

A: There are at least three answers to this question:

1. double *(*p)();

2. Build it up in stages, using typedefs:

```
typedef double *pd;      /* pointer to double */
typedef pd fpd();        /* func returning ptr to double */
typedef fpd *pfpd;       /* ptr to func ret ptr to double */
pfpd p;
```

3. Use the odecl program, which turns English into C and vice versa:

```
$ odecl
odecl> declare p as pointer to function returning pointer to double
odecl> *(*p)();
odecl>
```

odecl can also explain complicated declarations, help with casts, and indicate which set of parentheses the arguments go in (for complicated function definitions).

References: H&S Sec. 5.10.1 p. 116.

46. So where can I get odecl?

A: Several public-domain versions are available. One is in volume 14 of comp.sources.unix . (Commercial versions may also be available, at least one of which was shamelessly lifted from the public domain copy submitted by Graham Ross, one of odecl's originators.)

Boolean Expressions and Variables

47. What is the right type to use for boolean values in C? Why isn't it a standard type? Should #defines or enums be used for the true and false values?

A: C does not provide a standard boolean type, because picking one involves a space/time tradeoff which is best decided by the programmer. (Using an int for a boolean may be faster, while using char will probably save data space.)

The choice between #defines and enums is arbitrary and not terribly interesting. Use any of

```
#define TRUE 1          #define YES 1
#define FALSE 0         #define NO 0

enum bool {false, true; enum bool {no, yes;
```

as long as you are consistent within one program or project. (The enum may be preferable if your debugger expands enum values when examining variables.)

Some people prefer variants like

```
#define TRUE (1==1)
#define FALSE (!TRUE)
```

These don't buy anything (see below).

48. Isn't #defining TRUE to be 1 dangerous, since any nonzero value is considered "true" in C? What if a built-in boolean or relational operator "returns" something other than 1?

A: It is true (sic) that any nonzero value is considered true in C, but this applies only "on input", i.e. where a boolean value is expected. When a boolean value is generated by a built-in operator, it is guaranteed to be 1 or 0. Therefore, the test

```
if((a == b) == TRUE)
```

will succeed (if a, in fact, equals b and TRUE is one), but this code is obviously silly. In general, explicit tests against TRUE and FALSE are undesirable, because some library functions (notably isupper, isalpha, etc.) return, on success, a nonzero value which is not necessarily 1. A good rule of thumb is to use TRUE and FALSE (or the like) only for assignment to a Boolean variable or as the return value from a Boolean function, never in a comparison.

Preprocessor macros like TRUE and FALSE (and, in fact, NULL) are used for code readability, not because the underlying values might ever change. That "true" is 1 and "false" (and source-code null pointers) 0 is guaranteed by the language. (See also question 7.)

References: K&R I Sec. 2.7 p. 41; K&R II Sec. 2.6 p. 42, Sec. A7.4.7 p. 204, Sec. A7.9 p. 206; ANSI X3.159-1989 Secs. 3.3.3.3, 3.3.8, 3.3.9, 3.3.13, 3.3.14, 3.3.15, 3.6.4.1, 3.6.5 .

49. What is the difference between an enum and a series of preprocessor #defines?

A: At the present time, there is little difference. Although many people might have wished otherwise, the ANSI standard says that enums may be freely intermixed with integral types, without errors. (If such intermixing were disallowed without explicit casts, judicious use of enums could catch certain programming errors.)

The advantages of enums are that the numeric values are automatically assigned, that a debugger may be able to display the symbolic values when enum variables are examined, and that a compiler may generate nonfatal warnings when enums and ints are indiscriminately mixed (such mixing can still be considered bad style even though it is not strictly illegal) or when enum cases are left out of switch statements.

References: K&R II Sec. 2.3 p. 39, Sec. A4.2 p. 196; H&S Sec. 5.5 p. 100; ANSI X3.159-1989 Secs. 3.1.2.5, 3.5.2, 3.5.2.2 .

Operating System Dependencies

50. How can I read a single character from the keyboard without waiting for a newline?

A: Contrary to popular belief and many people's wishes, this is not a C-related question. The delivery of characters from a "keyboard" to a C program is a function of the operating system, and cannot be standardized by the C language. If you are using curses, use its cbreak() function. Under UNIX, use ioctl to play with the terminal driver modes (CBREAK or RAW under "classic" versions; ICANON, c_ccVMIN and c_ccVTIMEA under System V or Posix systems). Under MS-DOS, use getch(). Under other operating systems, you're on your own. Beware that some operating systems make this sort of thing impossible, because character collection into input lines is done by peripheral processors not under direct control of the CPU running your program.

Operating system specific questions are not appropriate for comp.lang.c . Several common questions are answered in frequently-asked questions postings in the comp.unix.questions and comp.sys.ibm.pc newsgroups.

51. How can I find out if there are characters available for reading (and if so, how many)? Alternatively, how can I do a read that will not block if there are no characters available?

A: These, too, are entirely operating-system-specific. Depending on your system, you may be able to use "nonblocking I/O", or a system call named "select", or the FIONREAD ioctl, or O_NDELAY, or a kbhit() routine.

52. How can my program discover the complete pathname to the executable file from which it was invoked?

A: Depending on the operating system, argv[0] may contain all or part of the pathname. (It may also contain nothing.) You may be able to duplicate the command language interpreter's search path logic to locate the executable if the name in argv[0] is incomplete. However, there is no guaranteed or portable solution.

53. How can a process change an environment variable in its caller?

A: In general, it cannot. If the calling process is prepared to listen explicitly for some indication that its environment should be changed, a special-case scheme can be set up. (Under Unix, a child process cannot directly affect its parent at all. Other operating systems have different process environments which could intrinsically support such communication.)

Stdio

54. Why does errno contain ENOTTY after a call to printf?

A: Many implementations of the stdio package adjust their behavior slightly depending on whether stdout is a terminal or not. To make this determination, these implementations perform an operation which fails (with ENOTTY) if stdout is not a terminal. Although the output operation goes on to complete successfully, errno still contains ENOTTY. This behavior can be mildly confusing, but it is not strictly incorrect, because it is only meaningful for a program to inspect the contents of errno after an error has occurred (that is, after a library function that sets errno on error has returned an error code).

55. My program's prompts and intermediate output don't always show up on my screen, especially when I pipe the output through another program.

A: It is best to use an explicit fflush(stdout) at any point within your program at which output should definitely be visible. Several mechanisms attempt to perform the fflush for you, at the "right time," but they do not always work, particularly when stdout is a pipe rather than a terminal.

56. When I read from the keyboard with scanf(), it seems to hang until I type one extra line of input.

A: scanf() was designed for free-format input, which is seldom what you want when reading from the keyboard. In particular, "\n" in a format string does not mean "expect a newline", it means "discard all whitespace". But the only way to discard all whitespace is to continue reading the stream until a non-whitespace character is seen (which is then left in the buffer for the next input), so the effect is that it keeps going until it sees a nonblank line.

57. So what should I use instead?

A: You could use a "%c" format, which will read one character that you can then manually compare against a newline; or "%*c" and no variable if you're willing to trust the user to hit a newline; or "%*[^\n]*%c" to discard everything up to and including the newline. Or you could use fgets() to read a whole line, and then use sscanf() or other string functions to parse the line buffer.

Miscellaneous

58. Can someone tell me how to write itoa (the inverse of atoi)?

A: Just use sprintf.

59. I know that the library routine localtime will convert a time_t into a broken-down struct tm, and that ctime will convert a time_t to a printable string. How can I perform the inverse operations of converting a struct tm or a string into a time_t?

A: ANSI C specifies a library routine, mktime, which converts a struct tm to a time_t. Several public-domain versions of this routine are available if your compiler does not support it yet.

Converting a string to a time_t is harder, because of the wide variety of date and time formats which should be parsed. Public-domain routines have been written for performing this function, as well, but they are less likely to become standardized.

References: K&R II Sec. B10 p. 256; H&S Sec. 20.4 p. 361; ANSI X3.159-1989 Sec. 4.12.2.3.

60. I seem to be missing the system header file <sgtty.h>. Can someone send me a copy?

A: Standard headers exist in part so that definitions appropriate to your compiler, operating system, and processor can be supplied. You cannot just pick up a copy of someone else's header file and expect it to work, unless that person uses exactly the same environment. Ask your vendor why the file was not provided (or to send another copy, if you've merely lost it).

61. Does anyone know of a program for converting Pascal (Fortran, lisp, "Old" C, ...) to C?

A: Several public-domain programs are available:

p2c written by Dave Gillespie, and posted to comp.sources.unix in March, 1990 (Volume 21).

ptoc another comp.sources.unix contribution, this one written in Pascal (comp.sources.unix, Volume 10, also patches in Volume 13?).

f2c jointly developed by people from Bell Labs, Bellcore, and Carnegie Mellon. To find about f2c, send the message "send index from f2c" to netlib@research.att.com or research!netlib.

FOR_C Available from:
Cobalt Blue
2940 Union Ave., Suite C
San Jose, CA 95124
(408) 723-0474

Promula.Fortran Available from
Promula Development Corp.
3620 N. High St., Suite 301
Columbus, OH 43214
(614) 263-5454

The comp.sources.unix archives also contain converters between "K&R" C and ANSI C.

62. Where can I get copies of all these public-domain programs?

A: If you have access to Usenet, see the regular postings in the comp.sources.unix and comp.sources.misc newsgroups, which describe, in some detail, the archiving policies and how to retrieve copies. Otherwise, you can try anonymous ftp and/or uucp from a central, public-spirited site, such as uunet.uu.net, but this article cannot track or list all of the available sites and how to access them.

63. How can I call Fortran (BASIC, Pascal, ADA, LISP) functions from C? (And vice versa?)

A: The answer is entirely dependent on the machine and the specific calling sequences of the various compilers in use, and may not be possible at all. Read your compiler documentation very carefully; sometimes there is a "mixed-language programming guide," although the techniques for passing arguments correctly are often arcane.

64. Why don't C comments nest? Are they legal inside quoted strings?

A: Nested comments would cause more harm than good, mostly because of the possibility of accidentally leaving comments unclosed by including the characters "/*" within them. For this reason, it is usually better to "comment out" large sections of code, which might contain comments, with #ifdef or #if 0.

The character sequences /* and */ are not special within double-quoted strings, and do not therefore introduce comments, because a program (particularly one which is generating C code as output) might want to print them.

65. I'm having trouble with a Turbo C program which crashes and says something like "floating point not loaded."

A: Some compilers for small machines, including Turbo C and Ritchie's original pdp11 compiler, attempt to leave out floating point support if it looks like it will not be needed. In particular, the non-floating-point versions of printf and scanf save space by not including code to handle %e, %f, and %g. Occasionally the heuristics for "is the program using floating point?" are insufficient, and the programmer must insert one dummy explicit floating-point operation to force loading of floating-point support. Unfortunately, an apparently common sort of program (thus the frequency of the question) uses scanf to read, and/or printf to print, floating-point values upon which no arithmetic is done, which elicits the problem under Turbo C.

In general, questions about a particular compiler are inappropriate for comp.lang.c. Problems with PC compilers, for instance, will find a more receptive audience in a PC newsgroup.

66. Does anyone have a C compiler test suite I can use?

A: Plum Hall, (1 Spruce Ave., Cardiff, NJ 08232, USA), among others, sells one.

67. Where can I get a YACC grammar for C?

A: Several grammars are floating around; keep your eyes open. There is one on uunet.uu.net (192.48.96.2) in net.sources/ansi.c.grammar.Z. FSF's GNU C compiler contains a grammar, as does the appendix to K&R II. Several have recently been posted to the net.

68. Where can I get the "Indian Hill Style Guide" and other coding standards?

A: Various standards are available for anonymous ftp from:

site	file/directory
cs.washington.edu (128.95.1.4)	Uftp/pub/cstyle.tar.Z
cs.toronto.edu	doc/programming
giza.cis.ohio-state.edu	pub/style-guide
prep.ai.mit.edu	pub/gnu/standards.text

69. Where can I get extra copies of this list?

A: For now, just pull it off the net; it is normally posted on about the first of the month, with an Expiration: line which should keep it around all month. Eventually, it may be available for anonymous ftp, or via a mailserver.

Thanks to Mark Brader, Joe Buehler, Christopher Calabrese, Stephen M. Dunn, Tony Hansen, Guy Harris, Karl Heuer, Blair Houghton, Kirk Johnson, Andrew Koenig, John Lauro, Christopher Lott, Rich Salz, Joshua Simons, and Erik Talvola, who have contributed, directly or indirectly, to this article.

Steve Summit
scs@adam.mit.edu
scs@adam.mit.edu@mit.edu

This article is Copyright 1988, 1990 by Steve Summit. It may be freely redistributed so long as the author's name, and this notice, are retained. The C code in this article (vstreat, error, etc.) is public domain and may be used without restriction.

SNUS och SWIPNET

Under 1990 kom det fram förslag om ett kommersiellt TCP/IP-datornät i Sverige. Från början kallades det för Basnät-90, men sedan blev det SNUS och SWIPNET. SNUS är en förkortning av Swedish Network Users Society och SWIPNET betyder Swedish IP Network.

SNUS har nu bildats som ett forum för användare av nätverk. SNUS verksamhet kommer inledningsvis att koncentreras kring Internet IP (TCP/IP). Den viktigaste uppgiften på kort sikt är att initiera driften av SWIPNET.

Kostnaden för medlemskap är 1000 kronor per år. ABC-klubben har bevakat bildandet av SNUS, men är inte medlem.

TCP/IP är ett protokoll som används för nätkommunikation över Ethernet i fram-

förrallt unixmiljö. Föreningen SNUS skall också ha seminarier och demonstrationer av Internet-(TCP/IP)-lösningar, utbildning i nätverksadministration samt bilda arbetsgrupper inom olika teknikområden.

När detta skrivs har SNUS slutit avtal med Comvik Skyport AB för driften av SWIPNET. Anslutning till SWIPNET-backbone sker via en router som placeras i kundens lokaler och en fast förhyrd förbindelse. Kunden ansvarar för trancheiver och trancheivkabel för att ansluta sitt nät till SWIPNET-routern.

Kostnaden för ett företags första anslutning till SWIPNET är beroende på företags storlek, enligt nedan:

Antalet anställda,	Hastighet,	Inträdes-,	Månads-
		avgift	avgift
Stor (fler än 25)	64 K	50.000	10.000
Medel (8 - 25)	19.200	35.000	6.800
Små (1 - 7)	9.600	25.000	2.800

Kostnader är i SEK. Hastighet i bps.

Normalt är SWIPNET tänkt för att användarna skall kunna utnyttja nätet från sitt företag. Man kan använda det för att skicka mail och news även om det inte direkt här huvudskälet. Ett annat sätt är att använda det för att logga in på andra maskiner och föra över filer.

Man lär efter påtryckning från några medlemmar i ABC-klubben fundera på hur föreningar skall kunna komma in i bilden. Vi har ju inga anställda men vi har en hel del medlemmar som kanske i en framtid skulle vilja ha klubbens system via SWIPNET. Detta förutsätter dock att SWIPNET blir en framgång och många företag ansluts.

För att nå datorer inom Internet i USA måste man dock ha sk "Connected Status" vilket innebär att någon i USA måste rekommendera att man kan komma in på delen av nätet i USA som ju bekostas av allmänna medel. Ett motiv för "Connected Status" är någon form av samarbete med myndigheter eller universitet i USA.

Man kommer att kunna nå SUNET och NORDUNET som är universitetsnätverket för Sverige och Norden. Dessutom kommer man att nå InterEuNET som är det nät i Europa som byggs upp mellan de olika Unix-användarföreningarna. Andra nät av samma typ i Europa kommer man säkert också att kunna nå via SWIPNET.

Personligen tycker jag att kostnadsstrukturen är konstig. Ett företag på 26 personer jämföras med ett mycket stort företag som Ericsson!

<1789>
Bo Kullmar

Modem för 9600 bps

Nu börjar modem som kan köra en högre hastighet än 2400 bps bli vanliga. Det är 9600 bps-modem och CCITT-standard heter V.32. Tillsammans med detta kommer också modem med CCITT-standard V.42 för felkorrigering och V.42bis för komprimering. Denna typ av modem klarar normalt även felkorrigering/packning enligt MNP 5-standard.

Detta innebär att filöverföringar och terminalkommunikation kan köras snabbare och stora filöverföringar blir möjliga även för er som ringer rikssamtal. Fram till nu har denna typ av modem varit ganska kostsamma, ofta mellan 10 000 och 15000 kronor exkl moms.

Ett höghastighetsmodem som inte använder CCITT-standard är USRobotics HST modem. Detta modem har blivit mycket populärt bland BBS:er i Sverige. Se även tidigare ABC-blad där de finns engelska texter om detta modem. Det finns också en variant av HST-modemet som även kan köra standarden V.32 och det modemet kallas HST Dual Standard.

ABC-klubbens styrelse har beslutat att satsa på 9600 bps-modem av typ V.32.

Det kommer också att vara möjligt att köra HST-standard om denna modemtyp blir godkänd för anslutning till televerkets nät.

Satsningen på 9600 kommer aldrig att innebära att användare med modem med lägre hastigheter inte kommer att kunna köra mot monitorn. Medlemmar kommer alltid att kunna köra 1200/75 bps-modem mot monitorn och även 300 bps om någon vill köra det.

Vi kommer att börja med ett 9600 V.32-modem eller ett HST-modem som också kan köra V.32. Vilket det blir beror i första hand på vilket modem som blir godkänt först. Vi har nämligen hittat ett nytt V.32-modem från Intel som kommer att marknadsföras i Sverige till ett mycket konkurrenskraftigt pris. Tanken är att använda detta modem i första hand och om HST-modemet blir godkänt även ha ett sådant så medlemmar med HST-modem kan köra monitorn.

Intel-modemet kommer ut på marknaden i början på 1991. Normalpriset ut till kund skall enligt uppgift bli ca 6900 kronor exkl moms. Det är ett mycket konkurrens-

kraftigt pris. Vi arbetar också på ett rabatterbjudande för medlemmarna som kanske kan hamna på ca 5000-6000 kronor exkl moms.

En speciell BBS-rabatt planeras som blir lägre än den summa som nämndes ovan. Vi får köpa modem till detta specialpris och det gör att USRobotics HST troligen bara kommer att förekomma i ett exemplar på ABC-klubbens monitor. Övriga 9600 bps-modem blir V.32-modem.

Vi rekommenderar alltså medlemmarna att skaffa höghastighetsmodem av typen V.32 om högre modemhastighet är önskvärd. Vi ser gärna att medlemmarna erbjuds rabatter på andra typer av bra höghastighetsmodem.

<1789>
Bo Kullmar

Ordförklaringar

BBS	= Bulletin Board System, ett engelskt namn på det vi kallar monitorsystem i ABC-klubben. Vanligen är det PC som privatpersoner sätter upp och låter folk köra på för att hämta filer och delta i konferenssystem.
CCITT	= internationell standardiseringsorganisation inom teleområdet
V.32	= CCITT-standard för 9600 bps-modem
V.42	= CCITT-standard för felkorrigering vid modemtrafik
V.42b	= CCITT-standard för komprimering av information vid modemtrafik
MNP 5	= företaget Microcoms standard för felkorrigering och komprimering vid modemkommunikation
HST	= företaget USRobotics standard för värdbar kommunikation i 14400/300 eller 9600/300 bps
bps	= bitar per sekund, dvs hastighet vid modemkommunikation

The GNU Bulletin

June 1990

The GNU's Bulletin is the semi-annual newsletter of the Free Software Foundation, bringing you news about the GNU Project.

Free Software Foundation, Inc. Telephone: (617) 876-3296
675 Massachusetts Avenue Electronic mail: gnu@prep.ai.mit.edu
Cambridge, MA 02139 USA

Contents

GNU'S Who
What Is the Free Software Foundation?
What Is Copyleft?
GNUs Flashes
Possible New Terms for GNU Libraries by Richard Stallman
"League for Programming Freedom" by Michael Bloom and Richard Stallman

Free Software Support
GNU Project Status Report
GNU Wish List
GNU Documentation
GNU Software Available Now
Contents of the Emacs Release Tape
Contents of the Pre-Release Tape
Contents of the X11 Tapes
VMS Emacs and Compiler Tapes
How to Get GNU Software
Freemac, an Extensible Editor for MS-DOS
Thank GNUs
FSF Order Form

GNU's Who

Joseph Arceneaux is developing Emacs Version 19. Jim Kingdon is working on GDB. Michael Rowan is writing a 'login' replacement to work with 'xdm'. Roland McGrath will again be on the payroll starting in July; he is finishing up the C library and maintains GNU make. David MacKenzie has been hired as a summer programmer and is maintaining the file utilities.

Brian Fox is maintaining various programs that he has written, including the 'readline' library, the 'makeinfo' and Info programs, BASH, and the new GNU 'finger'. Jay Fenlason continues with the GNU spreadsheet, Oleo, as well as maintaining 'tar', 'sed' and the GNU assembler.

Mike Haertel continues work on the C interpreter; he is also maintaining and improving the "bin" utilities and species of 'grep'. David Lawrence, who works for us at the Rensselaer Polytechnic Institute, is maintaining the Lisp libraries for Emacs. Kathy Hargreaves is now working on Ghostscript, having just finished making the regular expression routines mostly POSIX-compliant. Karl Berry is also working on Ghostscript.

Amy Gorin has been hired as a documentation writer, starting with the manual for 'tar'. Diane Barlow Close continues work on the 'BASH Programming Manual'. Grace Sylvan is writing the 'GNU C Reference Manual'.

S. Opus Goldstein is still doing a great job running our office. Erica Brigid is answering phone calls, handling correspondence, and making distribution tapes. Robert J. Chassell, our Treasurer, is working on an elementary introduction to programming in Emacs Lisp, in addition to the many Foundation issues not related to programming.

Richard Stallman continues as a volunteer who does countless tasks, including refining the C compiler, GNU Emacs, etc., and their documentation. Finally, volunteer Len Tower remains our electronic JOAT (jack-of-all-trades), handling mailing lists and gnuSENET, information requests, and the like.

GNU's Bulletin

Copyright (C) 1990 Free Software Foundation, Inc.

Written by: Michael Rowan, Robert J. Chassell, Richard Stallman, Leonard H. Tower Jr., and Michael Bloom

Illustrations: Etienne Suvasa

Japanese Edition: Mieko Hikichi and Nobuyuki Hikichi

What Is the Free Software Foundation?

The Free Software Foundation is dedicated to eliminating restrictions on copying, redistribution, understanding, and modification of computer programs. We do this by promoting the development and use of free software in all areas of computer use. Specifically, we are putting together a complete integrated software system named "GNU" (GNU's Not Unix) that will be upwardly compatible with Unix. Some large parts of this system are already working, and we are distributing them now.

The word "free" in our name refers to two specific freedoms: first, the freedom to copy a program and give it away to your friends and co-workers; second, the freedom to change a program as you wish, by having full access to source code. Furthermore, you can study the source and learn how such programs are written. You may then be able to port it, improve it, and share your changes with others.

Other organizations distribute whatever free software happens to be available. By contrast, FSF concentrates on development of new free software, working towards a GNU system complete enough to eliminate the need to purchase a proprietary system.

Besides developing GNU, the Foundation has secondary functions: producing tapes and printed manuals of GNU software, carrying out distribution, and accepting gifts to support GNU development. We are tax exempt; you can deduct donations to us on your tax returns. Our development effort is funded partly from donations and partly from distribution fees. Note that the distribution fees purchase just the service of distribution: you never have to pay anyone license fees to use GNU software, and you always have the freedom to make your copy from a friend's computer at no charge (provided your friend is willing).

The Foundation also maintains a Service Directory: a list of people who offer service for pay to users of GNU programs and systems. The Service Directory is located in file 'etc/SERVICE' in the GNU Emacs distribution. Service can mean answering questions for new users, customizing programs, porting to new systems, or anything else. Contact us if you want to be listed or wish a copy.

After we create our programs, we continually update and improve them. We release between 2 and 20 updates a year for each program. Doing this while developing new programs takes a lot of work, so any donations of pertinent source code and documentation, machines, labor, or money are always appreciated.

The board of the Foundation is: Richard Stallman, President; Robert J. Chassell, Treasurer; Gerald J. Sussman, Harold Abelson and Leonard H. Tower Jr., Directors.

What Is Copyleft?

In the previous section entitled "What Is the Free Software Foundation?" we state that "you never have to pay anyone license fees to use GNU software, and you always have the freedom to make your copy from a friend's computer at no charge." What exactly do we mean by this, and how do we make sure that it stays true?

The simplest way to make a program free is to put it in the public domain. Then people who get it from sharers can share it with others. But this also allows bad citizens to do what they like to do: sell binary-only versions under typical don't-share-with-your-neighbor licenses. They would thus enjoy the benefits of the freeness of the original program while withholding these benefits from the users. It could easily come about that most users get the program this way, and our goal of making the program free for *all* users would have been undermined.

To prevent this from happening, we don't normally place GNU programs in the public domain. Instead, we protect them by what we call "copylefts". A copyleft is a legal instrument that makes everybody free to copy a program as long as the person getting the copy gets with it the freedom to distribute further copies, and the freedom to modify their copy (which means that they must get access to the source code). Typical software companies use copyrights to take away these freedoms; now software sharers use copylefts to preserve these freedoms.

