

ANVÄNDARHANDLEDNING

ABC1600 FÖNSTERHANTERARE

FÖRORD

ABC1600 FÖNSTERHANTERARE är ett hjälpmedel för presentation på bildskärmen. Fönsterhanteraren, som utnyttjar grafiken hos ABC1600, är speciellt användbar är då den tas till hjälp för att konstruera egna meny fönster. Ett exempel på en sådan meny är den som visas då man startar kommandotolken (WSH).

Fönsterhanteraren kan användas tillsammans med valfri UNIX System III- eller UNIX System V-kompatibel programvara och klarar att arbeta med 16 olika fönster samtidigt.

Vi hoppas att denna användarhandledning skall hjälpa Dig att komma igång med att utnyttja grafiken och konstruera egna menyfönster så snabbt och smidigt som möjligt.

Fönsterhanteraren är utvecklad av Luxor Datorer AB i Motala.

Motala i november 1985
LUXOR DATORER AB

INNEHÅLL

| | |
|--|----|
| INNEHÅLL..... | 2 |
| INNEHÅLL AVSNITT 3.4, 3.5 | 3 |
| FÖRORD..... | 4 |
| | |
| 1 INLEDNING | |
| 1.1 Att använda handledningen..... | 5 |
| 1.2 Syntax..... | 5 |
| 1.3 Presentation..... | 7 |
| | |
| 2 PRESENTATION PÅ ABC1600's BILDSKÄRM | |
| 2.1 Teckenmarkören..... | 10 |
| 2.2 Grafikmarkören..... | 10 |
| 2.3 Det virtuella rummet..... | 12 |
| | |
| 3 FÖRPROCESSORN - WSHP | |
| 3.1 Introduktion till WSHP..... | 14 |
| 3.2 Syntax..... | 14 |
| 3.3 Infilens uppbyggnad: Dataposterna, Aktionposterna...15 | |
| 3.4 * Dataposter..... | 17 |
| 3.5 * Aktionposter..... | 41 |
| 3.6 Skrivning av enstaka strukturer på fil..... | 48 |
| 3.7 EXEMPEL 1, 2 | |
| | |
| 4 KOMMANDOTOLKEN - WSH | |
| 4.1 Introduktion till WSH..... | 55 |
| 4.2 Syntax och igångsättning | |
| 4.3 Några kommentarer till användning av WSH..... | 56 |
| | |
| 5 ORDFÖRKLARINGAR..... | 59 |
| | |
| 6 REFERENSLISTA - nyckelord till WSHP..... | 62 |
| | |
| 7 SAKREGISTER..... | 67 |

*-markerade avsnitt har utförlig innehållförteckning på följande sidor.

INNEHÅLL AVSNITT 3.4

| | | |
|--------|-----------------------------|----|
| 3.4.1 | Fönster..... | 19 |
| 3.4.2 | Ikoner..... | 26 |
| 3.4.3 | Strängar..... | 28 |
| 3.4.4 | Musmarkör..... | 29 |
| 3.4.5 | Rubrik..... | 32 |
| 3.4.6 | Bibliotek..... | 33 |
| 3.4.7 | Environ, Penviron, Lenviron | |
| 3.4.8 | Kommandon..... | 34 |
| 3.4.9 | Flaggor..... | 35 |
| 3.4.10 | Zoomlista..... | 36 |
| 3.4.11 | Ersätt en mus..... | 37 |
| 3.4.12 | Bakgrund..... | 39 |

INNEHÅLL AVSNITT 3.5

| | | |
|-------|---------------|----|
| 3.5.1 | Init..... | 41 |
| 3.5.2 | Meny..... | 42 |
| 3.5.3 | Val..... | 44 |
| 3.5.4 | Aktioner | |
| 3.5.5 | Terminal..... | 46 |

1 INLEDNING

1.1 ATT ANVÄNDA HANDLEDNINGEN

Denna handledning består av två huvuddelar.

Kapitel 3 beskriver infilen, som bestämmer menyfönstrets utseende. Förprocessorn (WSHPP) omvandlar infilen till binärformat - en infil till *kommandotolken (NSH)*.

Kapitel 4 beskriver start av kommandotolken. Då den startas sker själva uppritningen på skärmen.

Om man snabbt vill få en inblick i hur fönsterhanteraren fungerar i praktiken kan man, efter att ha läst avsnitt 1.3 PRESENTATION, börja med att studera exemplen som finns i avsnitt 3.7.

I kapitel 4 finns ett mindre exempel i direkt anslutning till de flesta funktionbeskrivningarna.

Handledningen kan också användas som uppslagsverk. Sakregistret innehåller de viktigaste uttrycken och även några fackuttryck - både på engelska och svenska. Några grundläggande begrepp finns förklarade i ett separat kapitel - 5 ORDFÖRKLARINGAR. De ord som finns upptagna i den listan är markerade med *kursiv fetstil* både i sakregistret och första gången de nämns i texten.

1.2 SYNTAX

Följande enkla regler har använts för framställningen i denna handledning.

- Ord och begrepp som skall anges i ett visst sammanhang symboliseras med klamrar runt det aktuella <ordet> som skall specificeras.

- För ett urval översatta engelska fackuttryck visas, första gången det förekommer i texten, även det engelska uttrycket genom (eng: 'ordet'). Ett exempel på detta är ordet skärm som i texten utmärks med: skärm (eng: 'screen').

- Citerade tecken och egennamn anges inom denna typ av 'citattecken'.

- Ett urval av de tecken, till exempel '&', ';' och ':', som citeras förtydligas inom parentes efter tecknet. Ett exempel: ':' (kolon).

- Indrag görs i beteckningar för att understryka nivåförhållanden (i strukturer - nyckelorden presenteras under strukturnamnet, i programexempel - ju lägre nivå på programrutinen desto större indrag för programkoden) och för ESC-sekvenser.

Exempel på detta är strukturen med dess nyckelord:

```
strukturen heter lås
    första nyckelordet
    andra nyckelordet
    sista nyckelordet
```

och

```
for i:=1 to 3 do
    begin strukturen lås
        while l>räknaren do
            begin...

        end
    end
next i
end
```

samt sekvensen

```
ESC N y c CTRL-k ESC el
```

- Kontrolltecken enligt *ASCII* anges med den standardiserade förkortningen. Ett exempel är tecknet 'Carriage Return', med decimalt värde tretton, som anges med 'CR'. Vid indrag anges dock citerade tecken utan citatmarkering.