The copyleft used by the GNU Project is made from a combination of a copyright notice and the "GNU General Public License". The copyright notice is the usual kind. The General Public License is a copying license which basically says that you have the freedoms we want you to have and that you can't take these freedoms away from anyone else. (The actual document consists of several pages of rather complicated legalbol that our lawyer said we needed.) The complete license is included in all GNU source code distributions and many manuals. We will send you a copy on request.

We encourage others to copyleft their programs using the General Public License; basically programs only need to include a few sentences stating that the license applies to them. Specifics on using the License accompany it, so refer there for details.

"As we enjoy great advantages from the inventions of others, we should be glad of an opportunity to serve others by any invention of ours."

-Benjamin Franklin

GNUs Flashes *****

* Emacs Lisp Reference Manual

It's here! We are now shipping the long awaited 'GNU Emacs Lisp Reference Manual'. It describes the GNU Emacs Lisp programming language in great detail. The manual is about 550 pages and can be ordered from us for \$50. Early drafts of this manual are still floating around from several years ago, mostly at 'ftp' sites. We strongly urge you to update to the new version since the improvements are innumerable. (As always you can 'ftp' the manual from the usual sources.)

* Smalltalk

GNU now has a Smalltalk system. It was written by Steve Byrne. We currently have Version 1.0 available on 'prep.ai.mit.edu'. Version 1.1 should be out soon. GNU Smalltalk is based on the "blue book" definition of the Smalltalk language. It is written in highly portable C and runs on a number of UNIX machines (Version 1.1 will offer more ports than the current version, which only runs on 4 or 5 machines). A graphical user interface is also planned. Response to the system has generally been very favorable, especially for a young system.

* GNU Finger

GNU Finger is now available. This is a daemon--based finger program that polls all of the hosts at a site for finger--related information. With GNU Finger, information can now be retrieved for a user on a site--wide basis instead of a host--specific basis. Our finger also displays bitmaps of users' faces where available. See "GNU Finger" under "GNU Project Status Report" for more information.

* Donation from the Open Software Foundation

We want to thank OSF for yet another donation of \$25,000. It is also planning significant improvements for the GNU Assembler and binutils.

* Long-named options

As we have mentioned before, we are adding long-named options to many of our utilities. We have done this by adding functionality to our 'getopt' library. 'getopt long' can now be used to parse long options as well as the normal single-lettered options allowed by the standard 'getopt'. For example, instead of remembering whether to use '-v' or '-V', you can use '+verbose' (or any unambiguous abbreviation) in all programs. Eventually we hope to provide command-line completion for long option names.

* DBM replacement

Version 1.3 of our 'gdbm' library has been released. It is a replacement for the 'dbm' and 'ndbm' libraries. Our 'gdbm' database is stored in one file which contains no large holes, supports read/write interlocking, handles keys and data of unlimited size, and provides routines supporting both the 'ndbm' and 'dbm' interfaces. These features separate 'gdbm' from its proprietary predecessors.

Possible New Terms for GNU Libraries *****

by Richard Stallman

We are considering changing the distribution terms for some GNU libraries, such as 'lib++' and the (as yet unreleased) C library.

The GNU General Public License was designed for utility programs, such as Emacs and GCC. It makes a sharp distinction between using the program and copying any part of it: Any program containing any significant portion of the GNU program must be freely redistributable to be permitted at all. However, merely using the program (for example, an editor) imposes no restriction on the work that is done with it.

Libraries blur the distinction between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work which is a derivative of the original library, and the ordinary General Public License treats it as such. As a result, developers of proprietary software have not used the GNU libraries.

The goal of the Free Software Foundation is to promote the freedom to share software, for software developers and for users; we develop software for sharing as a means to this end. As a pragmatic matter, if the conditions for use of this software are such that most developers choose not to use it, then we don't achieve the goal. At the same time, if the conditions are so loose that people can use the software without much additional sharing, then we don't achieve the goal.

It seems that the ordinary General Public License is too restrictive for libraries, and is discouraging their use rather than encouraging further sharing. However, it would not be best to do what proprietary software developers ask us to do---to permit completely unrestricted use of our libraries in proprietary software---because then the end users of that software wouldn't get a lot of additional freedom as a consequence of the use of our library. We need to find a proper middle ground.

Our idea is to require the distributor of the proprietary executable to make the source to our library available along with the object files for the rest of the application. The user could then recompile the library (perhaps with changes) and relink to get a usable program. This way the user will, in some sense, get the benefit of the free status of the library within the executable.

However, not all the details are settled, so we aren't announcing the precise new library terms just yet.

League for Programming Freedom *****

by Michael Bloom and Richard Stallman

The League for Programming Freedom is an organization of people who oppose the attempt to monopolize common user interfaces through "look and feel" copyright lawsuits. Some of us are programmers who worry that such monopolies will obstruct our work. Some of us are users who want new computer systems to be compatible with the interfaces we know.

"Look and feel" lawsuits aim to create a new class of government-enforced monopolies that would be broader in scope than ever before. Such a system of user-interface copyright would impose gratuitous incompatibility, reduce competition, and stifle innovation.

We in the League hope to prevent these problems by preventing user-interface copyright. The League is not opposed to copyright law as it was understood until 1986---i.e., copyright on particular programs. Our aim is to stop changes in the copyright system, which would take away programmers' traditional freedom to write new programs that are compatible with existing programs and practices.

The League for Programming Freedom will act against the doctrine behind look-and-feel suits by any means consistent with the law and intellectual liberty. We will write editorials, talk with public officials, file amicus curiae briefs with the courts, and boycott egregious offenders. If you have other ideas, please suggest them.

The League members are now voting on the question of opposing software patents, which threaten to make every design decision in software development a chance for a lawsuit. The League's founders consider software patents potentially more dangerous than look-and-feel copyright, and the members voting so far seem to agree. Final results will be known at the beginning of June.

The more members we have, the more effective we will be. Even if you cannot donate any time, simply joining will make a difference.

The dues are \$42 for professionals, \$21 for others, except students whose dues are \$10.50. To join, write to:

League for Programming Freedom,
1 Kendall Square #143,
P.O. Box 9171
Cambridge, MA 02139, USA.
Phone: (617) 492-0023.
'league@prep.ai.mit.edu'

Note that the League for Programming Freedom is a separate organization from the Free Software Foundation; please direct all League inquiries to the above address, not to the Foundation.

Free Software Support *****

The Free Software Foundation develops and distributes freely available software. Our goal is to help computer users as a community. We envision a world in which software is freely redistributable. This means software will be sold at a competitive market price rather than a monopoly established price; often, it will be given away. We see programmers as providing a service, much as doctors and lawyers now do---both medical knowledge and the law are freely redistributable entities for which the practitioners charge a distribution and service fee.

To help you find support and other consulting services, we maintain a list of people who offer such services. We call this list the GNU Service Directory. This list is contained in the file 'etc/SERVICE' in the GNU Emacs distribution. If you want to offer services, you can use this list to help make yourself known. (Contact us if you would like a copy of this directory or wish to be included.)

Most of the listings in the GNU Service Directory are for individuals, but one is for Cygnus Support, which is the first for-profit corporation that we know of that provides support *only* for free software. Their address is 'info@cygnus.com' or Cygnus Support, 814 University Ave., Palo Alto, CA 94301. FSF is not affiliated with Cygnus Support, but we hope that Cygnus Support is a harbinger of the future.

If you find a deficiency in any GNU software, we want to know. We maintain a considerable number of Internet mailing lists for making announcements, reporting bugs and for asking questions. The Emacs and GCC Manuals have chapters explaining where to send bug reports and what information to put in them. Incidentally, on the larger lists, it is not surprising to see an enquiry answered on the same day it is posted.

These mailing lists are also gatewayed into USENET news. If your site receives USENET, you can follow these discussions using news software. To find out more about the 'gnu.*' newsgroups, ask your system administrator.

If you don't have Internet access, you can receive mail and USENET news with a UUCP connection. Contact either a system administrator at a local UUCP site, or UUNET Communications, which can set up a UUCP connection for a modest fee. (UUNET is a non-profit organization that provides network connections.) You can contact UUNET by e-mail at 'info@uUNET.uu.net' or by paper mail at:

UUNET Communications Services,
3110 Fairview Park Drive - Suite 570,
Falls Church, VA 22042
Phone: (703) 876-5050

When we receive a bug report, we will usually try to fix the problem in order to make the software better. This may help you in the long run; however, we may not provide you with immediate assistance. This is not and should not be our job. Our task is so large that we must focus on that which helps the community as a whole, such as developing and maintaining software and documentation. We mustn't let ourselves be sidetracked into helping individuals one by one. We do not have the resources. Thus, do tell us how an installation script doesn't work or where the documentation is unclear---but please don't ask us to help you install the software or figure out how to use it.

If your bug report does not evoke a solution from us, you may still get one from the many other users who read our bug reporting mailing lists. Otherwise, use the Service Directory.

GNU Project Status Report *****

* GNU Emacs

GNU Emacs 18 is now stable. Only a few important bugs have been encountered since Version 18.55.

Berkeley is distributing GNU Emacs with the 4.3 distribution, and numerous companies also distribute it.

Version 18 maintenance continues and a new version, 18.56, is expected soon. It has no new features, however. Version 19 approaches release with a host of new features: before and after change hooks; X selection processing (including CLIPBOARD selections); scrollbars; support for European character sets; floating point numbers; per-buffer mouse commands; interfacing with the X resource manager; mouse-tracking; Lisp-level binding of function keys; and multiple X windows ('screens' to Emacs).

Thanks go to Alan Carroll and the people who worked on Epoch for generating initial feedback to a multi-windowed Emacs. Emacs 19 supports two styles of multiple windows, one with a separate screen for the minibuffer, and another with a minibuffer attached to each screen.

More features of Version 19 are buffer allocation, which uses a new mechanism capable of returning storage to the system when a buffer is killed, and a new input system---all input now arrives in the form of Lisp objects.

Other features being considered for Version 19 include:

- Associating property lists with regions of text in a buffer.
- Multiple font, color, and pixmaps defined by those properties.
- Different visibility conditions for the regions, and for the various windows showing one buffer.
- Incremental syntax analysis for various programming languages.
- Hooks to be run if point or mouse moves outside a certain range.
- Source-level debugging for Emacs Lisp.
- Incrementally saving undo history in a file, so that recover-file also reinstalls buffer's undo history.
- Static menu bars, and better pop-up menus.
- A more sophisticated emacsclient/server model, which would provide network transparent Emacs widget functionality.

* Kernel

We hope to use the Mach message-passing kernel being developed at CMU. The current distributed version of Mach is not free because it contains code from BSD of AT&T origin. However, Mach developers say that all this will be replaced with free code, or at least be moved into user processes. Mach will be free then. This version of Mach should be available in a couple of months (as of May 1990). Until this happens, and we see precisely what is available and on what terms, we can't say for certain whether we can use it.

We will not use Mach unless we can share it with everyone, and all users can redistribute it. In particular, if an export control prevents distribution outside the US, we will not use it.

If we can't use Mach, then we may start developing the GNU kernel with either MIT's TRIX kernel or Berkeley's Sprite system. TRIX is a remote procedure call kernel which runs and supports basic Unix compatibility at about the level of Version 7. It needs a lot of additional features. Sprite is at about the same architectural level as BSD Unix, but has a fancy distributed file system and process migration.

* GNU Debugger

The GNU source-level C debugger, GDB, is now being distributed along with the GNU C Compiler as GDB Version 3.5. Version 2.8, that used to be distributed on the Emacs tape, is now obsolete, and has been replaced by Version 3.5.

We have also started work on GDB Version 4. We have added watchpoints, remote cross-debugging, and a host of minor features. We plan to add over-the-ethernet debugging before the initial release of Version 4.

* C Compiler

The GNU C compiler (GCC) Version 1 is now quite reliable. It supports ANSI standard C. NeXT builds its entire system, including its port of the Mach kernel and NFS, with GCC. We have also been told that GCC successfully compiled a System V.3 kernel. GCC has compiled almost all of the BSD source tree.

GCC performs automatic register allocation, invariant code motion from loops, common subexpression elimination, induction variable optimizations, constant propagation and copy propagation, delaying popping of function call arguments, tail recursion elimination, and many local optimizations that are automatically deduced from the machine description.

Supported CPUs include the 680x0, Vax, 32x32, 80386, 80860, Sparc (Sun 4), SPUR, Convex, MIPS, Tahoe, Pyramid, and Alliant.

We are trying to stabilize GCC Version 1 while incorporating new improvements into Version 2. Version 2 now has support for nested functions, a certain amount of CSE between basic blocks, and a new feature for classifying instructions---which can be used to choose between long and short branches or to provide raw data for instruction scheduling. Instruction scheduling and perhaps global CSE will be added by the time Version 2 is finished. More general calling conventions are supported.

Version 2 supports both C++ and Objective C on the same basis as C itself: the name of the source file selects the language. Michael Tiemann of Cygnus Support has written the C++ front end for GCC (which is available in Version 1 as G++). The front end for Objective C has been donated by NeXT. It will support the Motorola 88000, AMD 29000, IBM RT, and TRON. Ports for the IBM 370, the 3b2, the Neube, a Gould machine (we don't know which one), and the HP Spectrum may be written.

Front ends for Modula, Fortran and Pascal are being developed by volunteers. There are rumors about various other languages. So far, no one has volunteered to write Ada or Cobol.

* C Library

Roland McGrath and others continue to work on the C Library. The C library currently contains all of the ANSI C and POSIX.1 functions, and work is in progress on POSIX.2 and Unix features. This means that the library will have not only all of ANSI, POSIX 1003.1, and POSIX 1003.2, but almost everything found in BSD and System V.

The GNU regular-expression functions ('regex') are now mostly conformant to the POSIX.2 standard.

* Ghostscript

We are distributing Ghostscript on tape. This program provides nearly all the facilities of a Postscript interpreter.

Ghostscript release 1.4 is now available. Staff members Karl Berry and Kathy Hargreaves are working on preparing a new release of Ghostscript, together with Peter Deutsch, the original author of the program. Kathy and Karl are also working on producing free PostScript font files.

Highlights of the new release include drivers for HP's Deskjet and Laserjet as well as the Epson LX-800 (all in low density mode). Ghostscript may be built with multiple drivers (e.g. an X Window driver and a printer driver), and you can switch between them dynamically.

The new release also includes about 20 scalable fonts algorithmically derived from the BDF fonts in the X11 distribution. These include Charter, Courier, Helvetica, New Century Schoolbook, Symbol, and Times fonts. All but the Symbol are provided in different variations.

Right now, Ghostscript will accept commands in PostScript and execute them by drawing on an X Window or writing a file that can be transferred directly to a printer. It also includes a C-callable graphics library (for client programs that don't want to deal with the PostScript Language), and supports IBM PCs and compatibles with EGA graphics as well (but don't ask us about this; we don't use PCs and don't have time to learn anything about them).

Ghostscript needs enhancements: to serve as a previewer for multi-page files; to serve other X clients by drawing on their windows; to be improved both in its performance and visual quality; and more fonts. Any suggestions for enhancements are welcome.

* GNU Finger

Brian Fox has released the new GNU Finger program. Finger is a daemon-based replacement for (or supplement to) BSD 'finger'. Our finger now handles the newer paradigm of each user having his/her own small Unix host (workstation). GNU finger has a per site server that will poll all of the machines at a site. Thus queries can now be site wide rather than host specific. For example, fingering bfox@pogo.ai.mit.edu with our finger would tell you if he was logged in and what host he is using. If he isn't logged in, it will tell you what host he was last using.

Our finger also does faces! If a site has face bitmaps online (and is running GNU Finger, of course) you can get the user's face in a X Window on your display.

* Oleo

Jay Fenlason is writing a spreadsheet named Oleo (which is better for you than the more expensive spreadsheet). Oleo is in alpha test right now; we do not know when it will be available. Jay says that "really brave" people can contact him about being alpha testers.

Oleo currently reads and writes SC and Multiplan SYLK files, but teaching it new formats is fairly simple. It has a full set of expressions as well as mathematical, financial, and string functions. Keys may all be rebound and Oleo also has primitive macro support.

Oleo uses the 'curses' library and an X11 interface is planned. Right now it runs on BSD Unix machines as well as IBM PC's and compatibles.

* groff

James Clark is writing 'groff', an implementation in C++ of the traditional Unix document formatting tools. So far 'troff', 'pic', 'tbl', 'eqn', 'man' macros, and a PostScript driver have been written. A version of the Berkeley 'me' macros will be included. He is currently implementing a driver which produces TeX '.dvi' format and a driver for typewriter-like devices. Useful additions would be 'mm' and 'ms' macros, 'grap', and 'refer'.

* GNU Mailer

Development of Smail is coming to an end. We are satisfied with the program's current features, except for the queueing system. We hope we can replace Smail's queueing system with the queueing system found in Zmailer. Otherwise a new one will have to be written.

* File Manipulation Utilities

We have added a collection of utilities for file manipulation to the Pre-Release tape. The collection includes 'ls', 'mv', 'cp', 'cat', 'rm', 'du', 'head', 'tail', 'cmp', 'chmod', 'mkdir', and 'ln'. These tools are either fully POSIX compliant or being worked on to become so. 'cp' now has options to preserve the last-modification timestamp on copies, thus replacing some uses of 'tar' or 'cpio'.

* Smalltalk

Our Smalltalk system will be available in tape form when we release our "Experimental" tape in November. It is currently available via 'ftp' on 'prep.ai.mit.edu'. The current version is 1.1, with Version 1.2 expected to be out soon. Thanks to Steve Byrne who wrote our Smalltalk based on the 'blue book' definition of the language.

GNU Wish List

Wishes for this issue are for:

- * Someone skilled in compiler maintenance who could take over GCC maintenance for RMS. This would probably be a full-time job.
- * A Sun with a SCSI port to be donated or loaned so we can make distribution tapes. We also need machines to be donated or loaned for use by FSF programmers and documentation people who are not located near our offices in Cambridge.

* Professors who might be interested in sponsoring or hosting research assistants to do GNU development, with full or partial FSF support. Several schools have done this and we welcome others to join in.

* Volunteers to help write utilities and documentation. Send mail to 'gnu@prep.ai.mit.edu' for the task list and coding standards.

* Speech and character recognition software (if the devices aren't too weird), with the device drivers (if possible). This would help the productivity of at least one partially disabled programmer we know.

* Grammar checking software for English and other natural languages.

* Copies of newspaper and journal articles mentioning the GNU Project or GNU software. Send these to the address on the front cover, or send a citation to 'gnu@prep.ai.mit.edu'.

* Money, as always. Please remember, donations are tax-deductible. With the latest donations, we have been able to expand our staff again. With the increased staff we have an even greater need for donations.

One way to give us a small amount of money is to order a distribution tape or two. This may not count as a donation for tax purposes, but it can qualify as a business expense.

GNU Documentation

GNU is dedicated to having quality, easy-to-use, on-line and printed documentation. GNU manuals are intended to explain the underlying concepts, describe how to use all the features of each program, and give examples of command use.

GNU documentation is distributed as Texinfo source files. Texinfo source yields both a typeset hardcopy and on-line presentations, accessed by a menu-driven system.

The following manuals, provided with our software, are also available in hardcopy; see the order form on the inside back cover.

The Emacs Manual describes the use of GNU Emacs. It also explains advanced features, such as outline mode and regular expression search. The manual tells how to use special modes for programming in languages such as C and Lisp, how to use the tags utility, and how to compile and correct code. It also describes how to make your own keybindings and other elementary customizations.

The Emacs Lisp Reference Manual covers the GNU Emacs Lisp programming language in great depth. It goes into data types, control structures, functions, macros, byte compilation, keymaps, windows, markers, searching and matching, modes, syntax tables, operating system interface, etc.

The Texinfo Manual describes how to write documents in Texinfo source code. It explains the markup language used to create both an Info file and a printed document from the same source file. This tells you how to make tables, lists, chapters, nodes, indices, and cross references. It also describes how to use Texinfo mode in GNU Emacs and catch mistakes.

The Termcap Manual is often described as "Twice as much as you ever wanted to know about Termcap." It describes the format of the Termcap database, the definitions of terminal capabilities, and the process of interrogating a terminal description. This manual is primarily for programmers.

The Bison Manual covers writing grammar descriptions that can be converted into C coded parsers. This manual assumes no prior knowledge of parser generators. It describes the concepts and then provides a series of increasingly complex examples before describing what happens in considerable detail.

The GAWK Manual describes how to use the GNU implementation of AWK. It is written for someone who has never used AWK, and describes all the features of this powerful string manipulating language.

The Make Manual describes the GNU Make utility, a program used to rebuild parts of other programs when and as needed. It covers makefile writing, which specifies how a program is to be compiled and what each part of the program depends on.

The GDB Manual explains how to use the GNU Debugger. It describes running your program under debugger control, how to examine and alter data as well as modify the flow of control within the program, and how to use GDB through GNU Emacs, with auto-display of source lines.

GNU Software Available Now

We offer Unix software source distribution tapes, plus VMS tapes for GNU Emacs and GNU C that include sources and VMS executables.

The first Unix tape (called the "Release" or "Emacs" tape) contains GNU Emacs as well as various other well-tested programs that we consider reliable. The second Unix tape (called the "Pre-Release" or "Compiler" tape) contains the GNU C compiler, related utilities, and other new programs that are less thoroughly tested. The third and fourth Unix tapes (called the "X11" tapes) contain the X11 distribution from the MIT X Consortium.

The GNU C compiler and various related programs are on a tape that we have called our "Beta Test" tape in the past. However, this software is becoming more stable; to reflect this change, we are renaming the tape the "Pre-Release" tape. (The tape also contains Ghostscript, which is not stable; for the moment, this is the best place to put it.)

Sometime in the Fall, probably in early November, we will introduce a new beta test tape of "experimental" software. We will put new major, test releases of existing more-or-less stable programs such as GCC, Emacs, and GDB on this tape when they appear, as well as new programs such as Smalltalk and the C library. The contents of the "Experimental" tape will fluctuate because versions will move to the other tapes when they become stable. We are calling this the "Experimental" tape to prevent confusion with the older Beta tape.

We will put Ghostscript on the "Experimental" tape when it appears. But as a convenience, we will continue to include it on the "Pre-Release" tape until the current order form expires in January 1991, even though this action is somewhat inconsistent with the tape naming conventions.

Please do not order an "Experimental" tape until at least November, unless you see an announcement sooner than that on the net---we have put nothing on this tape as yet!

See the order form on the inside back cover for details about media, etc. Note that the contents of the 1600bpi 9-track tapes and the QIC-24 DC300XLP 1/4 inch cartridge tapes for UNIX systems are the same. It is only the media that are different.

Contents of the Emacs Release Tape

The software on this release tape is considered fairly stable, but as always, we welcome your bug reports.

* GNU Emacs

In 1975, Richard Stallman developed the first Emacs, an extensible, customizable real-time display editor. GNU Emacs is his second implementation of Emacs. It's the first Emacs available on Unix systems that offers true Lisp---smoothly integrated into the editor---for writing extensions. It also provides a special interface to MIT's free X window system. The current version of Emacs is 18.55.

GNU Emacs has been in widespread use since 1985 and often displaces proprietary implementations of Emacs because of its greater reliability as well as its additional features and easier extensibility. DEC, Berkeley, and NeXT are all distributing Emacs with their systems. When Isaac Salzman set out to review various versions of Emacs, only one company wanted its product to be compared with GNU Emacs. In his review, which appeared in the July, 1989 issue of Unix Review, Salzman concluded, "When it comes to Emacs, GNU is the way to go."

pGNU Emacs (as of Version 18.55) runs on many Unix systems: Alliant, Altos 3068, Amdahl (UTS), Apollo, AT&T (3B machines & 7300 PC), CCI 5/32 & 6/32, Celerity, Convex, Digital (DECstation 3100; Vax running BSD or System V), Motorola Delta (running System V/68 release 3), Dual, Elxsi 6400, Encore (DPC, APC, & XPC), Gould, HP (9000 series 200, 300 or 800 (Spectrum) but not series 500), HLL Orion 1/05, IBM (RT/PC running 4.2 & AIX; PS2 running AIX), Integrated Solutions (Optimum V with 68020 & VMEbus), Intel 80386 (BSD, System V, & Xenix; not MS-DOS), Iris (2500, 2500 Turbo, & 4D), LMI (Nu), Masscomp, Megatest, MIPS, NCR (Tower 32), Nixdorf Targion 31, Plexus, Prime, Pyramid, Sequent (Balance & Symmetry), SONY News, Stride (system release 2), Sun (1, 2, 3, 4, SparcStation, & 3861), Tahoe, Tektronix (NS32000 & 4300), Stardent 1500 or 3000, Titan P2 or P3, Pmax, Texas Instruments (Nu), & Whitechapel (MG1).

GNU Emacs is described by the "GNU Emacs Manual", which comes with the software in Texinfo form. See "GNU Documentation" above. Also, since GDB is the only debugger that can debug Emacs without losing its mind, it is included on this tape as well as the Pre-Release Tape.

* GNU Emacs Lisp Reference Manual

We now include the Texinfo source to the "GNU Emacs Lisp Reference Manual" with Emacs. The manual describes the GNU Emacs Lisp programming language in detail and is for anyone who is interested in writing programs in GNU Emacs Lisp. See "GNUs Flashes" and "GNU Documentation" in this bulletin for more information.

* Bison

Bison is an upwardly compatible replacement for the parser generator Yacc, with additional features. It has been in use for several years. Bison is used for compiling GNU C, so it is included on the GNU Pre-Release tape as well. The "Bison Manual" comes with the software in Texinfo form (see "GNU Documentation" above).

* X Window System, V10R4

We are no longer including a copy of X10 on our distribution tapes. It is no longer supported by MIT, so distributing it does not make sense. X Version 11 (currently release 4) is now pretty stable and available from us on two separate tapes. See "Contents of the X11 Tapes."

* MIT Scheme

Scheme is a simplified, lexically scoped dialect of Lisp. It was designed at MIT and other universities to teach students programming and to research new parallel programming constructs and compilation techniques. MIT Scheme is written in C and runs on many Unix systems. It now conforms to the "Revised"3 Report On The Algorithmic Language Scheme" (MIT AI Lab Memo 848a), for which Tex source is included in the distribution. Another good source of documentation for Scheme is "Structure and Interpretation of Computer Programs", by Harold Abelson and Gerald J. Sussman with Julie Sussman, the MIT Press & McGraw-Hill Book Company, 1985.

* Yale T

A variant of Scheme developed at Yale University, T is intended for production use in program development. T contains a native-code optimizing compiler that produces code that runs at speeds comparable to the speeds of programs written in conventional languages. It runs on BSD Vaxes, 680x0 systems, SPARC workstations, MIPS R2000 workstations (including the Decstation 3100 Pmax), and NS32000 machines (including the Encore Multimax). T is written in itself and cannot be bootstrapped without a binary (included), but it is great if you can use it. Some documentation is included.

* `texi2roff'

`texi2roff', written by Beverly Erlebacher, translates GNU Texinfo files into a format that can be printed by the Unix Anttroff programs utilizing the mm, ms or me macro packages. It is included on all UNIX tapes so people who don't have a copy of TeX can print out GNU documentation.

* GNU Chess and NetHack

GNU Chess is a chess program, now in its second major version. The first was written by Stuart Cracraft. The second was written and donated by John Stanback. If a successor is found that is significantly stronger, it could become the new GNU Chess. GNU Chess has text-only and X display interfaces.

NetHack is a display--oriented adventure game similar to Rogue.

Contents of the Pre-Release Tape

The programs on this tape are becoming almost stable. The exception is Ghostscript, but which we are carrying on this tape as a convenience until January 1991, when we will distribute it only on the "Experimental" tape. As always, we solicit your comments and bug reports. This tape is also known as the Compiler tape, and used to be known as the "Beta" tape.

* GNU CC

The GNU C compiler is a fairly portable optimizing compiler. It generates good code for the 32000, 680x0 (optionally with 68881/2), 80386, 860, 88000, Alliant, Convex, Tahoe, and Vax CPUs, and for these RISC CPUs: Pyramid, SPARC, and SPUR. The MIPS RISC CPU is also supported. Machines using these CPUs include 386 running AIX, Alliant FX/8, Altos 3068, Apollo 68000/68020 running Aegis, AT&T 3B1, Convex C1 and C2, DECstation 3100, DEC VAX, Encore MultiMax (NS32000), Genix NS32000, Harris HCX-7 and HCX-9, HP-UX 68000/68020, HP running BSD, IBM PS/2 running AIX, Intel 386, Iris MIPS machine, ISI 68000/68020, MIPS, NeXT, Pyramid, Sequent Balance (NS32000), Sequent Symmetry (i386), SONY News, Sun 2, Sun 3 (optionally with FPA), Sun 4, SparcStation, and Sun386i. The current version is 1.37. It supports full ANSI C. Please refer to the "GNU Project Status Report" for more detail on GCC.

Included with the compiler are Bison (also on the Emacs release tape), and the perfect hash-table generating utility (Gperf), plus the Texinfo source of the "GCC Manual". This manual describes how to run and install the GNU C compiler, and how to port it to new processors. It describes new features and incompatibilities of the compiler, but people not familiar with C will also need a good book on C.

* Assembler and Object File Utilities

The GNU assembler (GAS) is a fairly portable, one pass assembler that is almost twice as fast as Unix 'as'. It is now at Version 1.35 and works for 32x32, 680x0, 80386, Sparc (Sun 4), and Vax.

We have free versions of 'ar', 'ld', 'nm', 'size', 'gprof', 'strip', and 'ranlib'. The GNU linker 'ld' runs significantly faster than the BSD version. Our 'ld' is the only one that will give you source-line numbered error messages for multiply-defined symbols and undefined references.

* COFF Support

It is possible to run the entire suite of GNU software tools on System V, replacing COFF entirely. The GNU tools can operate on BSD object files with a COFF header the System V kernel will accept. 'robotussin' is supplied for converting standard libraries to this format.

* `make'

GNU 'make' includes almost all the features from the BSD, System V, and POSIX versions of make, as well many of our own extensions. These extensions include parallelism, conditional execution, and text manipulation. Version 3 of GNU make is fairly stable and we do not anticipate a Version 3 release after 3.59. Work on Version 4---which will include many functional improvements---will begin sometime this summer. Texinfo source for the GNU make manual is provided; see "GNU Documentation" above.

* Debugger

Version 3.* of GDB, the GNU debugger, runs under BSD 4.2 and 4.3 on Vaxes and Suns (2, 3, and 4), Convex, HP 9000/300's under BSD, HP 9000/320's under HPUX, System V 386 machines (with either GNU or native object file format), ISI Optimum V, Merlin under Utek 2.1, SONY News, Gould NPL and PN machines, Pyramid, Sequent Symmetry (a 386 based machine), Altos, and Encore under Umax 4.2.

GDB features incremental reading of symbol tables (for fast startup and less memory use), command-line editing, the ability to call functions in the program being debugged, a value history, and user-defined commands. It can be used to debug C, C++, and FORTRAN programs.

GDB also provides for remote debugging over a serial line. Remote debugging is the most convenient way to develop software for systems which are too small to run a debugger; it allows you to have the features of GDB at your disposal even on such systems.

* BASH

The GNU Shell, BASH (for Bourne Again SHell), provides compatibility with the Unix 'sh' and provides many extensions found in 'csh' and 'ksh'. It has job control, 'csh'-style command history, and command-line editing (with Emacs and vi modes built-in and the ability to rebind keys).

* GAWK, FLEX, and 'tar'

GAWK is GNU's version of the Unix AWK utility; it comes with a Texinfo manual (see "GNU Documentation" above). FLEX is a mostly-compatible replacement for the Unix 'lex' scanner generator written by Vern Paxson of the Lawrence Berkeley Laboratory. FLEX generates far more efficient scanners than 'lex' does. GNU 'tar' includes multivolume support, the ability to archive sparse files, automatic compression and decompression of archives, remote archives, and special features to allow 'tar' to be used for incremental and full backups of file systems.

* Freed Files from the U.C. Berkeley 4.3-tahoe Release

These files have been declared by Berkeley to be free of AT&T code, and may be freely redistributed. They include complete sources for some utility programs, games, and library routines; and partial sources for many others.

* RCS and CVS

The Revision Control System is used for version control and management of large software projects. This is the latest version (4.0).

CVS, the Concurrent Version System written by Brian Berliner, manages software revision and release control in a multi-developer, multi-directory, multi-group environment. It is designed to work on top of RCS Version 4, but will parse older RCS formats with the loss of CVS's fancier features. For further details, see Berliner, Brian, 'CVS-II: Parallelizing Software Development', Proceedings of the Winter 1990 USENIX Association Conference.

* 'diff' and 'grep'

These programs are GNU's versions of the Unix programs of the same name. They are much faster than their Unix counterparts.

* Ghostscript and 'gnuplot'

Ghostscript is GNU's graphics language that is almost fully compatible with PostScript. For more information on Ghostscript, please refer to the section on Ghostscript in the "GNU Project Status Report."

'gnuplot' is an interactive program for plotting mathematical expressions and data. Oddly enough, the program was neither done for nor named for the GNU Project---the name is a coincidence.

* 'g++', 'libg++', and NIH Class Library

G++ is a set of changes for GCC that compiles C++, the well-known object-oriented language. This was the first compiler to compile C++ directly instead of preprocessing it into C, with great benefits for debugging and efficiency. G++ also was first with multiple inheritance and other new features later released by AT&T in 'cfront 2.0'. Since G++ depends on GCC, it must be used with the correspondingly numbered version of GCC. GDB Version 3.* includes support for debugging C++ code, which merges in the functionality of the old program GDB+.

'libg++' (the GNU C++ library) is an extensive and documented collection of C++ classes and support tools for use with G++.

The NIH Class Library (formerly known as OOPS (Object-Oriented Program Support)) is a portable collection of classes similar to those in Smalltalk-80 that has been developed by Keith Gorlen of NIH, using the C++ programming language.

Note that Interviews has been dropped from this tape since it appears on the "optional" X tape (See "Contents of the X11 Tapes" below).

* File Utilities and Miscellaneous

The file utilities, which include the programs listed in the "GNU Status Report," are now included here. We also include 'compress', 'perl' (Version 3.0), 'c-perf' (Version 2.0), 'f2c' (a FORTRAN to C translator), and GnuGo (the game of Go (Wei-Chi)) on this tape.

Contents of the X11 Tapes

The two X11 tapes contain Version 11, Release 4 of the MIT X window system. X11 is more powerful than, but incompatible with, the no-longer-supported Version 10.

The first FSF tape contains the contents of both tape one and tape two from the MIT X Consortium: the core software and documentation, and the contributed clients. FSF refers to its first tape as the 'required' X tape since it is necessary for running X or GNU Emacs under X. (The Consortium refers to its first two tapes as the 'required/recommended' tapes.)

The second, 'optional' FSF tape contains the contents of tapes three and four from the MIT X Consortium: contributed libraries and other toolkits, the Andrew software, games, etc. (The Consortium refers to its last two tapes as 'optional' tapes.)

VMS Emacs and Compiler Tapes

We offer a VMS tape of the GNU Emacs editor, and a separate VMS tape containing the GNU C compiler. The VMS compiler tape also contains Bison (needed to compile GCC), GAS (needed to assemble GCC's output), and some library and include files. Both VMS tapes include executables that you can bootstrap from, because the DEC VMS C compiler has bugs and thus cannot compile GNU C.

Please don't ask us to devote effort to additional VMS support, because it is peripheral to the GNU Project.

How to Get GNU Software

All the software and publications from the Free Software Foundation are distributed with permission to copy and redistribute. The easiest way to get GNU software is to copy it from someone else who has it.

If you have access to the Internet, you can get the latest software from the host 'prep.ai.mit.edu'. For more information, read the file '/u/emacs/GETTING.GNU.SOFTWARE' on that host. Please note that the Internet address of 'prep' is '18.71.0.38'.

If you cannot get the software from a friend or over the net, or if you would like to contribute some funds to our efforts and receive the latest versions, the Free Software Foundation distributes tapes for a copying and distribution fee. See the order form on the inside back cover.

There are also third party groups that distribute our software: they do not work with us, but have our software in other forms. For your convenience, some of them are listed below. Please note that the Free Software Foundation is not affiliated with them in any way, and is not responsible for either the currency of their versions or the swiftness of their responses.

These Internet sites provide GNU software via anonymous 'ftp':

soam.berkeley.edu, itstd.sri.com, wuarchive.wustl.edu, wsmr-simtel20.army.mil (under 'PD:UNIX.GNU'), bu.edu, louie.udel.edu, nic.nyser.net, ftp.cs.titech.ac.jp, funic.hut.fi, sunic.sunet.se, freja.diku.dk, gatekeeper.dec.com, mango.miami.edu (VMS G+), cc.utah.edu (VMS GNU Emacs), and uunet.uu.net.

Those on the SPAN network can ask rdss::corbet.

Information on how to obtain some GNU programs using UUCP is available via electronic mail from the following people. Ohio State also posts their UUCP instructions regularly to newsgroup 'comp.sources.d' on USENET.

hao!scicom!qetza!upba!ugn!nepa!denny, acornrc!bob, hqda-ai!merlin, uunet!hutch!barber, sun!nosun!illian!darylm, oli-stl!root, bigtex!james, postmaster!uunet.uu.net, and karl!tut.cis.ohio-state.edu (or 'osu-cis!karl').

Freemac, an Extensible Editor for MS-DOS

Freemac, a copylefted MS-DOS editor, is one of the few editors for small machines with a full extension language. It is the only such editor that tries to be compatible with GNU Emacs. For more information contact: 'nelson@clutx.clarkson.edu', or: Russell Nelson, 11 Grant St., Potsdam, NY, 13676. \$15 sent to that address will get you a copy. It is also available for 'ftp' on 'sun.soe.clarkson.edu'. Note that the Free Software Foundation does not distribute Freemac; please don't ask us about it.

Thank GNUs

Thanks to all those mentioned in GNUs Flashes and the GNU Project Status Report.

Thanks to the Japanese Unix Society for their large gift.

Thanks to Delta Microsystems who just donated an Exabyte tape drive.

Thanks again to the Open Software Foundation for their continued support.

Thanks to Digital Equipment Corporation for their gift.

Thanks to Bill Lewis, Dan LaLiberte, and the volunteers who worked on the 'Emacs Lisp Reference Manual'. Also thanks to Warren A. Hunt, Jr. and Computational Logic, Inc. for their support.

Thanks to the MIT Artificial Intelligence Laboratory & MIT Laboratory for Computer Science for their invaluable assistance of many kinds.

Thanks to Chris Welty as well as the Computer Science Department at Rensselaer Polytechnic Institute for splitting Dave Lawrence's salary with FSF and providing him computing facilities.

Thanks to Prof. Christof Koch of Caltech for his support of Brian Fox.

Thanks to Prof. Paul Hilfinger of the UCB CS Department for allowing Roland McGrath to use UCB resources.

Thanks to the University of Minnesota Department of Computer Science for allowing Mike Haertel to use their computers.

Thanks to Cliff Lasser of Thinking Machines, Inc. for the help with upgrading to SunOS 4.0.

Thanks to Village Center Inc and Nikkei Business Publishing, both of Japan, for their gifts.

Thanks to Information Systems and the Whitaker College Computing Facility at MIT for use of their machines to make our VMS master tapes.

Thanks go out to all those who have either lent or donated machines, including Hewlett-Packard for their donation of six 68030 workstations, Brewster Kahle of Thinking Machines Corp. (TMC) for the Sun 4/110, K. Richard Magill for the AT&T Unix PC, Doug Blewett of AT&T Bell Labs for two Convergent Miniframes, CMU's Mach Project for the Sun 3/60, Intel Corp. for their 386/1860 workstation, NeXT for a NeXT workstation, the MIT Media Laboratory for the Hewlett-Packard 68020 machine, SONY Corp. and Software Research Associates, Inc., both of Tokyo, for three SONY News workstations, and the MIT Laboratory of Computer Science for the DEC Microvax.

Thanks to all those who have contributed ports and extensions, as well as those who have contributed other source code, documentation, and good bug reports. Thanks to those who sent money and offered help. Thanks also to those who support us by ordering manuals and distribution tapes.

Free Software Foundation Order Form

This form is effective June 1990--January 1991.

Prices and contents of tapes are subject to change without notice.

All software and publications are distributed with permission to copy and redistribute.

TeX source for each manual is on the appropriate tape; the prices for tapes do not include printed manuals.

All software from the Free Software Foundation is provided on an "as is" basis, with no warranty of any kind.

Quantity Price Item

For Unix systems, on 1600 bpi 9-track tape in Unix tar format:

_____ \$150 GNU Emacs source code and other software.
The tape includes:
* GNU Emacs (the extensible, customizable, self-documenting real-time display editor)
* The GNU Emacs Lisp Reference Manual, as source code.
* MIT Scheme (a dialect of Lisp)
* T. Yale's implementation of Scheme
* Bison (a free, compatible replacement for yacc)
* Nethack (a rogue-like game)
* GNU Chess (a chess playing program with an interface to X).
* GDB (release version of the GNU source-level C debugger)

_____ \$150 GNU prerelease software, for Unix systems.
The tape includes:
* GCC (the GNU C Compiler, including COFF support)
* G++ (the C++ front end to GCC)
* lib-g++ (the G++ class library)
* NIH Class Library (formerly known as OOPS)
* Bash (GNUs' Bourne Again Shell)
* Bison (a free, compatible replacement for yacc)
* Flex (Vern Paxson fast rewrite of lex)
* Ghostscript (a Postscript interpreter)
* Gawk (the GNU implementation of the AWK programming language)
* Gas (the GNU Assembler)
* GDB (beta version of the GNU source-level C debugger)
* Gnuplot (an interactive mathematical plotting program)
* Compress (a file compression program)
* Perl (version 3.0; a programming language interpreter)
* RCS (Revision Control System)
* CVS (Concurrent Control System)
* GNU object file utilities (ar, ld, make, gprof, size, nm, strip, ranlib, et al.)
* other GNU utilities (make, diff, grep, tar, et al.)
* GNU GO (the GNU implementation of the game of GO)
* the freed files from the 4.3BSD-Tahoe distribution

_____ \$150 GNU Experimental software. Not ready before Nov. 1990.

_____ \$150 Required MIT X Window System X11R4, core software and documentation, and contributed client software.

_____ \$150 Optional MIT X Window System X11R4, contributed software including libraries, games, Andrew and toolkits.

For Suns and other Unix Systems, on QIC-24 DC300XLP 1/4 inch cartridge tape, Unix tar format:

_____ \$175 GNU Emacs and other software, as in the first item.

_____ \$175 GNU Beta Test tape, for Suns, as in the second item.

_____ \$175 GNU Experimental software, as in the third item (Nov. 1990).

_____ \$175 Required MIT X Window System X11R4, as in the fourth item.

_____ \$175 Optional MIT X Window System X11R4, as in the fifth item.

For VMS systems, on 1600 bpi reel-to-reel 9-track tape in VMS BACKUP format:

_____ \$150 GNU Emacs source code and binaries.

_____ \$150 GNU C compiler source code and binaries.
Includes Bison and GAS.

GNU Emacs manual, 1300 pages, phototypeset, offset printed, spiral bound, with a reference card.

_____ \$15 A single GNU Emacs manual.

_____ \$60 Box of six GNU Emacs manuals.

The following documentation:

_____ \$1 One GNU Emacs reference card, without the manual.

_____ \$5 Packet of ten GNU Emacs reference cards.

_____ \$50 Gnu Emacs Lisp Reference Manual, 1550 pages.

_____ \$10 GDB Manual, 170 pages, side stapled.

_____ \$10 Texinfo Manual, 1100 pages, side stapled. Texinfo is GNU's structured documentation system, included with GNU Emacs. Texinfo is used to produce both on-line and printed documents. This manual describes how to write Texinfo documents.

_____ \$10 Termcap Manual, 160 pages, side stapled. Documents the termcap library and GNU's extensions to it. The GNU termcap library is included with GNU Emacs.

_____ \$10 Bison Manual, 180 pages, side stapled.

_____ \$10 Gawk Manual, 1100 pages, side stapled.

_____ \$10 Make Manual, 1100 pages, side stapled.

_____ Sub Total

_____ If ordering from Massachusetts: add 5% sales tax.

_____ If outside of North America and Hawaii, for shipping costs:
- for tapes or unboxed manuals, please add \$15, and then add \$15 more for each tape or unboxed manual in the order;
total \$ amount for this item = \$15 + \$15 times number;
- for each box of Emacs manuals,
please add \$60.

_____ Optional tax deductible donation.

_____ Total paid

Orders are filled upon receipt of check or money order. We do not have the staff to handle the billing of unpaid orders. Please help keep our lives simple by including your payment with your order.

Please make checks payable to Free Software Foundation. Mail orders to:

Free Software Foundation, Inc.
675 Massachusetts Avenue
Cambridge, MA 02139

617-876-3296

The ANSI Standard

A summary for the C-programmer

Detta är en artikel om den nya ANSI standarden för programmeringsspråket C. Författaren har själv publicerat artikeln i USENET och har nedan skrivit en notering om texten som från början skrev för annat ändamål 1987. Trots detta anser han att den är aktuell idag. /Bo Kullmar

Author's Note:

The following summary was originally written for inclusion with the Waterloo C compiler (VM/CMS) and has also been published as an article in the Transactor for the Amiga (vol 1 no 3). I wrote it in 1987 and it has not yet been updated to reflect the final version of the Standard. I think, however, that it still presents the major changes quite nicely. You can also check out the second edition of K&R for more information. This document was originally done using Script/GML --- basically I just removed the formatting macros, which explains the weird spacing and varying line lengths. If there is enough interest I may polish this document up into TeX format and update it to reflect the final Standard.

The ANSI Standard: A Summary for the C Programmer

Eric Giguere
Computer Systems Group
University of Waterloo
giguere@csg.uwaterloo.ca

December 18, 1987

This document is Copyright (c)1987 by Eric Giguere.
Permission is granted to use and distribute this
document for non-commercial purposes.

Introduction

Programming languages are constantly evolving and diversifying. The C language is no exception, especially due to its increased popularity in recent years. The original specification document for C, "The C Programming Language" by Brian Kernighan and Dennis Ritchie (commonly referred to as K&R), is now almost ten years old. K&R has served as the C programmer's "bible", the de facto standard for C. But as the language has evolved, the need for a formal language standard has become apparent.

Under the auspices of the International Standards Organization (ISO), the American National Standards Institute (ANSI) began the preparation of such a standard through the leadership of the X3J11 Technical Committee. The proposed standard is now in final draft form and is expected to be approved by ANSI and ISO in 1988.

This document is a summary of the proposed standard's major changes to the language as it pertains to the C programmer. All information is drawn from official X3J11 documents: the "Draft Proposed American National Standard for Information Systems - Programming Language C" and the accompanying "Rationale for Draft Proposed National Standard for Information Systems - Programming Language C". (These publications will be referred to as the Standard and the Rationale, respectively.)

Purpose of the Summary

Few programmers have either the time or the interest to wade through the actual text of the draft Standard in its entirety. What interests them are the major points of the Standard and the changes it makes to what was defined in K&R and how it affects their current programs. By summarizing these changes, this document is intended to provide a quick reference that the average C programmer can read and understand in one session. In keeping with this goal, the Standard Library is only briefly mentioned. Readers interested in the specifics of the Library should consult the Standard itself or the documentation accompanying any ANSI-conforming compiler.

This document is not a criticism or a justification of the Standard, only a commentary. Nor is it a tutorial on the C programming language. Readers should also be aware that changes may occur to the Standard before its final acceptance by ANSI.

What to expect from the Standard

The Standard is not creating a new language definition. Its purpose, to quote the Rationale, is to "codify common existing practice". This means that the fundamental structure and syntax of the language as described in K&R has been left unchanged. The Standard has instead tried to unify the diverse extensions and dialects that have grown over the years (the existing practice) into a single cohesive language definition. Existing practice is often inconsistent, however, so many compromises have had to be made.

Perhaps the most important thing to remember about the Standard is that it is not intended to invalidate existing C code. Existing programs should compile with only minor changes when using an ANSI-conformant compiler.

Common Terms

Throughout this document the word implementation will be used to refer to any particular implementation of an ANSI-conformant C language interpreter or compiler.

1. Reserved Identifiers

The following keywords have been added to the language:

```
const
enum
signed
void
volatile
```

Explanations for each keyword follow in Sections 2 and 4. In addition, the identifier "entry" has been deleted from the list of reserved identifiers as it was never implemented by K&R or subsequent versions of C.

Each keyword is a reserved identifier; programs that currently use these keywords as variable names must be changed to compile under an ANSI-conformant implementation.

2. Data Types

Major language changes occur with respect to data types. The trend in the Standard has been to provide the language with stronger typing facilities.

Integers

The list of available integer types has been expanded to include "signed char" and the following variations:

```
unsigned int    signed int
unsigned long   signed long
unsigned short  signed short
```

"long int" and "short int" may also be used as variations of "long" and "short", respectively. As would be expected, the declarations:

```
signed x;
unsigned y;
```

may be used as shorthand for "signed int" and "unsigned int". All logical combinations for integer types are now allowed.

Whether or not a simple "char" is considered to be signed or unsigned is left up to the implementation.

While int is still the default type for variables and functions, at least one storage class (auto, register, static, extern) or type specifier must be present when declaring a variable. A declaration of the form:

```
x;
```

is no longer allowed and must be replaced with:

```
int x;
```

to compile.

Floats

The new type "long double" has been added for more precision. But like any "long" type, an object of this type is only guaranteed to be at least as large as a "double".

The type "long float" (a previous synonym for double) is now invalid. The only acceptable floating-point types are float, double and long double.

Structures and Unions

Member name spaces are now unique within structures and unions. That is, two different structures or unions may contain members with the same name without fear of conflict.

Structures and unions may now:

- * be assigned to another of the same type
- * be initialized when declared with the auto storage class
- * be passed as function parameters and return values

Enumerations

Already available in most compilers, enumerations have been added to the language. An enumeration is a way of declaring a set of integer constants. The declaration:

```
enum colours { RED, BLUE, GREEN };
```

would declare "colours" as an enumeration tag representing the integer constants RED, BLUE and GREEN. These enumeration constants are given integer values starting at 0 and increasing by 1 with each identifier.

An enumeration constant may be used wherever an integer is expected. The following is equivalent to the above enumerated type:

```
#define RED 0
#define BLUE 1
#define GREEN 2
```

Enumeration constants are not restricted to upper case, but upper case is a widely recognized convention for constants.

Variables may be declared to have enumeration type. The declaration:

```
enum colours x, y;
```

declares x and y to be integer variables capable of holding an enumeration constant of type "colours".