- Vissa ASCII-tecken förtydligas genom en decimal värdeangivelse inom parentes:

CR (decimalt 13)

- Tecken som förutsätter samtidig nedtryckning av flera tangenter anges med '-' (bindestreck) emellan:

CTRL-L

1.3 PRESENTATION

Då resultat av beräkningar som gjorts med hjälp av dator skall presenteras är det ofta fördelaktigt att göra presentationen i grafisk form - dvs med hjälp av diagram, figurer och symboler. Det är då man har nytta av ABC1600 FÖNSTERHANTERARE. Med dess hjälp bygger man enkelt upp egna menyfönster, exempelvis för att sammankoppla olika program.

Fönsterhanteraren i ABC1600 förutsätter inte att applikationsprogramvaran är förberedd för fönsterhantering. ABC1600-systemet svarar självt för alla åtgärder som krävs för samkörning med godtycklig UNIX System III- eller UNIX System V-programvara. Detta betyder att de program som används inte alls behöver uppdateras för användning tillsammans med fönsterhanteraren.

Då man startar sitt ABC1600-system startar fönsterhanteraren automatiskt. Flera program kan användas parallellt eftersom flera fönster, med information från de olika programmen, kan

visas samtidigt på bildskärmen. Ett och samma program kan även utnyttja flera olika fönster samtidigt. Fönstren emulerar terminalen DEC VT100. Till denna standard har sedan ytterligare funktioner adderats.

Ändringar, förflyttningar och andra fönsteroperationer sker med en mus-styrd *markör*. Om någon mus ej finns tillgänglig används istället tangentbordets *markörplacerare* eller funktions-tangenterna PF13, PF14 och PF15. (Markörplaceraren är den stora vita fyrkanten på tangentbordet).

Exempel på speciella fönsteroperationer är:

- Bläddring framåt eller bakåt mellan olika fönster.
- *Zoomning* av tecken.
- Kopiering mellan olika fönster.

Användning av *ikoner*, *rullgardiner* och andra hjälpmedel som underlättar olika typer av inmatning till programmet förenklas också med ABC1600 FÖNSTERHANTERARE. Hjälpmedlen gör datoranvändandet betydligt smidigare. Framförallt undviks tidsödande inmatningar av kommandon från tangentbordet.

FÖRPROCESSORN - WSHPP

Då fönsterhanteraren skall användas måste först en infil skapas. Detta görs med hjälp av förprocessorn - på engelska Window SHell PreProcessor, WSHPP. Infilen är en textfil som bland annat beskriver menyer, fönster och rullgardiner (eng: 'pull down menus') som används vid start av program som utnyttjar ABC1600 FÖNSTERHANTERARE.

Nyckelordet 'init' pekar ut var i programmet start skall ske. Då man angivit sitt val i den första menyn ger programmet information om vad som skall göras - starta ett program t ex. Man arbetar sedan vidare med nya menyval undan för undan medan WSH utför de operationer som angivits. Då exempelvis 'SHELL' väljs startas kommandot 'SHELL' varefter man återvänder till den ursprungliga menyn. Arbetet med menyval fortsätter till dess man avslutar med 'logout'.

KOMMANDOTOLKEN - WSH

Utdata från WSHPP ges i form av binärkod så att kommandotolken WSH (eng: 'Window SHell'), som är en koppling mellan användaren och ABC1600 FÖNSTERHANTERARE, kan arbeta snabbt och effektivt. För att veta vad som skall utföras börjar WSH med att läsa en datafil. Denna fil är binärkodfilen som skapats av förprocessorn, WSHPP.

2 PRESENTATION PÅ ABC1600's BILDSKÄRM

Beroende på vilken sorts information som skall ritas på bildskärmen används teckenmarkören eller grafikmarkören. Teckenmarkören används då hela tecken hanteras - bokstäver, siffror och övriga tecken. Då bilder - dvs diagram och figurer - skall byggas upp används grafikmarkören. Eftersom de båda markörerna alltid finns tillgängliga för användning kan både text och grafik blandas valfritt på en och samma bild. För att skärmen skall kunna utnyttjas till fullo för både breda och höga bilder kan bildskärmen vridas 90 grader. Benämningarna landskapmod och porträttmod (eng: 'landscape mode' och 'portrait mode') svarar mot de olika inställningarna.

2.1 TECKENMARKÖREN

Varje tecken på bildskärmen nås med en angivelse av dess koordinatpar (rad,kolumn). Denna angivelse benämns *adressering*. Adresseringen medför att teckenmarkören flyttas till avsett läge - markören placeras på det aktuella tecknet (eller läget för ett tecken om där inte redan finns något).

Normalt är bildskärmens yta uppdelad i 54 rader * 111 kolumner i landskapmod och 72 rader * 80 kolumner i porträttmod. Läget (rad,kolumn)=(1,1) är det övre vänstra hörnet (eng: 'home'), medan (54,111) eller (72,80) är det nedre högra hörnet.

2.2 GRAFIKMARKÖREN

Bildskärmen består av en punktmatrix om 768 * 1024 punkter (eng: 'pixels'). Både hela tecken och grafik ritas på bildskärmen genom att punkterna tänds eller släcks. Exempelvis är tecken i typsnitt A uppbyggda av 14 * 9 punkter i både landskapmod och i porträttmod. Dock ändras det utrymme varje

Origo är normalt nollpunkten, dvs har koordinaterna $(x,y) = (0,0)$. Dock kan det grafiska origot definieras om till valfritt ställe på bildskärmen. I den högra figuren ovan har det grafiska origot placerats i punkten $(895,575)$. Observera att koordinaterna för ett origo alltid är $(0,0)$ varför punkten $(895,575)$ nås med $(0,0)$ så snart origots placering omdefinieras.

Nysättning av det grafiska origots läge görs för att förenkla adresseringar på bildskärmen. Det grafiska origot är alltid osynligt och måste ges positiva koordinater. Om en adressering begärs till en punkt utanför bildskärmen eller fönstret ignoreras denna.

2.3 DET VIRTUELLA RUMMET

Då en stor eller komplex bild skall uppritas och bildskärmens $768 * 1024$ punkter inte räcker till används det *virtuella* rummet. Formatet på den bild som skall konstrueras bestämmer hur stort virtuellt rum som erfordras. Storleken är godtycklig och begränsas i praktiken av datorsystemets minneskapacitet. När det virtuella rummet definierats ritas bilden upp och man använder eller betraktar valfri del av bilden genom att flytta omkring en begränsad yta inom det virtuella rummets ram. Ytans storlek kan väljas valfritt från noll till att omfatta hela bildskärmen - dvs maximalt $768 * 1024$ punkter.

Då en del av det virtuella rummet skall visas på bildskärmen väljs det aktuella området ut genom att ett koordinatpar ges till datorn. Koordinatparet anger var i det virtuella rummet den begränsade ytans nedre vänstra hörn skall placeras. Om en adressering begärs till en punkt utanför det virtuella rummet ignoreras denna.