In practice, little or no checking is done to make sure enumeration constants are used, so the following assignments are equivalent:

```
x = BLUE;
x = 1; /* defeats purpose of enum */
```

Constant values may be directly assigned within an enumeration as well:

```
enum relation { EQUAL = 1, LESS_THAN = 2,
               GREATER_THAN = 4 };
```

If no value is specified for a given identifier, the constant is taken to have the value of the previous constant plus one.

The size of an enumeration type has been left unspecified; the implementation is free to store it in the most optimal fashion, providing that it always behaves like an int.

Void Type

The "void" data type has been added to indicate that an expression has no value. No variables can be declared with such a type, but expressions may be cast to void.

For example, the following declaration:

```
(void) printf( "hello world" );
```

specifically indicates to the compiler that the return value from printf (an integer) is to be ignored. As such, the following statement is illegal:

```
a = (void) func(); /* illegal! */
```

since the assignment operator expects a value to be returned for assignment.

Void pointers and void functions are discussed below and in Section 4.

Pointers and Arrays

Pointers are no longer synonymous with the "int" type. Pointers may only be compared with or assigned:

- * the integer value 0 (used to define a null pointer)
- * a pointer of the same type
- * a pointer of generic type (a void pointer)

Any other use of a pointer will generate a warning message upon compilation. Many assignment statements will require explicit casting of the right-hand values to avoid generating these messages.

A void pointer is a pointer that has no base type --- that is, it points to a type of unknown specification --- and is declared using the syntax:

```
void *ptr;
```

Indirection through a void pointer is not allowed; it must be cast to an appropriate pointer type first. Its main use is as a generic pointer.

Arrays with storage class auto may now be initialized. If specified, the size of an array must be an integral expression greater than zero.

Special Modifiers

The Standard makes available the two attributes "const" and "volatile" for use as type modifiers.

An object declared to be "const" cannot be modified (assigned to, incremented or decremented) by a program. Thus the following code is invalid:

```
const int x;
/* ..... */
x = 2; /* illegal! */
```

Initialization, however, is allowed:

```
const unsigned char masks[] = { 0x00, 0xff };
```

A const object (if it is of static storage duration) may be data that is put into read-only memory. Declaring such data with the const attribute allows the compiler to diagnose any attempts at modifying the data. Function parameters may also be declared as const to indicate that they are not modified by the function. This provides both extra documentation and, when function prototypes (described in Section 4) are used properly, consistent error-checking.

A "volatile" object is one that may be modified outside of program control. Memory-mapped I/O ports are a typical example. Declaring an object as volatile indicates that the compiler should always generate code to fetch the object's value from its actual memory location --- it may have changed since the last access by the program. (This disallows optimizations which could load the value into a register and possibly return erroneous results.)

```
volatile char *port1 = 0x00f3; /* ptr to I/O port */
```

```
while( *port1 & DATA_FLAG ) /* needs to be volatile */
    clear_io();
```

The const and volatile modifiers may be used (singly or together) in combination with any other valid type specifiers.

Pointers may also be declared to be const or volatile through the use of special syntax:

```
int const *a;
int *const b;
int *const *c;
```

In the example "a" is declared to be a "pointer to a const integer", whereas "b" is declared to be a "const pointer to an integer". The distinction lies in the placement of the const attribute. The declaration for "c" is even more confusing: it declares a "pointer to a const pointer to an integer". Consider the following statements:

```
a = NULL; /* ok */
*a = 0; /* error */
b = NULL; /* error */
*b = 0; /* ok */
c = &b; /* ok */
*c = NULL; /* error */
**c = 0; /* ok */
```

Because "a" is a pointer to a const int the value it points to may not be changed. Similarly, because "b" is a const pointer to an int it may not be modified, though the value it points to may. The pointer "c" may be modified, but not the pointer "b", though "b" (an integer) is modifiable itself.

The volatile modifier may also be used with pointers in conjunction with or separate from const.

Bit-fields

Bit-fields may now be of type int, unsigned int or signed int. Whether or not the high-order bit of an int bit-field is to be considered a sign bit is implementation-defined.

Vacuous Definitions

A vacuous definition consisting only of a struct or union specifier with a tag name is now allowed. Its purpose is to hide any outer declaration of the same name in the current block, as the definitions for "struct a" demonstrate in this example:

```
struct a
{
    int x;
};

int func()
{
    struct a st1; /* struct defined above */

    struct a; /* vacuous definition: it "clears"
               the current defn of struct a to
               make way for a new one */

    /* references to struct a now refer to the new
       definition within this block */

    struct b
    {
        struct a *y; /* refers to NEXT struct */
        int st2;
    };

    struct a
    {
        struct b *z;
        int st3;
    };

    st1.x = 1;
    st2.y = &st3; /* &st1 would give warning */
    st3.z = &st2;
}

/* old struct defn now back in scope */
```

Here the member "y" of "st2" is a pointer to the second "struct a", which is defined below it. If the vacuous definition

```
struct a;
```

had not been present, y would instead have been a pointer to the "struct a" defined in the previous (enclosing) scope level outside the function.

Conversions and Promotions

The Standard defines the integral promotions as follows: the char, short or bit-field types (with or without the signed or unsigned modifiers) may be used wherever an int is expected. The values will be converted to int if possible; otherwise they will be converted to unsigned int.

The usual arithmetic conversions used with most binary operators have been modified to reflect the new types available to the programmer. Of particular note: expressions of type float are not longer automatically converted to double for arithmetic purposes; such arithmetic may now be performed less accurately.

The Standard also specifies other rules regarding conversions. Where signed and unsigned integer values are concerned, the Standard now advocates value preserving as opposed to unsigned preserving rules: unsigned values are promoted to signed int if possible, otherwise they are promoted to unsigned int. Floating-point values must now truncate towards zero when converted to integral types. No rounding need occur when a double is demoted to float. Otherwise, the rules in K&R are unchanged.

Any program comparing or performing arithmetic on values of different types should be closely screened for possible changes in behaviour.

Minimum Type Limits

Any compiler conforming to the Standard must also respect the following limits with respect to the range of values any particular type may accept. Note that these are lower limits: an implementation is free to exceed any or all of these. Note also that the minimum range for a char is dependent on whether or not a char is considered to be signed or unsigned.

Type	Minimum Range
signed char	-127 to +127
unsigned char	0 to 255
short int	-32767 to +32767
unsigned short int	0 to 65535
int	-32767 to +32767
unsigned int	0 to 65535
long int	-2147483647 to +2147483647
unsigned long int	0 to 4294967295

Type	Minimum Precision
float	6 digits
double	10 digits
long double	10 digits

The Standard also specifies that these limits should be present as preprocessor macros in the header file `<limits.h>`.

3. Data Objects

Changes in this area have occurred mainly with respect to variable (object) linkage and initialization.

Initialization

Objects declared as either "static" or "auto" may be initialized by following the declaration with an equals sign, '=', and an initialization expression. External (inter-module) objects are discussed below in "Linkage".

If no initialization is given for a static object, all arithmetic types in the object are assigned 0 and all the pointers are set to NULL. If no initialization is given to an auto object, its initial value is undefined. These rules are unchanged from K&R.

All initializers for either static objects or auto arrays, unions and structures must be constant expressions.

Unions may now be initialized: the initialization value is assigned to the first member of the union.

The initialization expression for a scalar (integral, floating-point or pointer) object may optionally be enclosed in braces. Braces must enclose the initialization expressions for arrays, structures and unions. There can be no more initializers in an initialization list than there are objects to be initialized (there may be less, though, and any remaining uninitialized objects are handled as described above).

An array of "char" or "pointer to char" may be initialized with a string constant.

Linkage

In this section object refers to an object declared outside of any function.

The linkage of an object determines its scope within the program. An object with external linkage is known to all files in a program. An object with internal linkage is known only to the file in which it is declared. Current C compilers often differentiate between the two in incompatible ways, an issue which the Standard resolves.

An object is said to be defined if it includes an initializer. A defined object has internal linkage if the storage class static is specified; otherwise it has external linkage. An object can only be defined once.

Any object declaration without the extern modifier and without an initializer constitutes what is known as a "tentative definition" of the object. If an actual definition for the object is encountered in the same file, all tentative definitions are considered to be simple declarations referring to that object. Otherwise the first tentative definition is considered to be an actual definition with initializer equal to 0.

/* example drawn from the Standard */

```
int i1 = 1;          /* definition, external linkage */
static int i2 = 2;   /* definition, internal linkage */
extern int i3 = 3;    /* definition, external linkage */
int i4;              /* tentative definition */
static int i5;        /* tentative definition */

int i1; /* tentative def., refers to previous */
int i2; /* invalid -- linkage disagreement */
int i3; /* tentative def., refers to previous */
int i4; /* tentative def., refers to previous */
int i5; /* invalid -- linkage disagreement */

extern int i1; /* these are all valid references */
extern int i2;
extern int i3;
extern int i4;
extern int i5;
```

These complex rules provide the most flexibility and allow the majority of current C code to be compatible with the Standard.

The simplest way to declare an externally-linked object is to define it in one file (with or without initializer) and reference it in all others through the use of an appropriate extern declaration.

4. Functions

Additions to the language definition occur in the area of function declarations, function definitions and variable parameter lists.

Function Definitions

The Standard now allows the types of formal parameters to be specified within the actual function declaration at the start of the function definition. This new-style definition form more closely resembles languages such as Pascal and Modula-2:

```
int main( int argc, char *argv[] )
{
    /* ... */
}
```

If this style is used, a type must be specified separately for each formal parameter in the argument list. Mixing the new-style with the K&R-style in the same definition is not allowed.

This style is intended by ANSI to become the favored style, and a future Standard may disallow the K&R-style definition. For the moment, however, both styles may be used interchangeably.

Functions may also be defined as explicitly having no return values. Such functions are called void functions and are defined using the type "void":

```
void func( int a )
{
    /* ... */
    return;
}
```

Though the use of the return statement is allowed, such functions must not return an expression. If no type is explicitly specified, the function still defaults to int to retain compatibility with K&R.

Function Declarations and Prototypes

Function type declarations are also consistent with the new-style function definitions and may include a list of formal parameters. These parameters consist of type declarations with or without identifiers. Identifiers are cosmetic only and need only be included for readability. Some examples are:

```
int main( int, char *[] );
extern char *strcpy( char *dst, const char *src );
```

Note that declarations must be consistent as they will be checked by the compiler. Each declaration of a function should agree with all previous declarations in both the number and types of parameters.

The following declarations illustrate two special cases:

```
extern int func1( void );
extern int func2();
```

The first case explicitly declares that the function func1 does not take any parameters; that the parameter list is empty or void.

The second declaration states that no information is known on the number and types of any formal parameter. This is to provide compatibility with K&R.

A function declaration that provides the number and types of parameters is called a function prototype. The addition of prototypes to C allows for stricter type-checking by the compiler. When a prototype for a function has been declared, each subsequent call to that function is checked to make sure that the correct number of arguments has been supplied. As well, the type of each argument is compared with what was declared in the prototype. If different, the argument is converted to the required type as if it had been assigned to an object of that type. The default argument promotions (char and short to int, float to double) are not performed when a prototype has been declared. (Note: The default argument promotions are separate from the usual arithmetic conversions.)

If a function prototype occurs in the same file as the definition of that function, both the prototype and the definition must agree exactly if the definition is of the new style. In K&R-style definitions, the formal parameters are first widened by the default argument promotions and then compared to the prototype(s). If no prototype occurs in the file, the function definition itself serves as a prototype for the code following it.

Variable Parameter Lists

Certain C functions are designed to take a variable number of parameters. Unfortunately, some compilers use different schemes for handling such situations and what works in one implementation may not work elsewhere. The Standard therefore provides for the explicit declaration of such functions and portable facilities for handling them. A function that takes a variable number of parameters is defined by ending the parameter list (new-style only) with an ellipsis:

```
int printf( const char *format, ... )
    /* ... */
```

Thus the only thing known about printf is that it takes at least one parameter, the type of which is a pointer to "const char". Prototypes may also be declared in this fashion:

```
extern sprintf( char *dest, const char *format, ... );
```

The compiler will then make sure that each call to sprintf has at least two arguments, both of which are pointers to char.

The arguments themselves are accessed through the use of special macro facilities defined in the header file <stdarg.h>, part of the ANSI Standard Library.

5. The Preprocessor

The C preprocessor, long since recognized as an integral part of the language, has benefitted from a number of additions and clarifications in the Standard.

New Directives

The `$elif` directive has been added as a shorthand form of the `$else $if` preprocessor sequence.

The identifier "defined" is reserved during an `$if` or `$elif` so that:

```
$if defined( NULL )
$if !defined( TRUE )
```

are equivalent to:

```
$ifdef NULL
$ifndef TRUE
```

Also new on the list are the directives `$error` and `$pragma`. The former produces an error message at compile-time; the latter is implementation-defined in its use and effects.

File Inclusion

Besides the two allowable forms:

```
$include <fname1>
$include "fname2"
```

a third form:

```
$include fname3
```

is acceptable, provided that `fname3` is a macro which expands into one of the other two forms.

Macro Operators

Two new operators have been added for use within a macro replacement string. The `$$` (concat) operator concatenates two adjacent preprocessor tokens (a preprocessor token is any consecutive series of non-blank characters). The `$stringize` operator places the parameter following it in string form. For example, consider the following definition:

```
#define debug( s ) printf( "x" $ s " = %d00n", x $ s )
```

The following macro call:

```
debug( 1 );
```

expands to:

```
printf( "x" "1" " = %d00n", x1 );
```

which after string concatenation (see Section 7) gives the final result:

```
printf( "x1 = %d00n", x1 );
```

Program debugging through the use of macros has thus been made simpler.

Predefined Macros

Five new macros are predefined in the Standard, all of which are expanded to their appropriate values upon file compilation.

Macro	Expands to
DATE	current date
TIME	current time
FILE	current file name
LINE	current line number
STDC	non-zero value

The definition of `STDC` indicates an ANSI-conformant compiler.

None of these macros may be redefined by a program.

6. The C Library

A standardized library of routines aids the programmer and enhances portability. The Standard defines such a library, which is too large to describe here in any detail. The Standard Library is based on the library compiled by /usr/group, a UNIX user's group, with all the UNIX dependencies deleted.

The Standard Library also provides a set of standard library headers. These headers provide function prototypes for the set of routines that make up the library and define commonly-used macros. As well, the functions and their prototypes have been changed so as to be invariant to the default promotions --- all are declared using promoted types (such as `int` and `double`) for parameters. Thus parameters passed to a library function will always be of the same type, regardless of whether a prototype is in scope or not.

Macros may also be defined in a header file to take the place of actual calls to library routines. However, the library routines themselves must exist as the macros may be subjected to an `$undef` directive by the user at any time.

Among the most notable additions to the library are variable argument handling, numeric limits information, and locale (the current environment) information.

K&R library functions have also been converted to the new style and syntax, so that `malloc`, for example, now returns a "void *" as opposed to a "char **".

7. Miscellaneous

Numerous other minor changes and additions have occurred throughout the language:

- The escape characters `'\a'` and `'\v'` have been added for alarm (bell) and vertical tab.

- A series of special "trigraph characters" has been added as equivalents to ASCII characters which may not appear in the character sets of some countries; a trigraph is a three-character sequence starting with `"??"`:

Character	Trigraph
\$??=
X	??(
O	??/
A	??)
^	??'
a	??<
S	??!
a	??>
U	??-

- The suffixes `u` and `l` may be used with integer constants to specify unsigned and long values; both may be used together to specify unsigned long.

- The suffixes `f` and `l` may be used with floating-point constants to specify float and long double values.

- If the high bit of an octal or hex constant is set, it is considered to be unsigned.

- Adjacent strings separated only by white space are concatenated.

- The unary plus (`'+'`) has been added to force the evaluation of an arithmetic expression to occur before any others.

- A function can be called through a pointer using either the K&R-style syntax (`(*fp)()`) or the new-style `fp()`.

- External identifier length significance is still 6 characters with no case sensitivity (for compatibility with existing linkers).

- Internal identifier length significance is a minimum of 31 characters (case sensitive).

- Each macro function call is expanded only once, which prevents the definition of recursive macros.

- External identifiers beginning with an underscore are reserved for library usage.

- Identifiers beginning with an underscore followed either by a capital letter or another underscore are reserved for use as predefined macro names.

- Multi-byte character constants are allowed, though their values are implementation-defined.

- Hexadecimal character constants may be specified using `0x` followed by a series of hexadecimal digits (ex.: `'0xff'`).

Identifiers in current programs that are now reserved by the Standard will have to be altered to be portable across compilers.

References

American National Standards Institute, Inc.,
Draft Proposed American National Standard for Information Systems
- Programming Language C,
ANSI X3J11/87-221 (November 9, 1987).

American National Standards Institute, Inc.,
Rationale for
Draft Proposed American National Standard for Information Systems
- Programming Language C,
ANSI X3J11/87-219 (November 6, 1987).

Kernighan, Brian W., and Dennis M. Ritchie,
The C Programming Language,
Prentice-Hall, Englewood Cliffs, NJ (1978).

Plum, Thomas,
Notes on The Draft C Standard,
Plum Hall Inc., Cardiff, NJ (1987).

Lokalsamtal till USA

Många inom BBS-världen ringer direkt till USA för att köra modem. Det har visserligen blivit billigare på senare år men kostar fortfarande en slant. Enligt 1990 års telefonkatalog för Stockholmsområdet kostar det idag 5:35 per minut om man inte ringer mellan kl 10 och 22 vardagar.

Billigare blir det i regel om man använder ett datornät. Ett sådant är televerkets Datapak. Det kostar dock en slant att skaffa ett sådant konto och det är inte säkert att den lilla BBS:en i USA kan nås på det sättet.

Ett annat sätt är att få tag på ett användar-id på ett universitet eller högskola. Det kan ge tillgång till SUNET vilket i sin tur är kopplat till nät i USA. Genom denna ingång kan du nå en massa maskiner i USA, men det är främst Unixmaskiner och i regel inga PC BBS:er. Du måste i regel studera på någon högskola eller arbeta på en sådan för att få konto på en SUNET-ansluten maskin.

EUNET och USENET är nät inom Unix-världen som kan nås via medlemmar i Svenska Unixanvändarföreningen. Även om man kör uppringbar uucp kan man via konferenssystem USENET få program och källkod via konferenssystemet. Man kan dock inte på detta sätt logga in på en maskin i USA utan bara få vad de skickar ut i de olika nyhetsgrupperna. Skall man ha något speciellt är man hänvisad till brev dvs email, elektroniskt brev. Det finns vissa sk mailservers som man kan be att få saker och ting med automatik från.

i-Com

i-Com driver en nättjänst med hjälp av TYMNET nätet. TYMNET är ett paketförmedlande nät. TYMNET har modem som är anslutna till datornätet i USA varifrån du kan ringa ut till lokalsamtal inom USA. Finessen är att man via datornätet kan ringa lokalsamtal i USA och då blir det billigare. Jag fick tips om i-Com genom en notis en datatidning.

Det kostar 8 USD per månad att ha konto hos i-COM. Tidsavgiften är 25 USD (amerikanska dollar) per timme och till detta tillkommer 5 cent per samtal som rings ut och 15% tillägg på den lokala samtalstaxan. Dollarkursen är när detta skrivs ca 5.65 kronor vilket gör att den fasta kostnaden blir cirka 45 kronor per månad och nätkostnaden 2,35 kronor per minut. Till detta tillkommer alltså lokaltelefonkostnaden plus 15% och 28 öre för uppringningen.

I uppgifterna från i-Com står det att lokaltaxan varierar, men att det fn är cirka 10 öre per minut i Virginia.

i-Com har ett centralt system som du kan nå och varifrån du kan få uppgifter om telefonnummer att ringa till. Du kan dock få reda på nummer och sådant på annat sätt också, dvs genom listor som finns t ex genom att titta i Fidonet's nodelista.

Den högsta hastigheten som kan användas för att ringa in till i-Com i Stockholm är 2400 med MNP. Det finns enbart ett nummer i Sverige och det är 08-29 47 82. Möjligen är det ett gruppnummer. Vill du komma i kontakt med systemet så ringa upp numret med modem och ange för "A" som terminal-id och sedan när du får frågan om login så anger du "i-com" och som lösenord skriver du "host". Obs små bok-

stäver! Vill du prata med Idioma Communications i Stockholm som driver det hela här så ringer du 08-611 8402. De har också en BBS på nummer 08-611 8503.

Ingångar finns i Belgien, Danmark, Frankrike, Italien, Nederländerna, Schweiz, Storbritannien och Tyskland. I flera av dessa länder finns flera nummer. Nya linjer planeras för Helsingfors, Oslo och några extra för de länder som redan har portar.

Det blir alltså såvitt jag förstår billigare att ringa via i-Com om du vill ringa till USA med ett 2400 modem. Dessutom så slipper du problem med den amerikanska Bellstandard och modemstörningar kan bli mindre. Dock märks det att det går via ett nät så tecken som du skriver kommer ekas på din skärm långsamt. Så blir det dock också om du ringer upp själv direkt till USA.

Jag har på detta sätt ringt upp till några system i USA efter att i olika sammanhang sett telefonnummer i USENET. Det har fungerat bra utan problem. För att förenkla det hela gjorde jag en script i C-Kermit som ringer ut automatiskt och som frågar mig efter utringningsport i USA och lokalt telefonnummer innan mitt system ringer upp noden i Stockholm.

Ringde bland annat till en BBS som Motorola har i USA och de skrev att det nu fanns en BBS i Europa också i München. Räknade faktiskt ut att det verkade vara billigare att ringa till USA via i-COM än att ringa själv till Tyskland!

Anders M Olsson har skaffat konto på i-Com efter att jag berättade om i-Com i MSG. Anders hade först problem att ringa ut via i-Com och då uppmanade jag honom att fråga i i-Coms BBS.

Så här berättar Anders i ett brev till mig om lösningen och i-Com:

Jag frågade Sysop på I-Coms BBS och det visade sig att just i det riktnummerområdet i USA skulle man skriva en etta framför abonnentnumret

Alltså blev det D 1 683-4463.

Sen gick det mycket bra att köra. Jag testade att hämta en fil med Z-modem och fick en genomsnittlig hastighet på 225 tecken i sekunden.

Tidigare har jag aldrig kunnat ringa Sierras BBS på annat sätt än att jag har fått gå ner till 300 bps. På 2400 och 1200 har det bara haglat skräp hela tiden, och om jag har försökt att göra filöverföring så har det brutits efter någon minut. Jo en enda gång har jag lyckats koppla upp i 2400 bps på en kristallklar linje. Men det var som sagt bara en gång och aldrig mer...

Så det här med i-com verkar vettigt. Nu kan jag koppla upp och få en "klar" 2400 bps linje när jag vill. Och kombinerat med att priset inte är avskräckande så blir det ju nu ekonomiskt rimligt att hämta hem filer från USA.

<1789>
Bo Kullmar

Ordförklaringar:

BBS = Bulletin Board System, dvs modem-system av samma slag som vad vi i ABC-Klubben kallar för monitor (av historiska skäl).

Mailserver = ett program på en maskin som kan skicka tillbaka filer som begärs av användaren i brev.

EUNET = ett nät som tillhör vad som kallas EUUG, European Unix User Group. EUUG har nu ändrat namn till EurOpen.

USENET = ett konferenssystem som körs på tusentals maskiner i USA och Europa. främst Unixmaskiner.

USD = internationell kod för den amerikanska valutan dollar. (SEK är samma kod för vår svenska krona.)

Videotex

Det som jag i första hand använder videotex till är att söka efter telefonnummer. Det är mycket praktiskt och till och med numera billigare än att ringa televerket. Vi använder det också för att söka efter adresser till medlemmar i ABC-klubben.

Man kan dels få reda på namn till telefonnummer och sedan söka efter telefonnummer. Sökningen efter telefonnummer är uppdelad i företag och privatpersoner. En bra sak är att man inte behöver skriva riktnummer utan man kan även skriva ett ortnamn. Tyvärr måste man söka inom ett riktnumerområde vilket göra att man inte på en gång kan söka i hela Sverige.

Det kostar inget extra att söka i telefonkatalogen dvs kostnaden är bara den vanliga minutavgiften.

Idag provade jag på att söka i Gula Tidningen och även skriva in en annons där. Det fungerar så att om man går in och söker så kostar det 9 kronor som debiteras på televerkets faktura och sedan får man söka för det fram till kl 06.00 nästa dag.

Det är ganska bra att kunna söka med dator efter något, men man måste alltid börja och söka på ett ämnesområde. Att annonsera en gång kostar inget, men man måste veta de där numren för de olika ämnesområdena så då får man gå in där och söka och ta reda på numren.

I övrigt använder jag inte Videotex så mycket, men andra databaser som man kan nå på detta sätt kan vara bra, men då måste man ha ett speciellt konto. Detta gäller t ex bilregistret i Örebro.

Jag har inte den exakta kostnaden för att idag köra televerkets Videotex, men det har varit cirka 35 öre per minut kvällen och helger. Man har nu tänkt att införa 2400 modem, tidigare har man bara 1200 bps modem. Man ringer ett 020-nummer när man ringer till Videotex så man behöver inte betala rikstaxa.

<1789>
Bo Kullmar

PSpråk

(Text 1671) Jim Lundqvist <6649>

Ärende: BasicII/PC

Min BasicII som jag köpte av klubben fungerar bra på min AT så när som på grafiken. Kan jag få hjälp att köra mina gamla grafikprogram för ABC 806 utan att behöva ändra programkoden. Har försökt med olika inställningar i BASICINLSYS men får bara felmeddelande "no graphics option loaded".
mvh Jim

(Text 1672) Bo Kullmar <1789>

Ärende: BasicII/PC

Om du får felmeddelandet "no graphics option loaded" så får du se till så att du laddar grafiktionerna i BASICINLSYS. Det borde framgå ur manualen eller så kanske du fattar det om du tittar på textfilen BASICINLSYS eftersom där nog finns en kommentarssatt rad så att du bara kan ta bort "i" framför. Dock har de nog bara CGA och EGA grafik.

(Text 1686) Jim Lundqvist <6649>

Ärende: BasicII/PC

Tack för hjälpen med grafikprogrammen till ABC806! Får nu ut graferna på skärmen (EGA). Hur ska jag få ut dem på skrivare (seriell Facit 4509 matris- resp. HPlaserskrivare).

(Text 1697) Arne Lager <5798>

Ärende: Dvända mm-dd-yyyy till yyyy-mm-dd i Quick Basic?
MVH ARNE

(Text 1698) Lars Gjörling <6825>

Ärende: DATE\$ i Quick Basic v. 4.5
Datum\$ = Mid\$(Date\$,7) + "-" + Left\$(Date\$,5)
Print Datum\$

Monitor

(Text 459) Anders Franzen <5258>

Ärende: Underliga namn på directiroes!
Jaha, nu har jag vandrat lite i monitorn och till min fasa hittade jag helkryptiska namn på directories. Ta t ex "pc/src" eller "pc/sdn", om man med src menar "source" varför skriver man inte det?? och vad betyder sdn? Och det här med engelska verkar helsnurrigt! Jag ger upp, det är så svårt att hitta så jag struntar i det. Vad tusan är "genealogy"? Måste man ha lexikonet bredvid sig numera när man kör??? Annars funkade det mesta, tyvärr inte "cd..".

(Text 460) Kjell Svensson <5318>

Ärende: Underliga namn på directiroes!
CD .. fungerar med mellanslag mellan CD och punkterna
H/Kjell

(Text 461) Bo Kullmar <1789>

Ärende: Underliga namn på directiroes!
Nej, pc/sdn är inte alls source utan filer som distribueras via Fidonet som Per Sten skall lägga in. När de skickas ut så anges i vilket bibliotek de skall läggas upp och vi har tänkt att lägga upp dem i samma struktur här plus att länka in dem i de vanliga biblioteken också. Av praktiska skäl är det enklast att i första hand lägga upp dem i den struktur som anges från USA/Fidonet. Varför det heter sdn får du frågan någon som är mera kunnig i ämnet Fidonet. Tja, det var släktforskning det rörde sig om. När vi frågade om vi skulle ha svenska eller engelska namn så var det ovanligt få som svarade och det blev engelska. Självt hade jag kunnat tänka mig svenska namn också, men vi kan inte ha båda och vi kan knappast ändra nu eftersom jag tycker att det för mycket job. Dessutom förenklar det alltid i unix att ej ha biblioteksnamn som har åäö även om man kan ha det.

(Text 462) Nils Hansson <519>

Ärende: Underliga namn på directiroes/bibliotek
Visst står väl src för källkod?

(Text 463) Anders M Olsson <1019>

Ärende: Underliga namn på directiroes!
Betyder inte SDN Software Distribution Network?
Som jag har förstätt det är det ett sätt att sprida användarstöd programvara via Fido-net. Gimmicken är att man använder en speciell packningsmetod som ska garantera att filerna kommer fram i samma skick som de lämnar upphovsmännen till programmen. Dvs ingen ska kunna packa om programmen och lägga in virus eller göra andra ändringar.

(Text 465) Bo Kullmar <1789>

Ärende: Underliga namn på directiroes/bibliotek
Jo, jag har inte förnekast att src står för source dvs källkod, men jag kommenterade i första hand de där fidonetfilerna.

(Text 466) Anders Franzen <5258>

Ärende: Underliga namn på directiroes!
OK, men hoppas att när FIND-databasen blir klar och man gör där så ska man få se dels en kort rad som beskriver varje fil men dessutom en kort rad som beskriver varje directory. Bara nu inte den informationen också skrivs på engelska!!!
Kan inte ett directory ha två namn? Det vore fint om man med någon switch kunde välja språk.....!!!!

(Text 468) Egon Bosved <7723>

Ärende: Underliga namn på directiroes!
Kanske skall man ha antingen lexikon eller en ordlista med data-ord, så att man kan förstå ord som directory = katalog eller felaktigt ibland bibliotek och ord som switch = omkopplare? Egentligen bör man skriva på ett språk i taget. Programmerar man i t ex C, så är alla keywords, oj, namn på biblioteksfunktioner på engelska eller engelska akronym liksom globala variabler. Min autoexec.bat och config.sys är fulla av engelska ord. Sist men inte minst är alla bra böcker om programmering på engelska. Så varför inte skriva engelska namn på variabler och funktioner och konsekvent även i kommentarer. Sysslar man med datorer måste man ha ett hyggligt engelskt ordförråd. Program är ju internationella, nästan. :-)

(Text 470) Bo Kullmar <1789>

Ärende: Underliga namn på directiroes!
Nej, man kan inte ha så kallade symboliska länkar p bibliotek i detta Unixsystem som vi kör (dnix 5.3). Dvs man kan ej ha flera namn på ett och samma bibliotek.
Vad gäller find och sådant får Benny svara på det när han kommer till- baka från England (på söndag kommer han hem).

(Text 472) Bo Michaelsson <913>

Ärende: Underliga namn på directiroes!
I roten bör det finnas en textfil med förklaringar av de engelska orden. Din tanke är inte dum alls. Det gör det lättare för nyare datorister som nog har besvär med datorengelska. Med engelska kan alla få problem f ö. Det är ju ett ordrikt språk.

(Text 474) Per Sten <6366>

Ärende: Underliga namn på directiroes!
Ja, SDN står för precis det du säger. Det finns en annan distributionskanal från Fidonet som heter SDS. Står för Software Distribution System, även denna kommer att ligga in här. Tyvärr har jag haft allt för mycket annat för mig den senaste tiden och har inte hunnit lägga in något ännu.

(Text 503) Benny Löfgren <2615>

Ärende: Underliga namn på directiroes!
Varje fil kan beskrivas med upp till 2x64 tecken, som kan visas när man gör FIND och så småningom även när man gör DIR. Jag har inte tänkt på möjligheten att även beskriva bibliotek, men jag ska fundera på saken.

(Text 508) Anders Franzen <5258>

Ärende: Underliga namn på directiroes!
Ah, det låter mycket bra!

(Text 553) Tommy Kjellqvist <7685>

Ärende: kermi
Hur använder jag kermi i monitorn? har provat som förut med kermi, och även med kermi,s och kermi -s
Tommy K(TNSK)

(Text 556) Peter Mörtzell-Vincent <1343>

Ärende: kermi
SEND -k -i filnamn

(Text 559) Peter Isoz <2164>

Ärende: PROBLEM
Jag kan inte byta bibliotek i monitorn För-söker med CD PC Monitorn svarar: Kan inte byta bibliotek Tillfälligt fel eller gör jag något gale? Tacksam för hjälp
Hälsningar /Peter

(Text 560) Peter Isoz <2164>

Ärende: PROBLEM
Visst ja, små bokstäver va de.

(Text 590) Bo Michaelsson <913>

Ärende: Fotografi
Finns det något fotografi av det gamla monitorsystemet? Om inte så borde det tagas ett! Det har i alla fall arbetat sex år. Svartvitt, tack.

(Text 591) Karl Lindström <837>

Ärende: Fotografi
Borde vi inte samlas i klubblokalen på söndag för att dricka gravöl och ropa: Monitorn är död, länge leve monitorn!!!

(Text 593) Benny Löfgren <2615>

Ärende: Fotografi
Jo naturligtvis!!! Klockan 13.00 ska vi träffas och vederbörligen avtacka monitorn för trogen tjänst och hälsa monitorn välkommen! Den som vill och kan är naturligtvis välkommen till lokalen!
(Det kommer för övrigt av förställda skäl bli en del störningar i driften under söndagen, men jag hoppas på församlingens översende med den olägenheten.)

(Text 596) Benny Löfgren <2615>

Ärende: Söndagsmässan
***** S A M L I N G *****
Hallå gott folk!!! Jag vet att det är sent påtänkt, men det vore klart skoj om alla som har tid och lust kunde leta sig ner till klubblokalen på söndag klockan 13 eller så. Som bekant så tänker vi dra igång det nya systemet på heltid då, samt pensionera det gamla efter prick 6 års trogen tjänst. Detta är ett helt spontant och improviserat uppdrag, men försöka som duger...
Några kommentarer?
***** S A M L I N G *****

(Text 601) Stig C Holtzberg <4781>

Ärende: Gravöl
Tyvärr kan jag inte närvara vid gravölet över den gamla monitorn som betytt så mycket för mig här i de södra provinserna. Ofta har man där kunnat få snabb hjälp på blåbärsfrågor som verkat olösliga när man sitter här ensam. Jam mins som i går när jag med benägen telefonhjälp av Bo Kullmar försökte ta mig in första gången med en ABC800 och ett litet listat program i ABC-bladet. Ett hjärtligt tack till alla som startat och skött denna vidunderliga anläggning och tack till all som hjälpt mig under alla dessa år.
DEN GAMLA MONITORN ÄR DÖD **
LEVE DEN NYA MONITORN
Morfar.

(Text 602) Anders Anduren <7227>

Ärende: Gravöl
Stig C är inte ensam om sina upplevelser. Jag har haft liknande upplevelser i den 'gamla' basen. Alltid, jag menar alltid, har jag fått den hjälp jag be- hövt i min första kontakt med datorer. Skulle alla i världen vara så hjälpsamma som i ABC-klubben så skulle det inte vara några problem.....
Som sagt:
DEN GAMLA ÄR DÖD!
LEVE DEN NYE!
Anders i Tyresö

(Text 604) Benny Löfgren <2615>

Ärende: VÄLKOMMEN TILL NYA SYSTEMET
Nu har vi dragit igång det nya systemet på gruppnumret. Det gamla systemet har ingen linje ansluten längre. Vi kommer dock att öppna en linje från detta system så att det ska gå att köra ytterligare ett tag.

(Text 610) Bo Kullmar <1789>

Ärende: Gamla systemet
Det kommer att bli möjligt att nå gamla systemet via detta system. Dvs via ett kommando i monitorn. Det är en direkt kabel från tty21 i detta system till en 802a på gamla monitorn. Det kan då bara användas av en person i daget och är främst tänkt för de som vill läsa färdigt MSG och sådant i gamla systemet.

(Text 611) Börje Janson <4934>

Ärende: VÄLKOMMEN TILL NYA SYSTEMET
Kan du förklara lite närmare hur man skall göra för "att köra ytterligare ett tag" i gamla MSG.
Samtidigt vill jag säga ett stort GRATIS till alla oss som får förmånen att fortsätta i nya Monitorn, tack vare det enorma arbete som ett antal eldsjälar har lagt ned under sin fritid. Även stort TACK till er som under de gångna åren fixat så att allt har fungerat på ett närmast otroligt sätt, bl a med tanke på ganska jobbiga förhållanden.

(Text 624) Peter Mörtzell-Vincent <1343>

Ärende: Gamla monitorn
Hur kommer man till gamla monitorn (eller läser ev olästa brev)?

(Text 631) Benny Löfgren <2615>

Ärende: Gamla monitorn
Genom kommandot "OLDMON" som jag kommer att fixa i kväll.

(Text 663) Bo Kullmar <1789>

Ärende: Gamla monitorn
De gamla MSG-texterna finns i fil här under text/oldmsg tror jag det heter.

(Text 706) Benny Löfgren <2615>

Ärende: texten vid inloggning!
Om ni tröttnar på inloggningstexten, så prova CTRL-O ! Nackdel: Man riskerar att missa den eventuella nyhetsflash som kommer strax efter inloggningsmeddelandet.

(Text 765) Peter Mörtzell-Vincent <1343>

Ärende: Demokonto
Skulle man inte kunna införa något slags demokonto för att kunna visa blivande medlemmar ungefär vad man kan göra och från det kunna skicka brev till medlemmar?

(Text 766) Benny Löfgren <2615>

Ärende: Demokonto
Det finns möjlighet att ha demokonton i systemet, men vi kan inte idag begränsa användningen av msg annat än att stänga av det helt.

Datakom-munikation

(Text 7) Morgan Lantz <4359>

Ärende: x,y eller zmodem?
Vilken är bäst att använda må tro? Är något av dessa protokoll snabbare än dom andra?
Mvh Morgan Lantz.

(Text 9) Bo Kullmar <1789>

Ärende: x,y eller zmodem?
Så vitt jag fattar är zmodem snabbast, men testa gärna de andra också för det är nästan ingen som har gjort det.

(Text 10) Morgan Lantz <4359>

Ärende: Telix 3.12
Jag har nu tagit ned telix 3.12. Jag undrar hur man får åä och ö att ungera i telix! Vilke prg är bäst telix 3.12 eller procomm 2.4.2?
Mvh Morgan Lantz.

(Text 11) Bo Kullmar <1789>

Ärende: Telix 3.12
Med Sevenh, men jag vet inte om du behöver göra något i Telix för att få det att fungera. Tja, det är väl en smaksak vilket program som är bäst. Självt tycker jag att MS-Kermi är bäst.

(Text 13) Per Andersson <5581>

Ärende: Telix 3.12
Jag körde en 3.11 utan ändringar i Telix (tror jag). 7H fixade mina problem. (Om MS-Kermi hade ett integrerat dialingpaket så. Men jag har alltid kört båda växelvis)

(Text 14) Göran Eriksson <4988>

Ärende: Telix 3.12
Du kan ju ta hem TLXSW7.1.ZIP där jag satt ihop lite olika råd. För ABC-klubbens båda system kan du köra med full översättning eftersom de (ännu) inte använder ANSI.

Dessutom kan du ju konfigurera Telix för att skriva via BIOS och sedan använda Seven-Heaven eller likn. Detta ger dock inte IBM-versionerna av Å, ä och ö i "capture"-filer om du nu händelsevis skulle vilja ha det.

(Text 16) Benny Löfgren <2615>

Ärende: Telix 3.12

Problemet om man använder Telix egna translatabeller är att man gör teckenöversättningen INNAN man tolkar escapeskvenser. En redig bugg, skulle jag vilja säga. Tänk vad som händer om man skickar en VT100-sekvens för att positionera markören <ESC> Å x ; y H. Vänsterhaken översätts till svenska och det hela blir bara skrot! Annars är ju intern översättning att föredra tycker jag.

(Text 18) Per Andersson <5581>

Ärende: Telix 3.12

Tänk lite på din terminologi. ANSI har inte mycket med IBMs sätt att hantera skärmar och teckensat på, särskilt det senare, utom det att en del av sekvenserna råkar sammanfalla med sekvenser. På nåt sätt kan man uppfatta ditt inlägg som att IBM/PC teckensat skulle vara en standard. ANSI har ingen sån standard, och det kommer aldrig att bli nån heller.

(Text 31) Curt Gustavsson <3375>

Ärende: Telix 3.12

Jag kör Telix 3.12 utan 7H - Jag har ställt om i teckentabellen så att det blir riktigt med Å Å och Ö. Alltså teckentabellen i Telix - Ålt-W.

(Text 32) Benny Löfgren <2615>

Ärende: Telix 3.12

Det fungerar så länge man inte behöver använda escapeskvenser för att styra terminalen (cursoradressering osv), eftersom Telix det nötet gör tabellkonverteringen INNAN escapeskvensavkodningen sker. VT100 är ju rätt glad i escapeskvenser där Å (alltså vänsterhake) ingår...

(Text 74) Curt Gustavsson <3375>

Ärende: Telix 3.12

Jo, Benny har nog rätt! Men jag klarar mig än så länge med det jag kör. Intressant att följa Er andras resonemang - alltid lär jag mig något nytt!

(Text 84) Morgan Lantz <4359>

Ärende: proccomm.

Jo jag bara undrar hur jag skall få Å, ä och ö när jag tar ned texter från msg till min dator. När jag sedan läser så blir det inga Å, ä och ö! Fast jag har satt på 7H! Jag har 7h version 2.0! Måste man sätta värden i translate tabel i proccomm. Jag kör med proccomm 2.4.3!

Mvh Morgan Lantz.

(Text 85) Anders M Olsson <1019>

Ärende: proccomm.

7H påverkar överhuvud taget inte filer som laddas ner. Den ändrar bara tecken som kommer från tangentbordet och som skickas till bildskärmen.

Under förutsättning att du använder en editor som inte skriver direkt till bildskärmen, utan använder BIOS-anrop, kan du ha 7H påslagen när du läser texten i efterhand. Då blir det rätt. Eller så kan du ha 7H påslagen när du skriver ut filen på printer, då blir det också rätt.

Om inte något av detta skulle vara användbara alternativ kan du göra något program som omvandlar den nerladdade filen. Det finns tror jag ett par olika att välja på här i monitorn.

(Text 86) Lars Gjöring <6825>

Ärende: proccomm.

Ett enkelt alternativ är faktiskt att använda Proccomms Translate Table, som Morgan föreslog.

Denna gör ju precis det som 7H inte gör, nämligen tar hand om INKOMMANDE tecken från comporten och transformerar dem. Man skall då låta tabellen omvandla ascii-91 till 142, ascii-92 till 153 etc. Man får då direkt IBM-riktiga åäö... både på skärmen och i loggfilen. (Filer som överförs med Kermit, X-modem etc transformeras givetvis inte, endast vid ren ascii-överföring). Translate Table kan dock inte ersätta 7H, då den inte transformerar UTGÅENDE tecken. Man måste alltså då använda BÅDE Translate Table och 7H.

(Text 99) Lars-Göran Göransson <495>

Ärende: Serie ansluten FAX burk

Jag behöver en FAX burk som går att ansluta med RS232. Den behöver bara kunna sända FAX. Någon som har förslag? M.v.h L-G

(Text 101) Benny Löfgren <2615>

Ärende: Serie ansluten FAX burk
Det finns något som heter FAXBOX som klarar av det, men den var i alla fall för något år sen väldigt dyr. Annars finns det väl numera hela faxar som har RS232? Ett externt faxmodem är ett tredje alternativ, men då måste man skriva programvaran för att skicka text själv.

(Text 109) Dan Sundström <7597>

Ärende: Snabba modem

Jag är intresserad av att köpa ett snabbt modem (mer än 2400) eftersom jag har noterat att flertalet baser är bestyckade med modem som klarar höga hastigheter. Finns det någon som kan svara på följande förvirrade frågor? Är HST USRobotics egen standard, är det ett dylikt modem man ska skaffa? eller är det ett modem med V32 standard man ska ha? Jag har tyckt mig förstå att HST är snabbare än V32, i så fall borde det väl vara det modem som gäller. Är den här typen av modem Hayes-kompatibla och kan köras från tex Procomm? Är det slutligen någon som erfarenhet av sådana modem som kan berätta om våra teletenor klarar dom här hastigheterna eller blir det i realiteten så att man pga störningar blir tvungen att köra i 2400 med sitt svindyra modem?

Det var många frågor, hoppas någon kan svara på någon av dom!
Mvh Dan Sundström

(Text 111) Bo Kullmar <1789>

Ärende: Snabba modem

HST är en slags halv duplex standard som USRobotics har. Det finns texter på engelska om det i äldre nummer av ABC-Bladet. V.32 är 9600 och HST är en slags split speed 14000/300 och därför inte helt jämförbara. HST har dominerat kraftigt för det är billigare modem och för att folk köper sådana modem från USA billigt. Idag kommer nog V.32 standarden att dominera men det finns även ett HST modem som också har V.32, HST Dual Standard.

Alla modem av denn typen pratar i regel hayes och det är väl inget som skiljer dem från 2400 modem. V.32 modem har även felkorrigering och packing. Tidigare var det enbart MNP 4 eller 5, men numera även V.42 och V.42.bis. Det första är felkorrigering och det andra packing. MNP kommer från före-tagat Microcom, medan V.42/V.42bis är CCITT. HST-modemen försöker Dennis Bergström få godkända för anslutning till det svenska telenätet. De har dock ännu ej lyckats. Problemet nu är puls-ringningen som måste ske med separat relä och ett sådant finns ej i modem. De har även gjort andra förändringar i modem. Modemet blir dock ej billigt från DB. Det finns alltså många som har både HST och HST DS modem i Sverige och vad jag vet fungerar de bra. Fast är linjen dålig kan man inte köra med högsta "HST" hastigheten 14400.

(Text 114) Dan Sundström <7597>

Ärende: Snabba modem

Tack för ett utförligt svar! 14000 bps verkar klart lockande förutsatt att den höga hastigheten går åt rätt håll :-)
Jag roade mig att kika i en lista över Svenska databaser, där hittade jag 105 baser som hade HST-modem, 29 av dessa 105 uppgavs fungera även med V32. Ingen enda bas hade enbart V32. Med utgångspunkt från detta borde HST vara ett bra val (billigt, snabbt vanligt). Jag såg i en annons i Byte i dag att även HST-modemen har V42 standard.
Till sist en fråga: Finns det någon lag om att ett modem ska vara godkänt? jag menar, är det någon risk att personal från televerket kommer och dämmer en hammare i mitt modem?
Mvh Dan

(Text 115) Paul Pries <5322>

Ärende: Snabba modem

Tja, jag vet inte om jag hade satsat på ett HST modem.... Anledningarna till detta är minst två, nämligen:
1. HST är en så kallad företagsstandard, alltså inte en global standard överenskommen mellan olika teleadministrationer.
2. HST-modemen är inte godkända av svenska televerket.
Ett modem som inte är godkänt får inte anslutas till televerkets nät, men att gå så lång som att ta till hamre för att koppla bort ett icke godkänt modem är det nog ingen som gör. Däremot kan det kosta en slant! Skulle det vara så (jåkligt) att du skulle få fel på telefonen och ringer till vårt kära lurverk för att anmäla detta, och de skickar ut en reparatör som upptäcker ditt modem, kan följande hända:

1. Ditt modem kopplas ur.

2. Varesig felet berott på ditt modem eller på något annat får du betala den gängse felavhjälpningsavgiften, 2*440 pix + Allans påslag. Detta är minimiavgiften, det tillkommer sedan 440 pix per timme.... Värre än så här blir det nog inte. Nä, jag skulle nog satsa på ett godkänt V.32 modem! /Paul.

(Text 117) Dan Sundström <7597>

Ärende: Snabba modem

Det är nog riktigt att en sk global standard borde styra valet av modem. Mot detta talar att det är HST som tycks gälla! Om det i framtiden blir så att V32-standarderna tar över på bekostnad av HST så är det kanske tråkigt för oss amatörmodemare eftersom vi förlorar i hastighet!
De viktigaste parametrar som styr ett modemköp borde vara: Pris, hastighet och antalet baser som kan nås med modem. HST är kanske amatörmodemarnas standard? Det här Dual Standard modem som Bo Kullmar berättade om kollade jag upp för det verkar ju vara en heltäckande lösning, men priset låg runt 10-tusen skr, dubbelt så dyrt som enbart HST!
* Mvh Dan *

(Text 119) Bo Kullmar <1789>

Ärende: Snabba modem

HST-modemet kommer troligen med tiden att bli godkänt, men den versionen blir inte så billig som USA varianten.

(Text 121) Benny Löfgren <2615>

Ärende: Snabba modem

V32-V42bis kommer nog upp i samma hastighet minst som HST. Tänk också på att "styrkeförhållanden" mellan HST och V32 nog kommer att ändra sig eftersom det kommer billigare modem för V32 i rask takt. Enligt vad jag har sett av modem och specifikationer tycks V32 vara en klart bättre standard än HST rent tekniskt. Mitt råd för dagen är nog att vänta ett par månader eller så, det kommer ut många bra* modem på marknaden inom kort!

(Text 122) Nils Hammar <4341>

Ärende: Snabba modem

Jag har för mig att V33 är reserverat för att innehålla specifikation för 19200-modem, men jag tror inte att den standarden är utarbetad ännu.
När det gäller V32, så är det en fullduplex 9600, medan HST är en halvduple vilket i sig gör att det är mera kräm i 9600-modemet. Det kan man dock inte dra nytta av om man inte har ett filöverföringsprotokoll som går i "stereo". I BinkleyTerm 2.40 finns det ett s.k. Janus-protokoll som gör detta, men jag har inte provat det ännu. Det förutsätter dock att båda sidor har ungefär lika stor mängd att erbjuda varandra för att det skall fungera bra.

(Text 123) Peter Mörtzell-Vincent <1343>

Ärende: Snabba modem

Är det inte så att HST är ett slags split-speed? (Har för mig att det är Trailblazers som är halv duplex)

(Text 125) Bo Kullmar <1789>

Ärende: Snabba modem

Jo, HST är vänderbar split speed 14400/300 och TB är vänderbar halv duplex dvs det speciella TB protokollet.

(Text 127) Dan Sundström <7597>

Ärende: Snabba modem

Jomen HST har ju också V42bis och det borde väl betyda att HST har ett hastighetsövertag vid den sorts kommunikation som typ såna som jag ägnar oss åt? Men OK, V32 är kanske en bättre standard och det bästa är kanske att undersöka marknaden för att se vad ett sånt kostar innan man beslutar sig. Forfarande kvarstår faktum att det inte finns en enda bas i Svenska databaslistan där V32 krävs!
* Mvh Dan *

(Text 128) Dan Sundström <7597>

Ärende: Snabba modem

Trailblazer?? Hoppas att detta inte behöver vägas in för den som vill skaffa sig ett snabbt modem förutom HST och V32!! Det är tillräckligt rörigt ändå.
* Mvh Dan *

(Text 129) Henrik Stenquist <7731>

Ärende: Snabba modem, Trailblazers

Glöm dem om du inte vill ha en tillförlitlig förbindelse över Atlanten. Vi har dem på jobbet för att köra filer till huvudkontoret i Minnesota, men för en BBS-amatör är det inget att ha. De stöder bara MNPI-3 i vanliga läge och i det sk PEP-protokollet kan de bara prata med andra Trailblazers.