I figuren nedan används information från tre virtuella rum för att skapa den bild som skall visas på bildskärmen. I det nedre virtuella rummet har den begränsade ytan, som anger vilken del av det virtuella rummet som skall användas, markerats.

3 FÖRPROCESSORN - WSHP

De numeriska värden som ej specificerats tolkar fönsterhanteraren som noll.

3.1 INTRODUKTION TILL WSHP

Då fönsterhanteraren skall användas måste först en infil skapas. Detta görs med hjälp av WSHP. Infilen är en textfil som bland annat beskriver

- fönstrens storlek
- fönsterrubriker
- ikonernas storlek
- ikonernas funktioner
- bakgrundsmönster
- markörernas utseende

Genom att alltid använda WSHP för att göra dataunderlag till fönsterhanteraren kan framtida problem när det gäller inkompatibilitet helt undvikas. Program som automatiskt gör menyer behöver bara generera text som sedan bearbetas av WSHP.

Samtliga fönster som öppnas med hjälp av WSHP emulerar med få undantag terminalen DEC VT100.

Förutom att producera en infil till WSH kan WSHP producera enstaka datastrukturer till andra program som skapar fönster och ikoner och liknande.

3.2 SYNTAX

WSHP:s syntax är

```
wshpp Å-nÅ Å<infile>Å Å-x <struc> <outfile>
```

```
-x <struc> <outfile>... Å Å-o <outfile>Å
```

där *<infile>* är textfilen som skall bearbetas. Om infilen *<infile>* ej specificeras kommer 'standard-input' att användas istället.

Optionen '-o' specificerar filnamnet på resultatfilen från WSHP.

Optionen '-x' med sina två efterföljande argument specificerar en separat struktur som skall skrivas ut på en fil (se kapitel 3.6).

Om inga '-x'- eller '-o'-optioner ges skrivs data ut på filen '.window' av WSHP ('.window' är den fil WSH läser som standard). WSHP kommer inte att skapa någon sådan resultatfil om inga '-o'- och en eller flera '-x'-optioner specificeras.

Några eller alla resultatfiler kan ersättas med ett '-' med följden att motsvarande data skrivs ut på standard-utfilen. Alla meddelanden WSHP ges skrivs ut på '*standard error*' (felmeddelandefilen).

Optionen '-n' används då ingen resultatfil skall genereras. Felmeddelanden ges normalt på engelska, men byts till svenska då omgivning-variabeln (eng: 'environment variable') 'LANGUAGE' (som bestämmer standardspråket) sätts till 'swedish' (svenska) genom

```
LANGUAGE=swedish
```

eller kommandot

```
swedish
```

3.3 INFILENS UPPBYGGNAD

Infilen består av två typer av data - dataposter och aktionposter.

Det postspecifika numret skall vara ett heltal större än eller lika med ett. Alla rader som startar med ett '#'-tecken tolkas som kommentarer och ignoreras. Alla data- och aktion-posterna kan ges i godtycklig ordning.

DATAPOSTERNA:

| SATS | BESKRIVNING |
|-----------|---|
| window | Data för ett fönster. |
| icon | Data för en ikon. |
| string | En sträng att användas i både landskap- och porträtt-moderna. |
| pstring | En sträng att användas i porträtt-moden. |
| lstring | En sträng att användas i landskap-moden. |
| pointer | Data för skapandet av en mus-markör. |
| header | Data för en fönsterrubrik. |
| environ | Omgivningsträngar att användas för att modifiera omgivningar i både landskap- och porträtt-moderna. |
| penviron | Omgivningsträngar att användas i porträtt-moden. |
| lenviron | Omgivningsträngar att användas i landskap-moden. |
| directory | Anger ett biblioteknamn (eng: 'path name'). |
| command | Ett kommando (filnamnet och argumenten). |
| flags | Data för fönsterflaggor. |

zoomlist Data för en zoomlista.

substitute Ersättningstangenter för musen definieras -
- i de fall man ej har tillgång till en mus.

background Data för ett bakgrundmönster.

AKTIONPOSTERNA:

| SATS | BESKRIVNING |
|-----------------|--|
| init | Beskriver vad som händer vid initieringen. |
| menu | Beskriver ett menyfönster. |
| choice | Beskriver ett val som kan göras från ett menyfönster. |
| action | Beskriver vad som skall utföras vid respektive val. |
| terminal | Beskriver ett terminalfönster - i detta fall ett fönster som använder ett program. |

3.4 DATAPOSTER

Varje datapost namnges av det egna postnamnet omedelbart följt av ett nummer. Därpå följer ett ':' (kolon) som separerar namnet från det efterföljande datainnehållet som kan bestå av:

- a) En sträng ('string', 'pstring', 'lstring', 'directory').
- b) Flera strängar åtskiljda av ',' (kommatecken) ('environ', 'penviron', 'lenviron', 'command').

- c) Nyckelord (med motsvarande värden) och flaggor ('window', 'icon', 'pointer', 'header', 'flags', 'zoomlist', 'substitute', 'background').

Några exempel:

Strängen i a) är resten av raden efter första kolon.

```
string1:Nisse bor i ett litet rum.
pstring34:rad ett.Önrad två.
directory5:/usr/tommy
```

Strängarna i b) är de som finns mellan första kolon och ett komma eller en ny rad, mellan två kommatecken, eller mellan ett komma och en ny rad.

```
environ3:LANGUAGE=english,TERM=vt100
command76:bin/ls,ls,-l
```

Dataposter i c) består av fyrateckens nyckelord, eventuellt följda av ett värde, åtskiljda av kolon. Om det är ett numeriskt värde skall nyckelordet följas av ett '#' (bräddgård) och det numeriska värdet. Det numeriska värdet kan ges på antingen decimal, oktalt eller hexadecimal form.

```
window2:pxor#100:pyor#300:pxsi#400:pysi#400:ö
      :pvxo#0:pvyo#0:pvxs#400:pvys#400:ö
      :color#0:bord#1:pfnt#0x42:ö
      :pmod:stxt:novr:nmov
```

Syntaxen för de olika formerna är densamma som i språket C. Det innebär att värden som börjar med '0' tolkas oktalt, värden som börjar med '0x' eller '0X' (noll x) tolkas hexadecimalt, resterande tolkas decimalt. Då tecknet '#' saknas är satsen en flagga.

Om värdet är en sträng skall nyckelordet föjas av ett '='
(är lika med) och strängen avslutas med ett ':' eller ny rad.

En flagga består endast av ett nyckelord. Om nyckelordet är utskrivet är flaggan satt, om nyckelordet saknas är flaggan ej satt.