Fördelar: Hastighet, vi har i snitt 13000 baud över Atlanten med Crosstalk kommunikationsprogram och X-modem protokoll. Nackdel: Priset, Feletonverket vill ha +20 kr per styck! :- (Setup, relativt komplicerat att ställa in (över 100 S-register!!) Jag ska sätta upp en linä nu med Carbon-Copy och TB så får vi se vilken nettohastighet på filöverföring jag får, men som sagt: inget modem för BBS-bruk ännu.
Ingeniering /Henrik

(Text 130) Bo Kullmar <1789>

Ärende: Snabba modem

V42bis är ingen hastighet utan felkorrigering och packning.

(Text 135) Dan Sundström <7597>

Ärende: Snabba modem, Trailblazers

Ja, det var gott att höra att trailblazer inte var något för mej, det räcker gott med de två andra medentyperna att vela emellan!
* Mvh Dan *

(Text 136) Dan Sundström <7597>

Ärende: Snabba modem

Jo, jag menade att HST med V42bis borde vara snabbare än V32 med V42bis. Den där packningen förresten, en kvalificerad gissning är väl att den inte gör någon hastighetsförbättring på filer som redan är packade med ARC och dylikt?
* Mvh Dan *

(Text 137) Bo Kullmar <1789>

Ärende: Snabba modem

Skilnaderna mellan HST och V.32 är ju att HST är halv duplex och något snabbare i ena riktningen så det är nog rätt att HST är något snabbare.

Det är rätt ang. redan packade filer. Tvärt om lär MNP 5 kunna dra ner hastigheten i sådana fall, men V.42 bis lär enligt uppgift klara det bättre.

(Text 138) Kjell Svensson <5318>

Ärende: Snabba modem, Trailblazers
13000 Baud skall väl vara 13000 bps?
H/Kjell

(Text 139) Nils Hammar <4341>

Ärende: Snabba modem

Även Trailblazer har egentligen Split speed. Tekniken är multibärbar. Det protokoll som man skall se till att ha i sitt modem numera är V32, med felrättning/kompression MNP5 samt helst även V42/V42bis

(Text 140) Mattias Ericson <6615>

Ärende: Snabba modem, Trailblazers
Baud och bit/s är ju samma sak vid vanlig modulerung.

(Text 142) Anders Fransson <7544>

Ärende: Snabba modem

Enligt en notis i Industriell Datateknik, senaste numret, så lär det ha förekommit att olika leverantörers V.42bis inte gått att köra mot varandra. Det vore ju trist om det skulle behöva vara på det viset med en standard som är tänkt att bli ganska allmänt använd. Har någon haft liknande erfarenheter?

(Text 143) Per Andersson <5581>

Ärende: Snabba modem, Trailblazers
Och det har den väl inte? v22 och v22" kör ju t.ex. 600 baud, men 1200 och 2400 bps. Man kan troligen inte komma upp i så höga hastigheter utan att ha nån slags fas-modulerung.

(Text 145) Bengt Almén <6415>

Ärende: Snabba modem, Trailblazers
Vid kommunikation dator - modem (seriellt) så är bps och baud väl lika. Att det inte blir det på modem - modem komm. har med moduleringsförbandet att göra, är visst nåt elektriskt tror jag.

(Text 146) Paul Pries <5322>

Ärende: Bit & Baud.... Kan inte låta bli att kommentera, liksom de andra gångerna denna diskussion varit uppe...
Bps = Bitar per sekund, ett digitalt mått. Baud = Antal moduleringsstillfällen per sekund, ett analogt mått.
Enkelt eller hur?
/Paul.

(Text 148) Nils Hansson <519>

Ärende: Bit & Baud....

Så länge man använder den enklaste typen av modulerung, dvs att frekvensen ändras varje gång en bit ändras i data kan man man sätta baud = bps.
Baud är ett mått på hur snabbt två modem kan kommunicera med varandra, oberoende av hur informationen är kodat. Har man en smart kodning eller avancerad modulerung kan man få betydligt fler bitar per sekund före/ efter modulerungen än vad antalet baud anger.

Det är riktigt att mellan modem och dator är det bps man talar om (normalt har man ingen modulering där). Även snöret mellan datorn och modem har naturligvis en elektrisk begränsning i överföringshastighet (max baudrate).

(Text 171) Peter Mörtzell-Vincent <1343>
Ärende: MSKermit+PC-tecken
Hur ställer man lämpligast in MS-kermit (3.02) om man vill ha både 'svensk' ascii och pc-ascii samtidigt? Det gick automatiskt med den gamla kermiten (2.??). Som den är inställd nu verkar den 'sila bort' alla åttonde bittar (i den gamla fanns en över-sättnings-tabell). Vad menas med 'ASCII', 'Swedish', 'Latin' och 'Transparent' vid SET CHAR?

(Text 172) Bo Kullmar <1789>
Ärende: MSKermit+PC-tecken
ASCII är ASCII som den är definierad i USA standarden och omfattar alltså bara 127 tecken varken mer eller mindre. Swedish är 7-bits svensk ASCII. Latin1 är ISO 8859-1, dvs en 8-bitars standard. Transparent är raka rör och ingen förändring. Används om man vill ha IBM's ääö. Vill du ha både 'svensk' ASCII och IBM ASCII får du nog utgå från Trans-perant och sedan skriva egna konverteringar för att få den svenska. Ev. skall du sätta något med 8-bitar också, men jag kommer ej i håg det. Du får prova dig fram.

(Text 235) Mikael Lindroos <7410>
Ärende: COM 3-4
Finns det någon i detta avlänga land som kan tipsa mig om ett program, ev. med vidhängande kort, som gör det möjligt att an-vända COM 1,2,3 och 4 samtidigt t.ex. genom att adressera AT- datorns IRQ0-15 för COM3 och COM4? Snabba svar mottages med STOR tacksamhet!
Mvh, Micke L.

(Text 236) Stefan Andersson <7930>
Ärende: COM 3-4
Det finns ett kort som heter smartport och som säljs av ett företag som heter Omicron och ligger i stockholm. Kortet finns i två olika modeller, dels ett med 4 serieportar och ett med 8 serieportar. Kortet finns också för två olika operativsystem: MS-DOS och UNIX. Kortet levereras klara med driv- rutiner och liknade. Priset på dessa vet jag inte men jag har personligen använt sådana kort under 2 år utan det minsta problem (både under MS-DOS och UNIX). Kortet består egentligen av två stycken kort, ett utan för datorn och ett vanligt instickskort i datorn. Dessa två kort är förbundna med en flatkabel på cirka 20 cm. Kortet lämnar de övriga serieportarna i din dator helt opåverkade, utan använder sig av enbart interruptvektorn för COM3 och multiplexerar den. Hoppas att detta kort kan lösa dinna problem!
Hälsn Stefan

(Text 237) Bo Kullmar <1789>
Ärende: COM 3-4
Det finns ett program för änamålet i programbanken. Du kan säkert lista ut vad det heter. Det har något med 3-4 i namnet har jag för mig. Leta i /pc/comm.

(Text 271) Ferdinand Mican <912>
Ärende: Mac-kommunikation med svenska tecken
Jag testar en Mac SE med programmet Hayescomm II och saknar svenska tecken. Går det att få på samma sätt med filter-program typ 7H? I övrigt tycker jag inte om Macen heller, är väl förmodligen förstört av många år med PC:n.

(Text 275) Mattias Ericson <6615>
Ärende: Mac-kommunikation med svenska tecken
Hur kan du ogilla Maccen, den är ju överläsen PC på de flesta områdena (Jag studerar till dataguru på dagarna och använder PC då, för var dag som går ju mer PC-hatere blir man, iofs är det WP och SISS som jag hatar)
Hur som helst så tycker jag att du bör byta ditt komprogram till Microphone II 3.0, det är det bästa som finns, det näst bästa är väl MacTerminal.
Dessa program hanterar ÅÄÖ på ett korrekt sätt.
/Mattias

MS-DOS

(Text 10) Ingemar Forsmark <7080>
Ärende: Loggfil (dosvariant)
Jag vet att det går att skapa en loggfil direkt från autoexec men har glömt hur. Jag hade tänkt att tanka ned den info som kommer från viruschecken till en dosfil med ett kommando detta borde se ut så här nånting "SCANRES > C:\VIRUS.LOG" vad som saknas är att inte skriva över den gamla filen utan att öppna och lägga till
---** HUR GÖR MAN **---

(Text 11) Einar Eriksson <1720>
Ärende: Loggfil (dosvariant)
Man skriver väl SCANRES >> C:\VIRUS.LOG".

(Text 12) Ingemar Forsmark <7080>
Ärende: Loggfil (dosvariant)
Jepp såvareju (det var för enkelt :-). Tackar.
(Text 16) Magnus Frostlid <7755>
Ärende: Loggfil (dosvariant)
Enkelt, du sa "SCANRES > C:\VIRUS.LOG". Vlj istället "SCANRES >> C:\VIRUS.LOG". Observera, TVA ">".
Mvh

(Text 18) Claes Heland <6032>
Ärende: CONFIG.SYS
Hej, Är det någon som vet hur man omdirigerar utskrift som sker av devices man installerar i config.sys (tex. till > nul, som går att göra i autoexec.bat) eller är det bara i .bat-filer det går att göra så?
MVH /Claes

(Text 19) Paul Pries <5322>
Ärende: CONFIG.SYS / omdirigering...
Du gör lika dant dvs med ">". Tänk dock på att vissa program skickar sina texter till stödort istället för till stdoutput! Det brukar förvisso gå bra att omdirigera den också.
/Paul.

(Text 21) Nils Hansson <519>
Ärende: CONFIG.SYS / omdirigering...
Hur gör man omdirigeringen i MS-DOS av v stddor. I unix gör man 2>&l tex men det gick inte i MS-DOS.

(Text 22) Paul Pries <5322>
Ärende: CONFIG.SYS / omdirigering...
Öh, Jag har för mig jag har sett det någonstans men kan inte hitta det just nu. Jag skall återkomma med det för det är ju en ganska intressant fråga. 2>&l ger ju bara en koppling av stddor till stdoutput och så länge stdoutput inte är omdirigerad har det ingen verkan. Jag tror iofs inte att DET fungerar i MessDos...
/Paul.

(Text 23) Conny Westh <7433>
Ärende: MS-DOS 5.00
Nu har jag fått reda på att 5.0 är ute hos betastare här i sverige har för mig att ämnet behandlades här i MONITOR för en tid sedan, men är det någon som har testat denna betaversion av klubbens medlemmar eller som VET något konkret om v 5.0.....?????

(Text 25) Nils Hammar <4341>
Ärende: MS-DOS 5.00
Jag har provat en äldre beta litet grand, och den är ganska OK. Det är en hake som jag anser vara besvärande, och det är att man inte kan köra novell under denna DOS-version. Detta beror på att Microsoft håller på att bygga ut DOS:et så att det inte bara skall stödja 80x86-serien. Det är alltså förberedelser för detta. Annars är minnesbehovet mindre än vad DOS4 har. Systemfilerna är dessutom så pass små att man kan installera DOS5 utan att formatera om disken, om man har DOS 4 installerat! Man behöver bara skriva SYS C:, eller var man nu vill lägga filerna. DIR-kommandot är också förbättrat, så att man nu får en summa på alla filernas storlekar, förutom informationen om ledigt utrymme på disken. (Detta är en funktion som jag har saknat) Dessutom tillåter DOS 5 att det laddas i extended memory under förut-sättnin att man laddar HIMEM.SYS (Varför detta program? Andra program fixar extended memory ändå.)
Någon Windows har jag inte provat speciellt, men det kommer väl så småningo. En effekt/bugg är visst att man KAN få problem om man bootar om i fel ögonblick, men det är nog bara ett beta-fel. Detta är vad jag har erfart av DOS 5.

(Text 26) Stefan Berg <216>
Ärende: Virus
Jag tycker att det är nervöst, det här med virus. Det värsta är att hitta ett lämpligt program som säkert detekterar virus, utan att själv vara ett!
Det ligger ju ett antal programpaket under pc/utl/virus - är det något av dessa som kan rekommenderas framför andra? Personliga erfarenheter?

(Text 28) Anders Wedebrand <5960>
Ärende: Virus
Jag var för en tid sedan på virusseminarium som hölls av Alan Solomon. I kursavgiften ingick ett ex av Dr Solomon's Anti-Virus Toolkit. Denna ger (tillsammans med vad som sades på seminariet) ett säkert intryck. Detta är dock ett kommersiellt program som kostar ca 850:- + moms.

En sak jag lärde mig på seminariet var att det är lönlöst att försöka hindra virus från att komma in i systemet (med någon form av utility, alltså). Det enda som hjälper där är försiktighet, som t.ex att aldrig stoppa i en icke skrivskyddad diskett om det är möjligt att undvika, att bara ta emot disketter och program från pålitliga källor (vad som är pålitligt får man avgöra själv).
Det man istället bör lägga ner energi på är att snabbt upptäcka ett virusangrepp för att förhindra spridning och större skada. Ett hjälpmedel för detta är sk "scanners" som söker efter ett virus signatur i filerna. Ett annat hjälpmedel är ett checksumme-program, som räknar fram en checkskumma för filerna, för att det ska vara möjligt att upptäcka om en fil förändras (vilket kan tyda på att ett virus har kommit in i systemet).
mvh Ankan

(Text 29) Ulf Johansson <4560>
Ärende: MS-DOS 5.00
Har testat med Dos 5.0 Beta 2.24 och 3.33 ett längre tag nu utan större problem. Windows 3.0 går utmärkt ihop med 5an. Enda problem jag märkt är att jag nog gång fått boota om datorn vid uppstart men det kan ju bero på mycket.....
Annars är 5.00 superb.

(Text 30) Conny Westh <7433>
Ärende: Virus
Jag har provat att använda viruskontroll-programmet från Dr: Solomon man skall INTE lägga in det på hårddisken för där kan det naturligtvis själv smittas av ett listigt virus, men om man gör två saker:
1 - skrivskyddar alla disketter som man inte ständigt uppdaterar...
2 - gör i ordning en bootbar diskett med checksummeprogrammet på
3 - att man alltid kör denna diskett genom att göra en HÄRD-BOOT (dvs stänger fysiskt av maskinen med strömbrytaren, och på igen) så snart så snart man har installerat ett nytt program (man bör köra det vid minst ett tillfälle i veckan om man jobbar varje dag med systemet).
4 - Lita ALDRIG på att kommersiella program är virusfria, det är en ren myt (ta svenska Commodore som avskräckande exempel).
5 - PD/Freeware/Shareware är inte mer virusspådd än vad kommersiella program är om man får dessa via etablerade/pålitliga "kanaler" som exempelvis ABC-Klubben. Det är inte lönt att använda ett program som letar efter signaturer av vissa välkända virus, de har man ingen nytta av på nya virus och då invagga man i en falsk säkerhet som kan bli förödande. Det är bättre att använda ett eller gärna flera olika checksumme program (som beräknar checkskumma på olika sätt) eftersom vissa virus klarar att bevara samma checkskumma på infekterade program.
Eftersom Dr: Solomon's virusdödarprogram har blivit så stort på marknaden så innebär ju det att de som konstuerar virus kan ta hänsyn till detta och hitta kryphål i Dr Solomons Anti Virus Toolkit.
M.V.H. Conny Westh

(Text 31) Conny Westh <7433>
Ärende: MS-DOS 5.00
Vad jag saknar är möjlighet att köra befintliga DOS-program med mer än 640 k. och möjlighet att starta upp flera processer som skall kunna gå parallellt om DOS 5 eller DOS 6 kommer att klara det finns det INGEN som helst anledning för mig att byta till OS/2. Det viktigaste är dock möjligheten att köra programmen utan minnesgränser.
M.V.H. Conny Westh

(Text 32) Ulf Hedlund <6988>
Ärende: MS-DOS 5.00
DOS 5 har inga sådana utökningar, och det är väl tveksamt om det kommer några under DOS. Så länge man tvingas göra DOS bakåtkompatibelt är det antagligen omöjligt att förbättra det.
Varför sträva emot OS/2? Nu när minnet börjar bli billigt (< 500 kr/Mb) är ju inte det något större problem. Och jag antar att du redan har något extraminne eftersom du vill använda det över 640K-gränsen?

(Text 34) Bo Kullmar <1789>
Ärende: Virus
Många använder programmen från McAfee Associates. De är shareware och det finns flera versioner. Den vanligaste är SCAN.EXE som brukar finnas i ett arkiv som heter SCANVXX.ZIP där XX är versionnumret vilket just nu är 67. Detta program uppdateras ofta.
I serien ingår även CLEANP, NETSCN och VSHLD alla med versionsnummer efteråt. Vi skulle kanske lägga dessa i ett underbibliote som heter McAfee (fast då får väl en del problem med stora och små bokstäver...).

(Text 35) Bo Kullmar <1789>
Ärende: Virus
Har nu läst den text som jag kommemerar ordentligt och hävdar att det visst är lönt att använda program som letar efter signaturer av kända virus under förutsättning att man använder ett program som är uppdaterat. Ofta är det dessutom gamla kända virus som uppträder.
Det viktigaste är enligt min mening att hålla sig till "skäkra" källor. Detta system borde kunna räknas som säker källa minus programmen i upload biblioteket!

(Text 36) Nils Hammar <4341>
Ärende: Virus
Det verkar vara en god ide. Men varför inte bara kalla det "mcafee" Det verkar enklare.
Det brukar vara aktuellt att ladda ner hela samlingen när den uppdateras.

(Text 37) Nils Hammar <4341>
Ärende: Virus
Jag funderar faktiskt på alla de virus som har dokumenterats. Är något utrotat? Det vet man inte säkert förstås, men det skulle vara kul om man fick dokumenterat att vissa virus är utrotade.

PCprog

(Text 10) Morgan Lantz <4359>
Ärende: fax modem!
Jo jag bara undrar vad ett fax modem kostar? Det finns bla hidem 2400 som klarar av att sända fax. Klarar dessa modem även av att ta emot fax? Hur fungerar det att ta emot fax i så fall?
Mvh Morgan Lantz.

(Text 11) Göran Sundqvist <1255>
Ärende: fax modem!
Hidem liksom Dos i Los's modem klarar bara att sända fax. Dos i Los kostar 1995 + MOMS. Vi har köpt ett 10-tal utan problem.

(Text 12) Martin Björkman <3854>
Ärende: fax modem! HIDEEM 2400
Jag kör med ett sådant modem och det funkar mycket bra. Det är mycket praktiskt att kunna faxa och man kan faktiskt faxa både grafik-filer och WordPerfect dokument. OBS dock att det inte går att ta emot fax. Dessutom går det bara till s.k. grupp 3 faxar - vissa äldre faxar som går långsammare funkar modemmet ej mot.
Hälsning Martin / Göteborg

(Text 13) Bo Michaelsson <913>
Ärende: fax modem! HIDEEM 2400
Fungerar också fax-modemet som vanligt modem? Måste man ha speciella program för att de ska kunna hantera fax?

(Text 14) Martin Björkman <3854>
Ärende: fax modem! HIDEEM 2400
Jovist dety funkar som ett helt vanligt Hayes-modem. När man ska köra fax använder man en programvara BITFAX som medföljer och med vars hjälp man kan editera text, sända försätsbladet, skicka grafikfiler och lägga in faxteletnummer i en särskild fil. Man får även en logfil på alla sända fax. Vidare kan man programera in tidsbestämd sändning om man vill att ett fax ska gå iväg automatiskt en särskild tid. MVH Martin

(Text 15) Bo Michaelsson <913>
Ärende: fax modem! HIDEEM 2400
Hm-m, mycket intressant. Vem säljer modemet?
Tack för svar, förresten.

(Text 16) Göran Sundqvist <1255>
Ärende: fax modem! HIDEEM 2400
Dos i Los säljer för 195.-!
tel. 0657-10532 / 34 (fax 10533)

(Text 40) Paul Pries <5322>
Ärende: com-portar, fler än 2.
Har för mig att jag någonsin sett en drivrutin för fler än 2 samtidiga komportar. Är det nån som kommer ihåg vad den hette och var den fanns?
/Paul.

(Text 44) Anders Johnson <4001>
Ärende: com-portar, fler än 2.
Titta på COM34.ZIP. Det finns i gamla monitorn.

(Text 41) Morgan Lantz <4359>
Ärende: hur formatera 720 kb på en 3,5 tums disk?
Hur gör jag för att formatera 720 kb på en 3,5 diskett? Jag skall väl skriva format fast vad skall jag ha för parameter för att formatera 720 kb! Det bör väl gå att göra precis som man formaterar 360 på en 5,25 diskett med format a: /4 är det väl!
Tacksam för svar!
Mvh Morgan Lantz.

(Text 45) Anders M Olsson <1019>
Ärende: hur formatera 720 kb på en 3,5 tums disk?

FORMAT A: /N9 /T:80

(Text 46) Anders M Olsson <1019>
Ärende: com-portar, fler än 2.
Vad ska du använda för programvara mot COM-portarna?

I de flesta fall är det ganska meningslöst att lägga in "drivrutiner" för att kunna kora mer än två COM-portar. Det enda en sådan rutin kan göra är att lägga till BIOS-stöd för COM3: och COM4:.

Men eftersom de flesta program struntar i BIOS och går ut på serieportarna direkt har man ingen nytta av att BIOS kan supporta fler COM-portar.
Det finns kanske något undantag. Ibland finns det program som tittar i adress Tabellen i BIOS innan COM3: o COM4: anropas. Saken kompliceras ytterligare av att IBM aldrig har supportat mer än två COM-portar i sina vanliga PC och AT. (I PS/2 fungerar det på ett HELT annat sätt.) Det finns därför ingen standard för hur man ska koppla in COM3: o COM4:. En vanlig metod som tillämpas av en del korttillverkare går ut på att COM3 använder samma interruptkanal som COM1 och COM4 använder samma som COM2.

Det säger sig då själv att man inte kan använda COM1 och COM3 samtidigt. Om man t ex sätter en mus på COM1 kommer man aldrig att kunna använda COM3. Men det finns också andra sätt att koppla in COM3 o 4. Man kan t ex använda de höga interruptkanalerna i en AT. Men då är det inte alls säkert att man kan få sin programvara att fungera mot en sådan serieport.

(Text 48) Sven Wickberg <1384>
Ärende: hur formatera 720 kb på en 3,5 tums disk?
Jag har just i dag måste repetera det där: I DOS 4 kan man tydligen även skriva FORMAT A: /t:720.
Annars är det väl så att om man har en diskettstation som bara kan ta 720 K så formateras i 720 om man begär FORMAT - väl?

(Text 52) Bo Kullmar <1789>
Ärende: hur formatera 720 kb på en 3,5 tums disk?
Enklast är att använda PCTOOLS för där man kan säga till att man vill ha 720 KB användarvänligt. Detta gäller alla versioner av PCTOOLS och gäller då den inbyggda formateringen i själva programmet.

(Text 53) Bo Kullmar <1789>
Ärende: com-portar, fler än 2.
Allt är flyttat hit och den finns i /pc/comm här men den heter ej COM34.ZIP utan com34.zip!

(Text 60) Paul Pries <5322>
Ärende: com-portar, fler än 2.
Programvaran som skall köras är ett övervakningssystem som tar 2 portar, plus att man nu vill fjärrstyra hela eländet med PC-Remote (eller vad det nu hette...). Problemet blir ju, som jag ser det, att man behöver 3 samtidiga portar. Den stora frågan blir givetvis om detta låter sig göras?

Vi får väl se hur det går. Att använda de höga interruptkanalerna verkar vara en intressant lösning, om man får det att fungera.

/Paul.

(Text 62) Anders M Olsson <1019>
Ärende: com-portar, fler än 2. PS/2
I PS/2 har man standardiserat adresserna för upp till 8 COM-portar. Dessutom så fungerar interrupthanteringen på ett helt annat sätt. Det gör att man kan ha flera COM-portar på samma interrupt och använda dem samtidigt. Något som inte går i en vanlig PC eller AT. Där måste varje port ha en egen interruptkanal.

(Text 63) Anders M Olsson <1019>
Ärende: com-portar, fler än 2.

Ja, då är jag helt övertygad om att du inte har någon glädje av det program här i monitorn som ger BIOS-stöd för COM3 o 4.
Eftersom du vill använda tre COM-portar samtidigt måste du se till att alla tre verkligen är installerade på OLIKA interruptkanaler och att den mjukvara du tänker använda kan ställas in på de aktuella adresserna och interruptkanalerna.

(Text 70) Bo Michaelsson <913>
Ärende: hur formatera 720 kb på en 3,5 tums disk?
Norton går också bra.

(Text 81) Benny Löfgren <2615>
Ärende: Samma inlägg i flera möten
Jag har nyss besvarat din fråga i mötet DivData, och så hittar jag en ny kopia i nästa möte!! Det hör normalt inte till seden i MSG att skriva samma sak i mer än ett möte, så var vänlig och observera detta. Anledningen är uppenbar för alla som sitter och kör med något annat än lokalaxa. Dessutom undviker man på så sätt att få två parallella diskussioner om samma ämne, samt att folk svarar på saker och ting som man annars kunnat se att det redan finns svar på.

(Text 83) Morgan Lantz <4359>
Ärende: Samma inlägg i flera möten
Jo jag kan förstå att man skall hålla sig till ett och samma möte med ett inlägg. Fast det kava vara svårt att veta vart man skall skriva inläggen ibland. Jag kör själv med rikstaxa. Eftersom jag sitter på västkusten och monitorn ligger i Stockholm eller om det är strax utanför stockholm! Jag skall i fortsättningen hålla mig till ett möte med ett inlägg. Du har rätt i att en diskussion skall hållas i ett möte. Det är ju inte bra med parallella diskussioner! Som du påpekar!
Jag vill även tacka för svaret ang formatering i mötet divdata!
Mvh Morgan Lantz.

(Text 89) Anders Magnusson <6778>
Ärende: Samma inlägg i flera möten
Vill man "fiska i flera vatten", så att säga, kan man skriva sitt inlägg i ett möte och sedan kortfattat beskriva frågan i något annat möte med en hänvisning till ursprungsmötet. Kan ju vara bra om man vill få personer att läsa inlägget även om dessa inte är med i ursprungsmötet. Metoden bör dock bara användas i undantagsfall tycker jag.
Anders

(Text 112) Peter Isoz <2164>
Ärende: fax modem!
Vad gäller egentligen för faxmodem. Är det ett speciellt modem eller räcker det med programvaran? Vad gäller om man nöjer sig med att sända? Det är kanske Bell standard (det går ju bra att sända fax till USA trots att den "vanliga" modemstandarden är olika. Tacksam om någon redde ut begreppen.
Hälsningar /Peter

(Text 113) Kari Kiviranta <7841>
Ärende: fax modem!
Faxmodem kräver också ett FAX-program. Endast ett program räcker inte du måste ha ett speciellt modem och program. Det finns modem+program som endast sänder FAX om du vill klara dig utan att kunna ta emot. Om du köper senaste BIT så finns det bra artikel om det i den(reklam)

(Text 141) Einar Eriksson <1720>
Ärende: Databas program för PC
Jag skulle vilja veta lite vad det finns för bra Databas program för PC. Priser? Användarvänlighet? Prestanda? Finns det förresten demoversioner att få tag på?

(Text 142) Anders Stålhandske <7203>
Ärende: LIST SKYDD
Finns det något program som tar bort list skydd som är gjorda i basic program????
M.V.H *Anders*

(Text 143) Bo Kullmar <1789>
Ärende: LIST SKYDD
Vilken BASIC och vilket system? Det finns förresten ett möte för BASIC.

(Text 144) Sven Wickberg <1384>
Ärende: Databas program för PC
Vad menar du med "databas"? Om det är ett registerliknande program du vill ha kan jag varmt rekommendera PC-FILE. Vill du däremot ha ett program att köra en databrevlåda på får någon annan svara vad som finns.

(Text 146) Anders Stålhandske <7203>
Ärende: LIST SKYDD
Basicon är GWBASIC och systemet MS-DOS.

M.V.H *Anders*

(Text 147) Anders M Olsson <1019>
Ärende: LIST SKYDD
Tag hem UNPROTEC.ZIP som ligger i katalogen /pc/util/develop

(Text 148) Bo Michaelsson <913>
Ärende: Databas program för PC
Ett databasprogram som är riktigt vettigt är SPCS-Register. Det är ett lågprisprogram med en bra, välkänd bok som stöd. Kostar 500-600 kr?
Vill man kosta på sig mera så kan man välja dBASE IV eller möjligen dBASE III. De kostar mycket - 5000 - 15000 kr - beroende av programversion och möjligheter. Med dBASE kan man skriva egna program i det speciella dBASE-språket. Om man sedan trötnar på den miljön kan man köpa en kompilator för dBASE-kod. Kompilatorn heter Clipper och ger program som körs som andra program (.exe-typ) i PC.

(Text 154) Morgan Lantz <4359>
Ärende: Skadade sektorer på hårddisken!!!
Jo jag har haft problem med min hårddisk ett bra tag nu. Jag har ca 10 kb skadade sektorer. Jag har frågat här tidigare, om jag kan göra något åt dessa skadade sektorer! Kan jag flagga bort dessa skadade sektorer eller på något annat sätt se till att hårddisken inte läser eller skriver på dessa sektorer som är skadade?
Kan man använda NDD, på något sätt för att åtgärda detta lilla problem som jag har haft ett bra tag nu! Sedn lär man kunna förebygga och åtgärda fel på sin hårddisk med hjälp av pc-tools 6.0. Stämmer detta? Vad kostar pc-tools 6.0 i så fall?
Det konstiga var när jag installerade msdos 4.01 och formaterade om hårddisken i samband med installationen av dos 4.01. Så blev det plösslöst färre skadade sektorer. Från att ha varit 100 kb skadade sektorer till 10 kb skadade sektorer. Vad kan detta bero på?
På någon sätt så har antalet skadade sektorer ökat tidigare, när jag har kört format c: /s/v!
Jag hoppas att det är någon som kan svara på dessa frågor. Jag är tacksam för svar här i MSG!
Mvh Morgan Lantz.

(Text 165) Jan Sundström <7804>
Ärende: Skadade sektorer på hårddisken!!!
Försök att få tag på augustinumret av BYTE från i fjol (aug. -89). Där finns en utmärkt artikel som heter Hard Disk Maintenance Software och som behandlar just ditt problem. Där behandlas bl. a. olika orsaker till att spår kan "driva iväg" på disken (olika temperatur vid skrivning och läsning, t.ex.), samt ett par program som kan vara till hjälp: SpinRite samt Disk Technician Advanced. Dessa program kan dels vara till hjälp att diagnostisera fel på disken (DOS varnar ju aldrig för att sektorer är svåråstata) dels också kan de formatera om hårddisken.

"Hårdformatering" av disken är väl inte så märkvärdigt, men de här programen kan läsa in ett spår som kräver många omförsök, formatera om detta spår och lägga tillbaka data som fanns där! Praktiskt! Kanse kan man även använda något program från Norton eller senaste versionen av PC-tools för samma sak? Någon påstod det, men jag har inte kollat upp det.

(Text 198) Nils Hansson <519>
Ärende: Response el dyl
Har någon eller finns det här en liten programsnutt som läser av senaste tangenttryckning och lämnar dess kod som exit-status? (att används i batchfiler). Jag har sett en liten assemblersnutt på några rader som fixade det men jag kan inte hitta den igen.

(Text 200) Claes Börjesson <6928>
Ärende: Response el dyl
Det du vill ha finns som PD och heter BATQUES, jag skall skicka över filen. Norton har även ett liknande program som heter ASK.
Claes B.

PChård

(Text 13) Sven Wickberg <1384>
Ärende: Två frågor om HP-laserjet
Jag skulle hjälpa en kollega med anslutningen mellan HP-I och en IBM-dator ganska ny, ver 60 tror jag. Han visade mig först att poster från PC-FILE inte fick med sig (på skärmen synliga) ää vid utskrift på laserjeten. Detta problem behandlas i nästa inlägg. Men medan jag höll på att experimentera (med 7H och FILTER etc) så fann jag att laserjeten över huvud taget inte skrev ut om jag inte, efter överföringen från datorn, stängde av ON och tryckte en extra gång på FF. Vid överföringstillfället lyste både ON och FF, men som sagt inget kom ut. Vad är det för fel?

(Text 14) Sven Wickberg <1384>
Ärende: PC-FILE och HP-laserjet
Min kollega hade ett register över alla landstingen plus div andra adresser inalles 72 stycken. När han begärde utskrift med etikettprogrammet kom bara 12 etiketter ut! (Det problemet har jag inte kunnat lösa - det här förresten inte hemma i detta möte; men jag skrev en rapportfil i stället så vi fick ut adresserna. MEN...)
Laserjeten vägrade skriva ää. Den hoppade helt enkelt över dessa tecken: "Örebro län" blev "rebro ln".

En koll med FILTER från SPCS förvandlade de överhoppade tecknen till de vanliga parenteserna - tydligen FANNNS tecknen i basen, men tolkades inte av laserjet.
Efter en snabbtitt i dokumentationsfilen fann jag en uppgift om att man måste ladda in filen HPTRANS med argumentet DK/N. I maskinen fanns HPTRANS med varianten ROMAN, men inget DK/N. När jag skrev in DK/N och gjorde HPTRANS ON, så kom ää som de skulle! (Min kollega blev vederbörligen imponerad...)
Men jag har frågor. Måste HPTRANS alltid vara inladdad när man kör utskrift på Laserjet? Eller gäller det bara filer från PC-FILE? Skall man sätta in HPTRANS i auto-exec-filen, eller bör den finnas i någon lokal startfil, t ex den som laddar PC-FILE, och tas bort när man lämnar det programmet?

(Text 15) Paul Pries <5322>
Ärende: Två frågor om HP-laserjet
Att sidan inte skrivs ut automatiskt beror på att den inte är full, och programmet som sköter utskriften inte skickar någon avslutande formfeed.
Vad gäller ää så skall du inte behöva varesig filter eller 7H. Jag har för mig att man bara sätter charset i skrivaren till IBM 8-BIT. Kommer inte ihåg det precis exakt, men jag har för mig det var så vi gjorde det. Glöm inte att trycka på reset i mer än 5 sekunder efter att du ändrat något skrivarens konfiguration! Detta vållade oss stort bekymmer innan vi kom på det! Det står visserligen om det i manualen, men vem läser den innan man får problem??
Hoppas du lyckas!
/Paul.

(Text 16) Bo Kullmar <1789>
Ärende: Två frågor om HP-laserjet
Ja, en laser är en SIDSkrivare och skriver hela sidor och måste ha FF på slutet för att skriva ut sidan.
En HP laser har normalt Roman 8 som teckenuppsättning. Det är HP:s egen och har ää, men ej på samma ställen som IBM. Det är en 8 bitars teckenuppsättning. Man kan dock ställa in IBM ASCII eller Svensk 7-bitars (som p ABC). Roman 8 är standard så man får ställa om skrivaren om man vill ha IBM PC ASCII som standard.

(Text 18) Sven Wickberg <1384>
Ärende: Två frågor om HP-laserjet
Hur sätter man Charset i skrivaren till IBM 8-bit?
Om programmet gör utskrift när filen stängs borde det väl komma utskrift när man lämnar PC-FILE då? Jag får prova.

(Text 19) Göran Eriksson <4988>
Ärende: Två frågor om HP-laserjet
Teckenuppsättning IBM-PC som jag tror HP kallar den får du genom att skicka <Escape> (10U till skrivaren. Om du av någon anledning vill gå tillbaka till standarduppsättningen (Roman-8) så skickar du <Escape>(8U, <Escape> är förstas escape-tecknet, 27 decimalt, 1B hexadec escalt.

(Text 20) Sven Wickberg <1384>
Ärende: Två frågor om HP-laserjet
Alltså skicka Esc (10U) (vänsterparentes, nolla, stort U) för IBM-PC-teckenuppsättning och för att få tillbaka standard Roman-8 skicka esc (8U) (vänsterparentes 8 stort U).

Är det korrekt uppfattat? Man vet aldrig vad det blir för konstiga tecken här i msg.

Sedan är frågan: Hur skickar jag denna teckenföljd? Kan jag göra det från tgb? Om ej, så hur i stället?

(Text 22) Göran Eriksson <4988>

Ärende: Två frågor om HP-laserjet
Ja det ska vara escape-vänsterparentes-ettanolla-versalt U för IBM-PC och escape-vänsterparentes-åtta-versalt U för Roman-8. Glömde visst påminna dig om att om det är en någotsånär modern HP Laserjet så kan man ställa in detta via setup-menyn på själva skrivaren också. Se manualen. Det går att skicka tecknen från tangentbordet, från en batch-fil eller från ett program. Det finns PD-/SW-program som kan sända sådana här styrsekvenser, men jag minns i skrivande stund inte vad något enda av dem heter eller vad det finns något här i monitorn. Har du tillgång till BASIC så borde följande lilla snutt göra jobbet:

```
10 LPRINT CHR$(27);"(10U";
20 SYSTEM
och motsvarande för Roman-8.
```

(Text 23) Paul Pries <5322>

Ärende: Två frågor om HP-laserjet
För att sätta IBM 8-BIT bläddrar du fram detta val i skrivarens lilla textfönster, trycker på skrivarens "enter" knapp, och trycker sedan på skrivarens "reset" knapp tills texten "07 RESET" dyker upp i skrivarens textfönster.

Den version av PC-FILE vi använder skickar med en formfeed på slutet av varje utskrift, så det problemet har vi aldrig haft. Dock har samma sak spökat vid andra tillfällen. Kanske är det konfigurerbart....
/Paul.

(Text 29) Gösta Vestman <3085>

Ärende: Minnesutbyggnad
Min PC (386SX) är försedd med 4 minnesbankar för vardera 2 st SIMM-moduler (256 KB eller 1 MB). Jag har för närvarande 2 bankar fyllda med 256 KB = 1 MB. Om jag vill utöka minnet kan jag alltid fylla de återstående bankarna med 256 KB och få 2 MB. Nu till frågan.
Om jag vill utöka minnet ännu mer, kan jag då blanda 256 KB 1 MB SIMM-moduler, d v s fylla de nu tomma bankarna med 1 MB och få totalt 5 MB? Eller fylla ytterligare en med 256 KB och den sista med 1 MB och få totalt 3,5 MB?

(Text 33) Kari Kiviranta <7841>

Ärende: Minnesutbyggnad
Du kan fylla resterande platser med 256x9 och få då 2Mb med alla bankar fyllda. Skall du ha mera så måste du gå upp till 4 st 1Mb x9 för 4 MB totalt eller 8 st 1Mb x9 för 8Mb totalt. Tydigen skall man inte blanda 256Kx9 och 1Mb x9. Detta gäller till min -386SX och torde vara det vanligaste sättet. Sedan gäller att du får rätt hastighet på modulerna 100ns eller 80ns beroende vilken hastighet CPU:n kör på. Kör du 16 Mhz så använd 80ns, kör du lägre hastighet så kan du klara dig med 100 ns.

(Text 35) Lars Lindeberg <7124>

Ärende: PARK
Är det någon som vet varför man skall använda PARK prog.när man slutar.Har tittat på hur den gör när man använder den, då söker sig armen med läshuvuderna ut på skivorna den anstränger sig för att komma långs ut,men när så jag bryter strömen så åker armen in mot centrum.Det förefaller mig som att hemmaläget är vid centrum å inte vid det yttre stället som den anstränger sig att komma till,men ej ligger kvar. Har någon en förklaring till detta?? Tacksamm för tips./LL.

(Text 37) Kari Kiviranta <7841>

Ärende: PARK
Varje gång du stänger eller kopplar på din dator så går det en strömpuls till dit hårddisk som får skriv/läshuvudet att göra ett litet ryck. Detta ryck kan i VÄRSTA fall orsaka en skada på disken. Man använder PARK för att föra skriv/läshuvuden till en ofarlig area på hårddisken för att undvika ev skada på arean som innehåller information. Om dessa ofarliga områden ligger på yttre- eller inreparter, kommer jag inte ihåg, men det borde logiskt ligga vid dem inre.

(Text 39) Nils Hammar <4341>

Ärende: PARK
Det är inte det som är problemet. Han försöker använda PARK på en självparkerande disk! Strunta helt kallt i PARK på den disken vet jag.

(Text 40) Benny Löfgren <2615>

Ärende: PARK
Återigen: ALLA hårddiskar är "självparkerande", tom gamla 5MBs Seagate ST506:or hade den faciliteten! Möjligen kan "park"-program kunna tänkas ha någon funktion om man ska flytta disken.
En hårddisk är som allra känsligast under upp- och nedvarvning, då huvudet inte har byggt upp sin luftkudde att sväva på. I de lägena måste disken stå absolut stilla. Är den väl uppe i fart, alternativt helt stillastående tål den förvånansvärt kraftiga smällar (ok, relativt sett).

(Text 41) Nils Hammar <4341>

Ärende: PARK
Knappast alla diskar är självparkerande! T.ex. Miniscribes 3,5" 20MB hårddiskar är INTE det. Finns även andra diskar som inte är det, som t.ex. Seagate ST225 och ST238. Jag har studerat dem relativt ingående, och konstaterat att det förekommer ett stort antal som INTE är det. Dessa diskar härrör dock från en period i mitten av 80-talet, då det började bli vanligt med billiga hårddiskar, och då var detta en besparing. På senare tid har dock de flesta diskar självparkering medelst en fjäder eller något, som ser till att huvudena dras undan. Detta förhindrar alla hårda krascher som var vanligt ett tag, men en mjuk krasch kan man alltid få, även om risken är måttlig.
En mjuk krasch kan man dock alltid komma runt.

(Text 90) Anders Franzen <5258>

Ärende: XGA
Såg någonstans att IBM hittat på en ny grafikstandard kallad XGA. Är det någon som vet mer om detta?

(Text 91) Claes Börjesson <6928>

Ärende: XGA
IBM har nyligen haft ett antal "uppvisningar" där de visat sina senaste skapelser, bl.a. XGA. Det har någonting på 1000x1200 punkter och mer än 256 färger. Skärmen uppdateras flera gånger snabbare än en "gamla" skärm varför man kan anta att den är försedd med grafikprocessor (i allmänhet blir ju en skärmpupdatering långsammare ju högre upplösning man har). (Med uppdatering menar jag att grafiken skrivs ut snabbare inte antal delbilder per sekund).
Vid sin uppvisning visade man något "REVOLUTIONERANDE" ljud och grafik, multimedia !!!!!. Enligt IBM's egna ord är detta något helt nytt, att det har funnits både flerkanals ljud och bild på både hemdatorer och även andra (Mac) vet dem tydligen inte om. Att sedan en stor del av de "töntar" som är med på dessa visningar går på IBM's skickliga marknadsföring visar bara på hur duktiga IBM är på marknadsföring, och tyvärr att deras målgrupper (chefspersoner) fortfarande till stor del saknar vidsynhet och datormognad.
Claes Börjesson

(Text 92) Conny Westh <7433>

Ärende: XGA
Instämmer i föregående talares anförande angående chefs bristande vidsynhet och datormognad.
M.V.H. Conny Westh

(Text 93) Per Andersson <5581>

Ärende: XGA
Kan ju nämna att SUN kommit med en ny grafik värsting. 400 MIPS, grafikprocessor allan, och 24 bitars färg. Ljud klarar ju alla sparcstationer, om än kanske inte stereo? Man lär visst kunna köra video genom processorerna på den, dvs. man kan edita i bild-strömmen.

(Text 94) Mattias Ericson <6615>

Ärende: Parallellporten
Hur ser pinkonfigen ut på den 25-poliga parallellporten?
/Mattias

(Text 95) Jan-Olof Påvall <1116>

Pin	Dir	Name	Active	Function
1	Out	Strobe	Low	Indicates valid data available
2	Out	Data bit 0	High	Least significant bit of data byte
3	Out	Data bit 1	High	
4	Out	Data bit 2	High	
5	Out	Data bit 3	High	
6	Out	Data bit 4	High	
7	Out	Data bit 5	High	
8	Out	Data bit 6	High	
9	Out	Data bit 7	High	Most significant bit of data byte
10	In	Acknowledge	Low	Indicates data received and device is ready for more
11	In	Busy	High	Device cannot receive data
12	In	Paper end	High	Device is "out of paper"
13	In	Select	High	Device is "selected state"
14	Out	Auto feed	Low	Device to perform line feed after each line sent
15	In	Error	Low	Device unable to perform
16	Out	Init printer	Low	Reset device to initial state
17	Out	Select input	Low	Device can accept input
18		Ground		
19		Ground		
20		Ground		
21		Ground		
22		Ground		
23		Ground		
24		Ground		
25		Ground		

(Text 96) Peter Isoz <2164>

Ärende: Parallellporten
Enligt mina noteringar är den som följer:

```
1: Strobe,
2:bit 0, --
9:bit 7,
10: -Ack,
11:Busy,
12:PE
13:SLCT,
14:-AutofeedXT,
15:-Error,
16:-int
17:Slct in
18-25: Gnd
med OUT 892,N (i basic) kan man sätta
bitar 2-9
Lycka till
Hälsningar /Peter
```

(Text 97) Anders Franzen <5258>

Ärende: XGA
Ja, du har rätt, IBM lanserar sina "nyheter" som om det är ny teknik som introduceras fastän det egentligen bara är skäpman som andra redan gjort. Å andra sidan har IBM en förmåga att etablera standards och det dröjer inte länge förrän alla andra tillverkare apar efter. Kanske tack vare alla pundiga beslutsfattare.
Som vanligt är det bäst att ta det lugnt och avvakta. Vill man ligga längst fram i "utvecklingen" får man betala dyrt!

(Text 109) Björn Dahlberg <4428>

Ärende: Klockadress
Vad är adressen till klockan i en IBM-kompatibel AT eller 386SX. Jag skulle vilja läsa både datum och tid för att använda det i ett program jag håller på med

En fråga till: Var kan man lära sig om minnesresidenta program? Lämplig litteratur?
Hälsn Björn

(Text 110) Anders Franzen <5258>

Ärende: Klockadress
På adress 40:46C finns klockan som fyra bytes. Denna räknare räknas upp med ett varje klocktick, dvs ca 18 ggr/sekund. Läser du detta klocktick (antalet tick sedan midnatt) måste du själv omvandla det till timmar, minuter osv.
Genom att anropa BIOS kan man också få tiden. På AT och högre finns support för att hämta och ställa hårdvaruklocka. Dessa funktioner anropas bäst från assembler. Man kan fråga DOS vad klockan är. Då får man direkt reda på datum eller klockslag och slipper omvandla. Men denna frågan görs också bäst i assembler. Man kan använda funktioner i något högnivåspråk, t ex time(..) i C. Det är det enklaste sättet!

(Text 113) Mikael Lindroos <7410>

Ärende: Klockadress
Hmmm...du menade förstast 0000:046C eller 0040:006C eller hur...:-)

(Text 114) Benny Löfgren <2615>

Ärende: Klockadress
Inledande nollor brukar normalt kunna ignoreras utan problem, undantag är t ex c om använder '0' respektive '0x' för att beteckna oktala och hexadecimala tal.

(Text 115) Bo Michaelsson <913>

Ärende: Klockadress
Man kan också, om man använder C, hämta programmet TID.ZIP. Det innehåller källkod för att hämta tiden.

(Text 116) Björn Dahlberg <4428>

Ärende: Klockadress
TID.ZIP verkar vara ok men jag skulle vilja ha det som ett minnesresident program som känner av klockan och ex vis varje timme skriver ut en textfil med datum och en textfil med tiden. Kan någon hjälpa mig?
Hälsn Björn

(Text 117) Anders Franzen <5258>

Ärende: Klockadress
Ja, du har rätt, jag skrev fel.

(Text 118) Anders Franzen <5258>

Ärende: Klockadress
Fihandtering i minnesresidenta rutiner kräver avancerad assemblerprogrammering pga sättet som MSDOS är uppbyggt på.

Pascal

(Text 9) Morgan Lantz <4359>

Ärende: Turbo Pascal
Jo jag har nu testat att skriva ett prgram som öppnar ett fönster i turbo pascal. Det går bra i vanliga fall. Fast nu har jag tänkt göra detta i grafik läge. När jag öppnar ett fönster i grafik läge så händer det inte något alls!
Jag undrar om det går att öppna fönster i grafikläge,den enhet som jag använder mig av är tpw73! Om det inte går att öppna fönster i grafik läge med den enheten. Så undrar jag om det finns någon annan fönsterhanterare till turbo pascal som klarar av detta som jag vill göra! Att öppna fönster och dylikt i grafikläge!
Sedan så undrar jag hur man bär sig åt för att flutta ett fönster i sidled, som man kan göra i turbo pascal med filarna. Om man nu har fler fönster än ett och vill flytta sig till nästa fönster .Precis som i ed! Som jag nu skriver med!
En sak till, hur bär man sig åt för att kunna få ut innehållet i ett bibliotek i bredd på skärmen ett fönster. Som man får i turbo pascal och turbo c!
Jo jag håller på med ett program som ritar ut matematiska funktioner på bildskärmen. Detta går ju snabbt och så. Fast hur man skall bära sig åt för att kunna mata in en ny funktion vet jag ännu inte! Detta skall man kunna göra när man kör programmet. Det är ju enkelt att lägga in en funktion i själv programmet. Fast då blir man ju låst vid den funktionen. Det skall vara lätt att ange ny funktion när man kör programmet!
Jaha det vara väl allt för denna gång. Fast jag kommer nog att ställa fler frågor kring turbopascal. Eftersom jag mest programmerar i detta språk! Fast jag tycker ju att klubben kan sända ut en modula 2. Vi som nu kör pc!
Jaha hoppas någon av er nu kan svar på dessa små frågor. Tacksam för svar!
Med vänlig hälsning (4359) Morgan Lantz.

(Text 10) Jan-Olof Svensson <6057>

Ärende: Matematiska funktioner
Du vill alltså göra ett program som fungerar ungefär som Eureka The Solver. Jag antar att du får läsa in funktionen, t.ex. $y = \sin x * e^{-0.3x}$, till en strängvariabel som programmet sedan får analysera så att motsvarande fördefinierade funktioner kan anropas med lämpliga argument.