'ö' (baklänges snedstreck) kan användas som 'escape'-tecken i strängar. Detta fungerar precis som i språket C (Strängen 'öe' innebär ESCAPE - ASCII decimalt = 27). Inledande och avslutande ' ' (mellanslag) och tab-er är signifikanta i alla strängar. En rad kan fortsätta på nästa rad om raden avslutas med 'ö'.

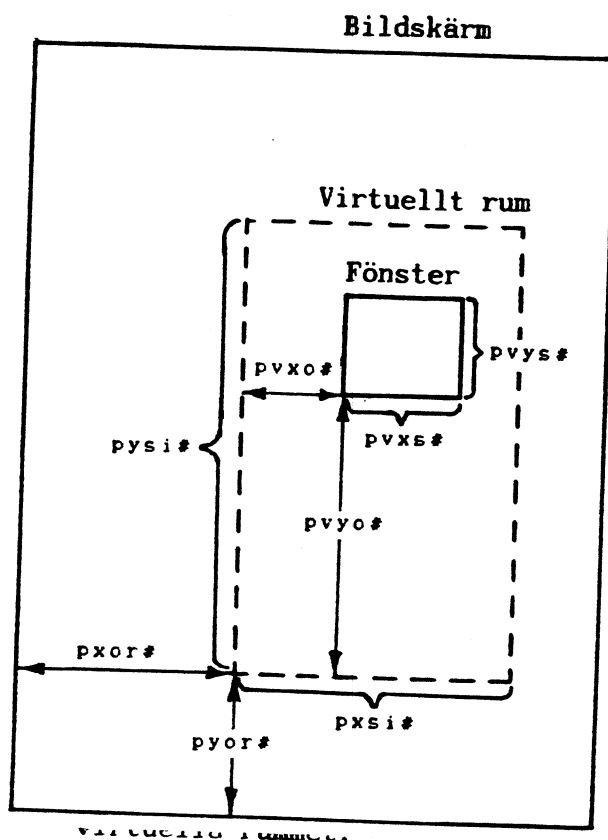
3.4.1 FÖNSTER

Dataposten 'window' ger data för ett fönster. Varje nyckelord har ett motsvarande element eller flagga i 'winstruc'-strukturen (se kapitel 6 REFERENSLISTA - nyckelord till WSHPP).

Figuren på nästa sida visar betydelsen av nyckelorden i porträttmoden. Motsvarande gäller för landskapmoden.

| NYCKELORD | BESKRIVNING |
|-----------|---|
| pxor# | porträtt-moden: x-koordinaten för det virtuella rummets nedre vänstra hörn. |
| lxor# | landskap-moden: x-koordinaten för det virtuella rummets nedre vänstra hörn. |
| pyor# | porträtt-moden: y-koordinaten för det virtuella rummets nedre vänstra hörn. |

Porträttmod



- $lxsi\#$ landskap-moden: storleken i x-led för det virtuella rummet.
- $pysi\#$ porträtt-moden: storleken i y-led för det virtuella rummet.
- $lysi\#$ landskap-moden: storleken i y-led för det virtuella rummet.

pvxo# porträtt-moden: x-koordinaten för det nedre vänstra hörnet av fönstret relativt det virtuella rummets nedre vänstra hörn.

lvxo# landskap-moden: x-koordinaten för det nedre vänstra hörnet av fönstret relativt det virtuella rummets nedre vänstra hörn.

pvyo# porträtt-moden: y-koordinaten för det nedre vänstra hörnet av fönstret relativt det virtuella rummets nedre vänstra hörn.

lvyo# landskap-moden: y-koordinaten för det nedre vänstra hörnet av fönstret relativt det virtuella rummets nedre vänstra hörn.

pvxs# porträtt-moden: fönstrets storlek i x-led.

lvxs# landskap-moden: fönstrets storlek i x-led.

pvys# porträtt-moden: fönstrets storlek i y-led.

lvys# landskap-moden: fönstrets storlek i y-led.

colr# fönstrets bakgrund: 0=svart, 1=vitt.

bord# typ av ram runt fönstret.

De olika ramtyperna är - ingen alls.
 1 ett streck.
 11 två streck.

| RAM | VÄ | HÖ | UPP | NBR |
|-----|----|----|-----|-----|
| 0 | - | - | - | - |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 11 | 11 | 11 | 11 |
| 3 | 11 | 1 | 1 | 1 |
| 4 | 1 | 11 | 1 | 1 |
| 5 | 1 | 1 | 11 | 1 |
| 6 | 1 | 1 | 1 | 11 |
| 7 | 11 | 11 | 1 | 1 |
| 8 | 11 | 1 | 11 | 1 |
| 9 | 11 | 1 | 1 | 11 |
| 10 | 1 | 11 | 11 | 1 |
| 11 | 1 | 11 | 1 | 11 |
| 12 | 1 | 1 | 11 | 11 |
| 13 | 11 | 11 | 11 | 1 |
| 14 | 11 | 11 | 1 | 11 |
| 15 | 11 | 1 | 11 | 11 |
| 16 | 1 | 11 | 11 | 11 |

pfnt# första typsnittet i porträtt-moden. Om decimal kod används betecknas typsnitt A med 65, om hexadecimal kod används betecknas A med 0x41.

lfnt# första typsnittet i landskap-moden.

usrb# största antal användardefinierade rutor som kan skapas vid ramens vänsterkant. Hur en ruta skapas beskrivs i den engelska handledningen 'USERS MANUAL ABC1600 WINDOW HANDLER', kapitel 'Window requests', stycke 'Add an user defined box'.

tsig# signal som används för att meddela att fönstret flyttats till den översta nivån. Skall vara '0' om ingen signal önskas.

- nsig#** signal som används för att meddela att fönstret flyttats från den översta nivån. Skall vara '0' om ingen signal önskas.
- rsig#** signal som används för att meddela ett fönster att det måste uppdateras eftersom fönstret självt eller något annat fönster har flyttats. För menyfönster ställer WSH denna till '0' om 'stxt'-flaggan är satt. Om 'stxt'-flaggan ej är satt kommer WSH att ge signalen ett lämpligt värde. För övriga fönster skall denna nollställas om ingen signal önskas.
- csig#** signal som skall sändas till processer i ett fönster där ikonerna 'cbox' används. Skall vara '0' om ingen signal önskas.
- hscr** de tre ikonerna som flyttar texten i x-led skall visas i fönsterramen.
- vscr** som 'hscr' men flyttar texten i y-led.
- cbox** ikonerna 'close box' som används för att stänga fönster skall visas i fönsterramen. WSH återställer denna flagga för menyfönster.
- sbox** ikonerna 'size box' som används för att ändra fönstrets storlek skall visas i fönsterramen.
- mbox** ikonerna 'move box' som används för att flytta fönstret skall visas i fönsterramen.
- zbox** ikonerna 'zoom box' som används för att ändra textstorleken skall visas i fönsterramen.
- avis** Ikonerna för att flytta texten i x- eller y-led är endast synliga om ej hela det virtuella rummet är synligt i x- respektive y-led. Detta gäller helt oberoende för de båda ikonerna.