Om du har en systemskiva där det lilla kalkylprogrammet MC (MicroCalc) ingår så kan du kanske få ut någonting av den källkoden.

Mvh Jan-Olof Svensson

(Text 11) Jan Sundström <7804>

Ärende: Att rita funktioner?

Ätminstone i tidigare versioner av Turbo Pascal, så fick man snabbt till ett enkelt kalkylprogram (Micro Calc eller något sådant). Om man tittar där, så kan man se hur man gör en parser för funktioner som matas in i form av en sträng. Det är inte på nybörjarnivå att själv skriva en sådan, men det är inte så svårt att direkt planka tillämpliga kodavsnitt.

En ide vore kanske att ta det hela ett steg vidare. Att strängen skall tolkas om och om igen för varje nytt funktionsvärde, gör ju inte det hela så snabbt precis. Men med hjälp av adresserna till olika matematiska funktioner (i Turbo Pascal 5.5 kan man ju ha pekare till funktioner) kunde man kanske ordna till en liten kompilator som kompillerar en inmatad sträng vid run-time, lägger koden i en sträng och som man sedan kan anropa via en pekare?

(Text 12) Benny Löfgren <2615>

Ärende: Att rita funktioner?

Man kanske inte behöver gå så långt som att göra självmodifierande maskin-kod med anrop till pascals runtime-moduler... Man kan t ex göra ett slags pseudokod (p-kod, de som kórt Basic II vet vad jag menar) som man kan tolka via en enkel dispatcher. Man får då lägga upp p-koder för de stack-operationer man behöver (pusha/poppa flyttals-element) och en p-kod för varje funktion man behöver.

En addition av två tal skulle då kunna se ut så här efter parsing: push arg1, push arg2, call add (som poppar två argument från stacken, adderar och pushar resultatet), pop resultat.

(En dispatcher är en rutin som hoppar till en av flera funktioner utgående från ett värde (p-kod i detta fall) som slås upp i en tabell. Enklaste exemplet torde vara basics ON x GOSUB a,b,c som hoppar till rad a, b eller c beroende på om x är 1, 2 eller 3.)

(Text 13) Morgan Lantz <4359>

Ärende: Felhantering i turbo pascal 5.5!

Jo jag vet ju att man kan ta hand om fel i turbo pascal med ioResult. Jag bara undrar hur jag gör ex när jag vill ta hand om följande fel. När jag gör en procedur i tp som skall se till att man kan byta bibliotek så blir det ju fel om man angär ett bibliotek som inte finns! Hur skall jag ta hand om detta fel i tp så att man inte hamnar i editorn igen. Jag vill att det skall skrivas ut ett felmeddelande på skärmen. Ex biblioteket finns inte! Det var ett tag sedan jag använde mig av ioResult. Så jag minns inte riktigt hur man använder ioResult.

I för sig så kan man kanske även använda sig av de doserror som finns nmär man använder engeten dos! Någon som vet något om detta så är jag tacksam för svar! Mvh Morgan Lantz.

(Text 14) Peder Gierzt <6133>

Ärende: Hayes-kommandon

Kan någon hjälpa mig med följande: Jag skulle vilja skriva en liten program-snutt som ger följande kommando till mitt modem:

Sänd ATSO=0(+ENTER) till COM2:

Hur gör jag det? På förhand tack!

(Text 15) Lars Gjöring <6825>

Ärende: Felhantering i turbo pascal 5.5!

```
begin
  (*I-*)
  ChDir(biblnamn);
  if ioResult < 0 then writeln('Kan inte hitta biblioteket!');
  (*I-*)
end.
```

I ovanstående skall (* och *) motsvara måsvingar. (*I-*) gör att programmet inte stannar med felmeddelande. (*I-*) sätter på denna egenskap igen.

(Text 16) Lars Gjöring <6825>

Ärende: Hayes-kommandon

Jag har skickat in filen asyn4u.zip, som innehåller en tpu-fil som ger tillgång till alla procedurer/funktioner, som behövs för kommunikation via com1 och com2. Källkod till demoprogram medföljer. Efter att ha skrivit uses asyn4u är det väldigt enkelt att skicka vad som helst till com2.

```
Program SkrivarTest;
uses Printer;
BEGIN
  writeln(Lst,'Det här skriver jag ut på printern.');
```

END.

(Text 21) Björn Dahlberg <4428>

Ärende: Start batch

Jag önskar starta ett annat program från ett pascal-program. Användningen skulle kunna vara ett menyprogram med vars hjälp man kör igång div andra program och kommer sedan tillbaka till huvudprogrammet Hälsn Björn

(Text 23) Lars Gjöring <6825>

Ärende: Start batch

Läs Reference Guide sid 251-252, Exec procedure. Se upp med det där att reducerar maximum heap size, annars, kom- mer det inte att finnas tillräckligt utrymme för det program som skall köras.

(Text 28) Lars-Börje Cid <7390>

Ärende: The Turbo Engine (TTE)

Skall inom kort lägga in en demoversion av TTE jämte exempel på vad TTE åstadkommer.

TTE är en programkodsgenerator för Turbo Pascal. I TTE beskriver du dina menyer, foormulär, register m.m.

Allt länkas med varandra och resulterar i en källkodsfil som är färdig att kompilera. Programmen blir mycket proffsiga, och framförallt dialoghanteringen blir väldigt trevlig.

Eller vad sägs om att följande beskrivs för varje enskilt inmatningsfält:

Ledtext, Längd, Typ, Decimal, Får lämnas tomt, spara innehållet tömning, Versaler, Avbrytstangenter, Normala sluttangenter, Vad skall hända innan inmatningen börjar, vad skall hända när fältet blivit inmatat. Du kan flytta fälten fritt i den dialog du skapar. Du kan förstora, förminska samt flytta dialogfönstren hur du vill. Du beskriver dina indexregister, och använder dem med enkla funktions-anrop: SF find-match, SF write ..etc... TTE ser till att registren hanteras korrekt med avseende på nycklar.

Mer information kommer...

(Text 29) Morgan Lantz <4359>

Ärende: The Turbo Engine (TTE)

Mär lägger du in den där demo varianten av TTE då??

Mvh Morgan Lantz.

(Text 30) Lars-Börje Cid <7390>

Ärende: The Turbo Engine (TTE)

Så snart som möjligt.

Jag skulle ha gjort det tidigare, men pga en bugg i TTE är jag tvungen att vänta.

Men lite mer info kan Ni få: TTE Handhar Registerhantering, Formulärhantering, Menyhantering. Med TTE följer en mängd UNITar med många många nyttiga funktioner.

Ex på form.

Du får upp ett fönster, detta fönstret skall inrymma din dialog. Du kan alltså flytta, förstora, förminska fönstret m.m.

Du beskriver text som skall finnas i fönstret samt inmatnings- fält.

Dialogen anropas sedan med:

SF Form (Formno); Formno är då numret på dialogen.

I Proceduren SF_Form sker ALLT som hör till dialogen, även anrop till för-/efterbearbetning av inmatningsfält etc.

Skall skynda på med detta...

(Text 31) Arne Lager <5798>

Ärende: Turbo pascal v. 5.5

Finns det någon som kan ge mig ett litet handtag när det gäller att få utskrift på skrivare i Turbo pascal?

MVH ARNE

(Text 33) Lars Gjöring <6825>

Ärende: Turbo pascal v. 5.5

Du kan använda uniten Printer.tpu! Om du i ditt program skriver 'uses Printer' så kommer ett 'filnamn' Lst att deklaras, bli 'assignad' till LPT1 och öppnad. Du kan då använda write(Lst,'.....') och writeln(Lst, '.....')! Exempel:

(Text 34) Björn Dahlberg <4428>

Ärende: Date Time

Är det någon som har en programsnutt som gör följande: Hämtar datorns satta datum och lagrar det i form av en textfil på formen åå-mm-dd (ex. 90-11-07) Det-samma för tiden (ex. 15.35.00) Helst i turbo-pascal Hälsn Björn

(Text 55) Claes Tullbrink <6083>

Ärende: Date Time - quick'n'dirty

Hej! Tills någon skickar in en programsnutt kan du ju eventuellt använda doss- kommandona date och time istället för pascal. Skapa en fil "cr" som innehåller bara ett enter (13). Skapa en batfil som ser ut så här:

```
date < cr. > datum.fil
```

```
time < cr. > tid.fil
```

Vill du ha både datum o tid på samma fil - ange samma filnamn på rad 1 och 2 och byt ut > mot >>

>> bara på rad 2: du sparar bara den sista datum- och tids-uppgiften.

>> även på rad 1: Filen byggs på för varje gång du kör bat-filen.

Det smutsiga: Du får in även frågan om nytt datum - tid på filen.

Jag har nog en pascalsnutt på jobbet, som gör det du vill, jag kan ta hem den imorgon.

Hälsningar Claes

(Text 36) Lars Gjöring <6825>

Ärende: Date Time

program DateTime;

uses dos;

var

file1, file2 : text;

Y,M,D,W : word;

H,M,S,S100 : word;

procedure OpenFiles;

begin

assign(file1,'DATE.TXT');

rewrite(file1);

assign(file2,'TIME.TXT');

rewrite(file2);

end;

procedure SaveDate;

begin

GetDate(Y,M,D,W);

write(file1,Y,'-');

if M<10 then write(file1,'0');

write(file1,M,'-');

if D<10 then write(file1,'0');

writeln(file1,D)

end;

procedure SaveTime;

begin

GetTime(H,M,S,S100);

if H<10 then write(file2,'0');

write(file2,H,':');

if M<10 then write(file2,'0');

write(file2,M,':');

if S<10 then write(file2,'0');

writeln(file2,S)

end;

BEGIN

OpenFiles;

SaveDate;

SaveTime;

Close(file1); Close(file2)

END.

(Text 57) Björn Dahlberg <4428>

Ärende: Date Time - quick'n'dirty

Det går inte med date och time för då kommer det med en massa onödiga ord. Jag ska nämligen kunna sätta just date och time med de skapade filerna. Pascal-snuttarna har jag f.ö fixat själv. Nu har jag en ny utnämning: Ett minnesresident program som varja ny dag kl.00.00 skriver en textfil på hårddisken med det nya datumet. Detsamma för tiden men oftare, så varje minut skriver textfil med aktuell tid. Är det någon som kan hjälpa mig?????

Hälsn Björn

(Text 58) Björn Dahlberg <4428>

Ärende: Date Time

Tack, det fungerar ungefär som jag tänkt mig. /Björn

(Text 59) Anders Dohrman <6401>

Ärende: Fylläsning i TP

Är det någon som vet hur man fixar fylläsning från Turbo Pascal. Det enda jag hittar som har med det att göra är en global variabel som heter FileMode. Den verkar inte så väldigt användbar, eller ?

(Text 65) Mikael Lindroos <7410>

Ärende: Fylläsning i TP

Du kan använda Dos-funktionerna Lock (AX=\$5C00) och Unlock (AX=\$5C01) för att låsa respektive låsa upp vald area i din fil. Observera dock att du först måste definiera filen som "Shareable" i ditt nätverk. I Novell=> FLAG FIL.SUF Shareable. För användning se nedan:

Om funktionen returnerar värdet 0 har läsningen gått bra, annars har ett fel uppstått, t.ex. har någon annan redan låst denna area. Unlock ser likadan ut förutom att Ax skall vara \$5C01 i stället. OBS! VIK-TIGT!!! När du skrivit färdigt MASTE du se till att låsa upp arean igen innan du stänger filen, annars lär du aldrig mera komma åt denna area!!!

(Text 67) Anders Dohrman <6401>

Ärende: Fylläsning i TP

Tack för det tipset. Ska prova.

(Text 68) Anders Franzen <5258>

Ärende: Fylläsning i TP

I samband med fylläsning kan ett hett tips vara att om man läser flera filer i sitt program att alltid låsa dessa filer i samma ordning! Exempel: Om man använder de två filerna A och B bör man när man skall låsa dessa alltid börja med t ex A och sedan låsa B. Vid uppläsning bör man sedan låsa upp i motsatt ordning.

Om någon vill låsa i en annan ordning, t ex först B och sedan A kan man få problem. Antag att denna någon vill låsa B precis när du låst A. Ditt program försöker sedan låsa B men den är ju redan låst. Det andra programmet försöker låsa A som du redan låst. Resultatet blir att de båda programmen läser varandra!

Man måste tänka till ordentligt i en fler-användarmiljö!

(Text 69) Anders Franzen <5258>

Ärende: Fylläsning i TP

Jag har aldrig förstätt det där med att låsa areor av en fil i MSDOS. Om man som du säger inte läser upp en area efter sig men att det inte beror på slarvigt

program utan annan händelse som får programmet att avslutas onormalt, hur gör man då?

MSDOS tycks överlämna "recovery" helt till programmeraren!

(Text 71) Anders Dohrman <6401>

Ärende: Fylläsning i MS-DOS / nätverk

Det låter livsfarligt det här ! Kan Windows sköta om det här med fylläsning åt en om man sätter upp flera dos-partitioner och kör samma program i? Hur bra fungerar det i såfall ?

(Text 72) Bo Engborg <2369>

Ärende: Fylläsning i MS-DOS / nätverk

Det finns ett verktygspaket "B-Tree Filer" multi user version with network support, för TP, av Turbo Power Software.. ca 1000:-

Function Lock(var F; FilePosition,FileLength: longint): word;

Var

Reg: Registers;

begin

if FileRec(F).Mode = fmClosed then begin

Lock := -1;

Exit;

end;

with Reg do begin

Ax := \$5C00; hDOS call 5Chd

Bx := FileRec(F).Handle;

Cx := DoubleWord(FilePosition).HiWord;

Dx := DoubleWord(FilePosition).LoWord;

S1 := DoubleWord(FileLength).HiWord;

D1 := DoubleWord(FileLength).LoWord;

hDOS(Reg);

if Odd(Flags) then

Lock := Ax

else

Lock := 0;

end;

end;

error if carry flag set!

Några ord om Editor

Radannonser

Köpes

Flexskiveenhet för ABC80, format D.

<7950>

Ingela Andersson
0171-822 42

Köpes

1. ABC806 + ABC812 färgskärm
2. MANUALER (original/kopia) till ABC800-programmen:
"MULTIBAS - v. 1.2" och
"REG800 - v. 1.1"
Har originalprogram utan dokumentation!
3. PROGRAM till ABC800M + ABC830
"DIAGRAM 800",
"BASREGISTER 800",
andra nytttoprogram som bokföring etc.
4. SCHACK-program till ABC 802 + 834-drive (om sådant existerar?).

<7759>

Mikael Wiesel
Middagsgatan 59
582 68 Linköping
013-16 07 91

Säljes

ABC800-prylar:
2 st ABC806 CPU-lådor
1 st ABC812 skärm
1 st Tranfor Datastore 20 MB+640 kB+exp

1 st Tranfor Datastream 20 (streamer)
1 st Tranfor Datadisc 56
1 st Tranfor Datadisc 88 (2x8"+exp)

Priside: 3 000:-

Övrigt

DS90/ABC9000 8" floppy, streamer, HD
4 st Tidata terminaler
1 st Facit Twist terminal (vridbar)

Priside: 12 000:-

Vid snabb affär kan detta diskuteras

Björn Lantz
08-771 60 36

Säljes

ABC806
ABC812 skärm
ABC77 tangentboard
ABC830 flexskiveenhet
Itoh CX4800 plotter

Förfrågningar till
Lennart Pettersson,
08-734 29 58

Skriftligt anbud ställs till
Skytteholmsgymnasiet
Ankdamskatan 14-16
171 43 Solna

I ABCbladet 2. 1990 och där MSG: PC Prog inlägg 6260 skriver Sven Wickberg om en ny EDITOR från SPCS Växjö.

I inläggen efter skrev de olika inläggsförfattarna om "sina" editorer och inte något mer om EDITOR kom fram.

Jag har genom jobbet (1989) kommit i kontakt med EDITORs föregångare ED eller PTED som den heter i programpaketet från CADPOINT Borås (1987, fast det provade jag inte då).

PTED är utvecklat av SPCS men ej marknadsfört som särskilt program.

ED som jag råkade komma i kontakt med kunde hantera radlängd betydligt större än som var möjligt med Q&A eller WP. Jag provade på .EXE-filer och ACADs .OVL-filer och de gick in utan protest, dock märkte jag det gick trögt att stega när raderna var längre än 65000 tecken, vad det berodde på visste jag inte då.

Jag använde ED till inskrivning text för ett register till en bok och fick fyra filer på tillsammans 555000 tecken när jag lade ihop dem. Av en händelse skrev jag in filnamnet för den ihopkopierade filen istället för en delfil jag skulle ha in i ED för bearbetning. Detta märkte jag först när jag gick till slutet och fann annat än det jag förväntade mig.

Jag gjorde senare en del experiment och fann att när maskinen startades ren utan annat än DOS 3.20, files 20, buffers 20, keybsw, så fanns = 589000 bytes enligt CHKDSK rapport att jag kunde få in 565000 tecken. ED.COM är ca 17000 bytes stor

Om filen var för stor så avslutades programmet direkt med meddelandet "Filen för stor"

Så kom då beskedet i MSG om EDITOR som jag redigerade för bladet med all annan MSG. Jag ringde DUSTIN och frågade, för deras sommarkatalog kom just då. Likaså ringde jag SPCS om brochyren, men väntade inte på den utan beställde från Dustin - när sedan brochyren kom hade SPCS ett introduktionspris på 495:- inkl moms medan Dustin hade det rekommenderade priset 595:- plus moms. Men det var ju firman som betalade.

Hur är då EDITOR eller ED.COM som den heter på disketten.

ED kommer med en hjälpfil ED.HLP och en makrofil ED.MAC.

Startar man som vanligt med en .BAT-fil anropas ED i sitt eget bibliotek men man står kvar i anropsbiblioteket.

ED tar då 77760 bytes från minnet. Startar man utan ED.HLP tar det upp 64704 bytes eller lika mycket mindre som ED.HLP är. Detta är nu inget problem jämfört med gamla ED, ty nu arbetar programmet mot disk om det behövs och hjälpskrmar finns "i programmet" hela tiden. När man kör som vanligt och använder Hjälp=F1 finns 14 skärmar tillgängliga.

Som en jämförelse har jag hämtat in 555K filens senaste upplaga om 756K som två fönster, sedan givit kommandot besöka dos och tittat hur stor .TMP-filen är (1.5MB) samt kört CHKDSK för minneskontroll och sedan gått tillbaka in i ED igen. Inga problem.

Problem vid tabeller och dess hantering med sortering - bearbetning i BASIC kan övervinnas i nya ED, man ställer vid installationen tabulering så att mellanslag skrivs i stället för tabuleringstecken och om mellanslag skall skjutas in i text eller stega över i INSERT-mod.

ED är även skärmorienterad lika som gamla Word Star, så man är kvar i kolumnen fast man stegar sidor upp och ned, inte som WP där man följer radslutet. Dock kan man inte göra blockkopiering i kolumner som WS eller WP.

ED arbetar med alla typer av filer - som det ser ut på skärmen, skräp i skräp ut. Börjar man en egen text skrivs den ned på disk som ren ASCII.

En artikel om EDITOR finns i Computer Sweden nr 22, 7/9 90 där dock författaren inte vetat om gamla ED.

För mig som arbetat med TV-editorn och senast och pågående med VDO-editorn i ABC-80 har jag med gamla och nya ED ett passande verktyg för PC-körning.

<1182> Claes Schibler

Här nedan följer skärmdumpar på de inbygda hjälpskrmarerna i EDITOR

Fil=F2 Block=F5 Sök=F6 Gå Radera=F7 Övrigt Utskrift=F8 Hjälp=F1

Filnamn:

C:\BATFIL

Tryck på RETURN så får man upp aktuellt bibliotek

På Tillägg
Filnamn: F8=Filter | F9=Sort | F10=**

C:\SPCSEDIT**

U:	K:
Z:	R:
N:	V:
P:	W:
G:	I:
T:	Q:
E:	..
O:	INFO.TXT
S:	ED.HLP
H:	ED.MAC
J:	INFO.BAT
C:	
M:	
F:	
A:	
Y:	
L:	
D:	
X:	

<KAT>

Utskrift=F8

Hela texten Shift F8
märkt Block Alt F8
Avbryt utskriften

Hjälp=F1

1 skärm	Introduktion
3 skärmar	Tangenter
1 skärm	mUs
6 skärmar	Kommandon
2 skärmar	Makro
1 skärm	Parametrar

14 skärmar	

Gå Radera=F7 Övrigt Uts

ALARM

Hjälp ej tillgänglig.

Esc = Kvittens

Fil=F2

Ny	Alt N
Infoga	
Byt fönster	Alt X
Dos	Alt F9
Spara	Alt S
avsluta Utan spara	Alt Q
spara+Avsluta	Alt E

Gå

textens Början	Ctrl PgUp
textens Slut	Ctrl PgDn
Föregående sida	PgUp
Nästa sida	PgDn
radens början	Home
radens slut	End
fönstrets Vänsterkant	Ctrl Home
fönstrets Högerkant	Ctrl End
ord vänster	Ctrl ←
ord höger	Ctrl →
Radnummer	Alt O
Märkt block	Alt B

Block=F5

Början	Alt F5
Slut	Alt F6
Flytta	Alt M
Kopiera	Alt C
kopiera Andra fönstret	Alt W
Justera	Alt T
Radera	Alt D
O-markera	Alt U
konvertera till Versaler	
konvertera till Gemener	
konvertera till Invers	
föregående Tabulator	
Nästa tabulator	

Radera=F7

Block	Alt D
Raden	Alt K
Vänster om markör	Alt J
Höger inkl. markör	Alt L
Återkalla	Alt I
Ord	Alt 6
Föregående ord	Alt 5
Tabulator	Ctrl BS

Sök=F6

Sök	Alt F
sök och Byt	Alt R
Föregående (gemener=versaler)	F3
Nästa (gemener=versaler)	F4
föregående (gemener<>versaler)	Shift F3
nästa (gemener<>versaler)	Shift F4
Motsvarande (<{[]}>) / ord	Alt G

Övrigt

rad-Automatik till	Alt A
rad-automatik Från	Alt A
Högermarginal	Alt H
Tabulator-avstånd	Alt Y
tabulatorLinjal på/av	
lägg in Styrtecken	Alt Z
konvertera till Versaler	Alt 1
konvertera till Mixat	Alt 2
konvertera till Gemener	Alt 3
konvertera till Invers	Alt 4
skriv Datum	Alt 7
visa teckenKod	Alt 8
Editorns kommandorad	F9
iNdragnin på/av	
Om editor	Alt F1

LÄSÖVNING I FINASTE KANSLISVENSKA

"Ingen uppmärksam läsare av ABC-Bladet lär sväva i okunnighet om att ett icke obetydligt antal av de alster som fyllt Bladet under de gånga året har burit signum <1384>. Att detta gillats av redaktören torde kunna betraktas som en truism, men har detsamma gällt läsekretsen? För visso har skribenten genom åren haft kontakt med ett litet, dock icke försumbart, urval av denna som uttryckt sitt gillande. Det ogillande som kommit inläggen till del har merendels härrört från betydligt mer erfarna programkonstruktörer som haft åtskilligt att säga om amatörmässigheten i författarens programkod. Men detta har merendels givit sig till känna i form av uppföljande kritisk granskning med presentation av alternativa metoder, något som för läsekretsen måste anses mera positivt än negativt."

Huu, vad är det för språk! Skall vi släppa in den gräsligaste kanslisvenska i Bladet? Varför inte i stället säga som det är:

Sven Wickberg skriver massor i Bladet. Det är bra för redaktören, men är det också bra för läsekretsen? Jovars, åskilliga har genom åren uttryckt sitt gillande. De som inte gjort det har vanligen tillhört DSG och tyckt att programexemplen varit dåligt kodade. Men det har ju lett till att de visat bättre kodning, och det kan väl alla vara glada för.

Som sagt var.

Kom ihåg att anmäla adressändring i tid

<Annon>

Datahjälp i Målilla AB

Datahjälp i Målilla AB startade 1981 och sysslar endast med ABC-utrustning.

Vi levererar idag till försvaret, skolor, kommuner och företag. Vi utför också service på ABC-utrustning.

Vi har för en tid sedan övertagit en stor del ABC-produkter från Motala, Linköping och Stockholm och säljer dessa till bra priser.

För Er som vill komplettera Er utrustning eller få den lagad är Ni välkommen att ringa till oss så skickar vi en prislista

Specialerbjudande för ABC-klubbsmedlemmar

CPU-kort till ABC80 inkl kretsar	200:-
Kåpor till ABC80	30:-
16 kB externt RAM till ABC80	50:-
Numeriskt tangentbord till ABC80	100:-
512 kB internt till ABC806	
inkl videokort	900:-
Epson LX-400 (även MS-DOS-kompatibel)	2560:-
Serieinterface 8145 (2 kB) till Epson skrivare	395:-
Disketter 5.25" 96 tpi 10 pack	45:-

OBS! Alla ovanstående produkter är nya och med garanti

Övrigt:

Cat-net centraler från 1200:-

tillbehör till Epson PX-4, PX-8 och HX-20

IBM-prommar till Epson skrivare

Litteratur, diskettboxar, papper och färgband

Plus mycket annat

Alla priser är inkl moms men frakt tillkommer

Postadress

Box 64

570 82 Målilla

Besöksadress

Stockholmsvägen 72

570 82 Målilla

Telefon

0495-213 35

Telefax

0495-213 46