- bbox ikonen 'blow up' skall visas i fönsterramen. Växlar mellan ursprunglig och senaste fönsterstorlek och -placering.
- hbox ikonen 'help box' som är en hjälpruta skall visas i fönsterramen.
- pmod porträtt-modens koordinater ges.
- lmod landskap-modens koordinater ges.
- stxt spara det virtuella rummets textinnehåll.
- lock lås fönstret på den översta nivån.
- novr fönstret får ej överlappas av annat fönster.
- ncur teckenmarkören (eng: 'text cursor') görs osynlig.
- nmov fönstret får ej flyttas eller ändra storlek.
- alls fönstret måste motsvara hela det virtuella rummet.
- spec speciellt fönster - dessa är alltid av lägre nivå än övriga fönster. De är främst tänkta att användas som menyfönster. Ett speciellt fönster kan ej förekomma som toppnivåfönster.
- kscr säkerställer att teckenmarkören är synlig i fönstret varje gång en tangent nedslås.
- wscr säkerställer att teckenmarkören är synlig i fönstret varje gång något har skrivits i fönstret.
- amsp reserverar utrymme så att en egen musmarkör för fönstret kan användas.
- rltv lägg till aktuellt fönster till faderfönstret (relativt det nedre vänstra hörnet).

- ncpi förhindrar att text kopieras in i det aktuella fönstret.
- ncpo förhindrar att text kopieras från det aktuella fönstret.
- text storleken av det virtuella rummet, fönstret, och fönstrets ursprung förväntas ges i form av tecken i stället för punkter. Om text-flaggan är satt refererar elementen 'wp_vxorig', 'wl_vxorig', 'wp_vyorig' och 'wl_vyorig' till det övre vänstra hörnet istället för det nedre vänstra hörnet - dvs man övergår till adressering enligt teckenmoden.
- wgrp detta fönster skall tillhöra en grupp av fönster.
- rulc detta fönster skall följa relativt faderfönstrets övre vänstra hörn. (Relevant endast då fönstret har ett faderfönster).
- rurc detta fönster skall följa relativt faderfönstrets övre högra hörn. (Relevant endast då fönstret har ett faderfönster).
- rllc detta fönster skall följa relativt faderfönstrets undre vänstra hörn. (Relevant endast då fönstret har ett faderfönster).
- rlrc detta fönster skall följa relativt faderfönstrets undre högra hörn. (Relevant endast då fönstret har ett faderfönster).

Följande exempel visar hur det virtuella rummets nedre vänstra hörn placeras på bildskärmens koordinat $(x,y)=(350,400)$. Fönstrets nedre vänstra hörn placeras i $(0,0)$ relativt det virtuella rummet. Både fönstrets och det virtuella rummets storlek är $(150,200)$. Fönstrets adresserbara område har omfattningen $(0,0)$ till $(149,199)$. Färg=Vit, ram=dubbel runt om, typsnitt=B.

Fönstret kan användas i både porträtt- och landskap-mod eftersom koordinater har angivits för båda.

I ramen kommer ikonerna 'close box', 'size box', 'move box' och 'zoom box' att presenteras.

Fönstrets textinnehåll kommer att sparas.

```

window6:pxor#350:pyor#400:pxsi#150:pysi#200:ö
        :pvxo#0:pvyo#0:pvxs#150:pvys#200:ö
        :lxor#350:lyor#400:lxsi#150:lysi#200:ö
        :lvxo#0:lvyo#0:lvxs#150:lvys#200:ö
        :colr#1:bord#2:pfnt#66:lfnt#66:ö
        :pmod:lmod:cbox:sbox:mbox:zbox:stxt

```

3.4.2 IKONER

Dataposten 'icon' ger data för en ikon. Varje nyckelord har ett motsvarande element eller flagga i 'winicon'-strukturen (se kapitel 6 REFERENSLISTA - nyckelord till WSHPP).

NYCKELORD - BESKRIVNING

- pxor# porträtt-moden: x-koordinaten för ikonens nedre vänstra hörn.
- lxor# landskap-moden: x-koordinaten för ikonens nedre vänstra hörn.
- pyor# porträtt-moden: y-koordinaten för ikonens nedre vänstra hörn.

- lyor# landskap-moden:y-koordinaten för ikonens nedre vänstra hörn.
- pxsi# porträtt-moden: storleken i x-led.
- lxsi# landskap-moden:storleken i x-led.
- pysi# porträtt-moden: storleken i y-led.
- lysi# landskap-moden:storleken i y-led.
- cseq= specificerar en teckensekvens. Det program som läser fönstret uppfattar denna sekvens som om den inmatats från tangentbordet.
- pmod koordinater ges för porträtt-moden.
- lmod koordinater ges för landskap-moden.
- pres sekvens sänds då vänster knapp nedtrycks.
- rlse sekvens sänds då vänster knapp släpps.
- inve invertera den ikon vi pekar på.
- entr sekvens sänds då vi flyttar oss in till ikonen.
- leav sekvens sänds då vi flyttar oss från ikonen.
- rmov avlägsna ikonen då en sekvens sänts.
- rqst sänd endast sekvensen om det finns en väntande läs-begäran (eng: 'read request') till fönstret.
- schk kontrollera om 'entr' eller 'leav' är fullbordade då ikonen placeras. Om 'entr' är satt och musmarkören pekar på området där ikonen skall placeras eller om 'leav' är satt och musmarkören pekar utanför det område där ikonen skall placeras sänds teckensekvensen.

Den sänds vid det tillfälle ikonen placeras på bildskärmen - dvs endast just då ikonen skapas.

lzer sekvensen sänds endast om fönstret är på den översta nivån.

text ikonkoordinaterna förväntas vara givna i form av tecken.

Följande exempel placerar ikonen i det nedre vänstra hörnet av ett virtuellt rum. Storleken på ikonen $(x,y)=(150,75)$.

Ikonen kan användas i både porträtt- och landskap-mod eftersom koordinater har angivits för båda.

Ikonen sänder en sekvens då vänster knapp nedtrycks eller släpps och inverteras då man pekar på den.

```
icon23:pxor#0:pyor#0:pxsi#150:pysi#75:ö
      :lxor#0:lyor#0:lxsi#150:lysi#75:ö
      :pmod:lmod:pres:rlse:inve
```

3.4.3 STRÅNGAR: string, pstring, lstring

Följande exempel visar hur man genom att använda de rubricerade 'string', 'pstring', eller 'lstring' skriver

Detta är rad 1

Detta är rad 2

string14: Detta är rad lÖn Detta är rad 2
 pstring2: Detta är rad lÖn Detta är rad 2
 lstring9: Detta är rad lÖn Detta är rad 2

(där 'Ön' betyder ny rad - LF)

Följande exempel visar hur en linje kan ritas från
 (x,y)=100,325 till (x,y)=675,800

```
string36:Öe:100;325mÖe:675;800;0;ld
pstring7:Öe:100;325mÖe:675;800;0;ld
lstring1:Öe:100;325mÖe:675;800;0;ld
```

3.4.4 MUSMARKÖR

Dataposten 'pointer' ger data för uppbyggnad av en musmarkör.
 Varje nyckelord har ett motsvarande element i 'npstruc'-strukturen (se kapitel 6 REFERENSLISTA - nyckelord till WSPPP).

NYCKELORD - BESKRIVNING

| | |
|-------|-------------------------------------|
| xsiz# | musmarkörens bredd |
| ysiz# | musmarkörens höjd |
| xpnt# | x-koordinaten för den pekande delen |
| ypnt# | y-koordinaten för den pekande delen |

Med nyckelorden 'andm#' (eng: 'ANDmask') och
 'orma#' (eng: 'ORMask') erhålls tre operationer

(ett- och nollställa samt lämna opåverkad) med vilkas hjälp man bygger upp musmarkören enligt formeln

((befintlig bakgrund) OCH 'andm#') ELLER 'orma#'.

andm# en serie om 32 OCH-masker för att bygga upp musmarkören. De olika elementen skiljs åt med ', '.

- Ettställd bit påverkar ej bildskärmen.
- Nollställd bit skapar mörkt fält i musmarkören.

orma# en serie om 32 ELLER-masker för att bygga upp musmarkören. De olika elementen skiljs åt med ', '.

- Ettställd bit skapar ljust fält i musmarkören.
- Nollställd bit påverkar ej bildskärmen.

Följande tre exempel visar hur tre olika musmarkörer byggs upp. Markörernas storlek är 8 * 8 punkter. Den pekande delen av markören anges relativt det övre vänstra hörnet - i våra exempel (x,y)=(4,4), dvs den övre vänstra nollan av de fyra i mitten.

andm# Den första markörens utseende bestäms endast av 'andm#' eftersom 'orma#' nollställts. Figuren nedan visar koden på hexadecimal form och markörens utseende - ett svart kors.

```
pointerl:xsiz#8:ysiz#8:xpnt#4:ypnt#4:0
      :andm#0xe7ffffff,0xe7ffffff,0xe7ffffff,0
          0x00ffffff,0x00ffffff,0xe7ffffff,0
          0xe7ffffff,0xe7ffffff:0
      :orma#0,0,0,0,0,0,0,0
```

```
e7ffffff = 11100111111111111111111111111111
e7ffffff = 11100111111111111111111111111111
e7ffffff = 11100111111111111111111111111111
00ffffff = 00000000111111111111111111111111
00ffffff = 00000000111111111111111111111111
e7ffffff = 11100111111111111111111111111111
e7ffffff = 11100111111111111111111111111111
e7ffffff = 11100111111111111111111111111111
```

orma# Den andra markörens utseende bestäms endast av 'orma#' eftersom 'andm#' ettställt. Figuren nedan visar koden på hexadecimal form och markörens utseende - en vit rektangel.

```
pointerl:xsiz#8:ysiz#8:xpnt#4:ypnt#4:ö
      :andm#0xffffffff,0xffffffff,0xffffffff,ö
          0xffffffff,0xffffffff,0xffffffff,ö
          0xffffffff,0xffffffff:ö
      :orma#0x00000000,0x00000000,0x00000000,ö
          0x00000000,0xff000000,0xff000000,ö
          0xff000000,0xff000000
```

```
00ffffff = 00000000000000000000000000000000
00ffffff = 00000000000000000000000000000000
00ffffff = 00000000000000000000000000000000
00ffffff = 00000000000000000000000000000000
ffffffff = 11111111000000000000000000000000
ffffffff = 11111111000000000000000000000000
ffffffff = 11111111000000000000000000000000
ffffffff = 11111111000000000000000000000000
```

Både 'andm#' och 'orma#'

Den tredje markörens utseende bestäms både av 'andm#' och 'orma#'. Figuren nedan visar koden på hexadecimal form och markörens utseende - den övre hälften bestäms av 'andm#' och den undre av 'orma#'.

```
pointerl:xsiz#8:ysiz#8:xpnt#4:ypnt#4:0
:andm#0xe7ffffff,0xe7ffffff,0xe7ffffff,0
0x00ffffff,0x00ffffff,0xe7ffffff,0
0xe7ffffff,0xe7ffffff:0
:orma#0x00000000,0x00000000,0x00000000,0
0x00000000,0xff000000,0xff000000,0
0xff000000,0xff000000
```

```
11100111
11100111
11100111
00000000
11111111
11111111
11111111
11111111
```

3.4.5. RUBRIK

Dataposten 'header' ger data för en fönsterrubrik. Varje nyckelord har ett motsvarande element eller flagga i 'headstruc'-strukturen (se kapitel 6 REFERENSLISTA - nyckelord till WSHP).

| NYCKELORD | BESKRIVNING |
|-----------|---|
| head= | rubriksträngen. |
| invh | inverterad fönsterrubrik. |
| invt | inverterar toppnivåfönstrets fönsterrubrik. |

Följande exempel visar hur rubriken 'MITT PROGRAM' kan skrivas in i fönstrets ram. Rubriken är inverterad i förhållande till fönstret.

```
header28:head= MITT PROGRAM :invh
```

3.4.6 BIBLIOTEK

Följande exempel visar hur biblioteket /usr/sven/bin kan specificeras.

```
directory9:/usr/sven/bin
```

3.4.7 ENVIRON, PENVIRON, LENVIRON

Dessa dataposter specificerar hur programmens omgivning skall modifieras innan WSH kan exekvera program. Om den specificerade omgivningsvariabeln redan existerar kommer den senare att ersätta den förra. Om den aktuella omgivningsvariabeln däremot inte redan existerar kommer den att läggas till omgivningslistan (eng: 'environment list'). För att specificera 'PATH' till '/usr/mydir/bin' och 'TERM' till 'vt100' kan följande exempel användas

```
environ3:PATH=/usr/mydir/bin,TERM=vt100
```

Genom att modifiera omgivningen är det möjligt att, till program som använder termcap-filen ge information om det virtuella rummets storlek. Följande exempel gäller ett virtuellt rum som är av storleken 33 rader gånger 132 kolumner. 'abcl600w' är en rad i termcap-filen, och skall endast användas för detta ändamål.

Men eftersom användningen av biblioteket 'libtermcap.a' (ett C-bibliotek) förutsättes, beror resultatet av det program som använder TERMCAP-beskrivningen. Prova därför alltid funktionen innan den används - det finns fall där kombinationen ej fungerar.

Ett sätt att undvika felaktigt resultat av TERMCAP-beskrivningen är att skapa en helt ny terminalbeskrivning i termcap-filen. I den specificeras bl a det nya fönstrets exakta storlek.

```
environ7:TERM=win,TERMCAP=w0öwinöw:co#132:li#33:tc=abcl600w:
```

Motsvarande syntaxregler gäller för 'penviron' och 'lenviron'.

3.4.8 KOMMANDON

Följande exempel visar hur kommandot 'ls -l' specificeras. '/bin/ls' är filnamnet, 'ls' är argument '0' och '-l' är argument '1'.

```
commandl:/bin/ls,ls,-l
```


3.4.9 FLAGGOR

Dataposterna flags ger data för nya flaggor till redan öppnat fönster. Varje nyckelord har en motsvarande flagga i 'flgstruc'-strukturen (se kapitel 6 REFERENSLISTA - nyckelord till WSHP).

| NYCKELORD | BESKRIVNING |
|-----------|--|
| lock | lås fönstret på den översta nivån. |
| novr | fönstret får ej överlappas av annat fönster. |
| ncur | teckenmarkören (eng: 'text cursor') görs osynlig. |
| nmov | fönstret får ej flyttas eller ändra storlek. |
| alls | fönstret måste motsvara hela det virtuella rummet. |
| spec | speciellt fönster - hindrar fönstret från att definieras till översta nivån. |
| kscr | säkerställer att teckenmarkören är synlig i fönstret varje gång en tangent nedslås. |
| wscr | säkerställer att teckenmarkören är synlig i fönstret varje gång något har skrivits i fönstret. |
| ncpi | förhindrar att text kopieras in i det aktuella fönstret. |
| ncpo | förhindrar att text kopieras från det aktuella fönstret. |
| rulc | detta fönster skall följa relativt faderfönstrets övre vänstra hörn. (Relevant endast då fönstret har ett faderfönster). |

- rurc** detta fönster skall följa relativt faderfönstrets
övre högra hörn. (Relevant endast då fönstret har
ett faderfönster).
- rllc** detta fönster skall följa relativt faderfönstrets
undre vänstra hörn. (Relevant endast då fönstret
har ett faderfönster).
- rlrc** detta fönster skall följa relativt faderfönstrets
undre högra hörn. (Relevant endast då fönstret
har ett faderfönster).

Följande exempel visar hur nya flaggor kan specificeras för
ett fönster. I detta fall görs teckenmarkören osynlig. Fönstret
får ej flyttas eller ändra storlek.

flags25:ncur:mmov:

3.4.10 ZOOMLISTA

Dataposten 'zoomlist' ger data för en zoomlista. Varje
nyckelord har ett motsvarande element eller flagga i 'zoomlist'-
strukturen (se kapitel 6 REFERENSLISTA - nyckelord till WSHPP).

NYCKELORD - BESKRIVNING

plst= typsnittlistan för porträtt-moden
llst= typsnittlistan för landskap-moden
pmod ger porträtt-mod-listan
lmod ger landskap-mod-listan

Följande exempel visar en zoomlista som möjliggör växling mellan fönstrets standardtypsnitt och typsnitten I, L och R. Växlingen sker då man pekar på ikonen 'zoombox' och kan utföras både i porträtt- och landskap-mod.

```
zoomlistll:plst=ILR:llst=ILR:pmod:lmod
```

3.4.11 ERSÄTT EN MUS

Dataposten 'substitute' ger data för en uppsättning ersättningstangenter för musen (eng: 'mouse substitute keys'). Varje nyckelord har ett motsvarande element i 'substit'-strukturen (se kapitel 6 REFERENSLISTA - nyckelord till WSHP).

NYCKELORD - BESKRIVNING

init# flagga som visar huruvida de ersatta tangenterna vid uppstart är aktiva (eng: 'enabled') eller ej. '1' motsvarar aktiv tangent - vid nedtryckning av tangenten sänds en signal till datorn för tolkning. '0' motsvarar inaktiv (eng: 'disabled') tangent - som ersättningstangent för musen ignoreras varje nedtryckning.

Vid uppstart är de normalt inaktiva - växling sker med CTRL-SHIFT-PF15.

onof# tangent som används till att bestämma om musens ersättningstangenter skall vara inkopplade eller ej.

Vid uppstart är de definierade som CTRL-SHIFT-PF15.

mpup# flyttar musmarkören en steglängd uppåt.

mpdo# flyttar musmarkören en steglängd nedåt.

mple# flyttar musmarkören en stegl åt vänster.

mpri# flyttar musmarkören en stegl åt höger.

mpul# flyttar musmarkören en stegl uppåt vänster.

mpur# flyttar musmarkören en stegl nedåt vänster.

mpdl# flyttar musmarkören en stegl nedåt vänster.

mpdr# flyttar musmarkören en stegl nedåt höger.

lpup# flyttar musmarkören en lång steglängd uppåt.

lpdo# flyttar musmarkören en lång steglängd nedåt.

lple# flyttar musmarkören en lång stegl åt vänster.

lpri# flyttar musmarkören en lång stegl åt höger.

lpul# flyttar musmarkören en lång steglängd
uppåt vänster.

lpur# flyttar musmarkören en lång steglängd
uppåt höger.

lpdl# flyttar musmarkören en lång steglängd
nedåt vänster.

lpdr# flyttar musmarkören en lång steglängd
nedåt höger.

pcmd# ersättning för den vänstra musknappen.

cwin# ersättning för den högra musknappen.

mtxt# ersättning för den mellersta musknappen.

step# Steglängd för en normal förflyttning av musmarkören.

lstp# Steglängd för en lång förflyttning av musmarkören.

Följande exempel visar en start av ersättningstangenterna för musen. Med CTRL SHIFT PF15 övergår markörplaceraren från att ha styrt teckenmarkören till att istället styra musmarkören. Med CTRL SHIFT PF15 - samma kommando - växlar man tillbaka till den ursprungliga funktionen igen. PF13, PF14 och PF15 motsvarar vänster musknapp, mittenknappen resp höger knapp. Steglängden vid normal resp lång förflyttning sätts till 4 resp 16.

```

substitutel:init#0:onof#0xfe:ö
      :mpup#0xa1:mpdo#0xa3:mple#0xac:mpri#0xa4:ö
      :mpul#0xad:mpur#0xa5:mpdl#0xaf:mpdr#0xa7:ö
      :lpup#0xb1:lpdo#0xb3:lple#0xbc:lpri#0xb4:ö
      :lpul#0xbd:lpur#0xb5:lpdl#0xbf:lpdr#0xb7:ö
      :pcmd#0xcc:cwin#0xce:mtxt#0xcd:step#4:lstp#16

```

3.4.12 BAKGRUND

Dataposten bakgrund ger data för ett bakgrundsmönster. Nyckelordet har ett motsvarande element i 'chbgstruc'-strukturen (se kapitel 6 REFERENSLISTA - nyckelord till WSHPP).

| NYCKELORD | BESKRIVNING |
|-----------|---|
| bmap# | bitmönstret av en 16 gånger 16 -matris representerar mönstret. De 16 elementen skall åtskiljas av ',,'. |

Följande exempel skriver 'ABC1600' som bakgrundmönster.

```
background2:bmap#0x0,0x338e,0x4a50,0x4a50,0
0x7b90,0x4a50,0x4a50,0x4b8e,0
0x0,0x4c44,0x50aa,0x54aa,0
0x5aaa,0x52aa,0x4c44,0x0
```

Följande konvertering från hexadecimal kod till binär-kod bildar mönstret 'ABC1600'.

```
0x0      =0000000000000000
0x338e  =0011001110001110
0x4a50  =0100101001010000
0x4a50  =0100101001010000
0x7b90  =0111101110010000
0x4a50  =0100101001010000
0x4a50  =0100101001010000
0x4b8e  =0100101110001110
0x0      =0000000000000000
0x4c44  =0100110001000100
0x50aa  =0101000010101010
0x54aa  =0101010010101010
0x5aaa  =0101101010101010
0x52aa  =0101001010101010
0x4c44  =0100110001000100
0x0      =0000000000000000
```

3.5 AKTIONPOSTER

Varje aktionpost namnges av det egna postnamnet, i de flesta fall omedelbart följt av ett nummer. Därpå följer ett ':' som separerar namnet från den efterföljande beskrivande delen som består av dataposter, aktionposter, eller i vissa fall speciella aktioner.

3.5.1 INIT

Init-aktionen består av en lista av aktioner som skall utföras vid initiering. Aktionerna exekveras enligt den i listan specificerade ordningen. Följande aktioner kan utföras vid initieringssatser

| SATS | BESKRIVNING |
|------------|---|
| substitute | Anger de tangenter vilka används som ersättning för musen. Inga tangenter kommer att definieras om 'substitute' inte finns angivet. |
| background | Ett nytt bakgrundmönster. Standardbakgrunden används om ingen bakgrund finns angiven. |
| pointer | Uppbyggnaden av musmarkören. Om inte markörens utseende specificeras kommer markören att ritas upp med standardutseende. |
| terminal | Öppnar ett terminalfönster med ett program under exekvering i fönstret. |
| menu | Det menyfönster som visas vid start och vilket måste specificeras. |

inverse Skärmen sätts i läge 'inverse video'. Detta är en speciell aktion varför inget nummer skall anges.

normal Skärmen sätts i läge 'normal video'. Detta är en speciell aktion varför inget nummer skall anges.

I följande exempel visas en INIT-aktion. Aktionen innebär att en datapost 'substitute' - som specificerar ersättningstangenter för musen - har angetts. Efter detta fortsätter man automatiskt till aktionposten 'menu3'.

Endast en datapost 'init' kan specificeras varför inget INIT-nummer skall anges.

init:substitutel:menu3

3.5.2 MENY

Meny-aktionen beskriver ett menyfönster, en rullgardin eller dylikt. Följande satser kan specificeras:

| SATS | BESKRIVNING |
|--------|--|
| window | Data för fönstret .Om detta redan är öppnat kontrollerar WSH om fönstret innehåller de önskade strängarna och ikonerna. Om de finns där kommer de inte att anges på nytt. Om de inte redan finns där kommer de nya ikonerna att sättas upp och de nya strängarna att visas. Endast ett fönster skall specificeras. |

- header** Rubriken för en fönstermeny. Rubriken är en option.
- choice** Beskriver de val som är möjliga att göra från den aktuella menyn. Om ingen aktion specificeras måste ändå ett val anges.
- action** Den specificerade aktionen kommer att exekveras omedelbart. WSH väntar ej på inkommande val från musen. Om val redan har angetts kommer de att ignoreras.
- string** Innehåller grafik och tecken för menyfönstret. Strängarna kan användas i både landskaps- och porträttmoderna.
- pstring** Som 'string', men kan endast användas i porträttmod.
- lstring** Som 'string', men kan endast användas i landskapmod.
- pointer** Uppbyggnaden av musmarkören då den pekar på denna meny. Om ingen markör specificeras kommer den globala musmarkören att användas. 'amp'-flaggan för menyfönstret måste sättas för att det skall vara möjligt att bygga upp en egen musmarkör.

Om både en aktion och val ges kommer ett varningsmeddelande att skrivas ut på bildskärmen då WSHPP omvandlar infilen till binärformat. Detta informerar om att valen har ignorerats.

Följande exempel öppnar ett fönster som är beskrivet i dataposten 'window11'. Menyn innehåller också tre stycken val och en sträng. Beroende på om man har valt landskap- eller porträttmod används 'lstring12' eller 'pstring13'.

```
menu5:window11:choice20:choice21:choice22:pstring13:0
:lstring12
```

3.5.3 VAL

De aktioner som skall utföras sammanbinds med den aktuella valaktionen då den väljs. Följande två satser måste specificeras vid ett val

| SATS | BESKRIVNING |
|--------|--|
| icon | Ikonen. |
| action | De aktioner som skall utföras då ikonerna väljs. |

Följande exempel innebär att 'action17' utförs då man aktiverar 'icon7'.

choice9:icon7:action17

3.5.4 AKTIONER

Aktionen består av en lista av aktioner som skall utföras. De exekveras enligt den i listan specificerade ordningen. Följande satser kan specificeras

| SATS | BESKRIVNING |
|------------|--|
| flags | Nya fönsterflaggor för det aktuella menyfönstret. |
| substitute | Nya mus-ersättande tangenter definieras. |
| background | Nytt bakgrundmönster definieras. |
| pointer | Ny global musmarkör sätts upp. |
| terminal | Öppnar ett nytt terminalfönster med ett program under exekvering i fönstret. |

menu Gå till specificerad meny.

Följande speciella aktioner kan specificeras i en aktionlista
(inget nummer skall anges efter dessa speciella aktioner)

| SATS | BESKRIVNING |
|---------|---|
| close | Det aktuella menyfönstret stängs. |
| restore | Skärmen ritas upp pånytt. |
| inverse | Skärmen ställs i läge 'inverse video'. |
| normal | Skärmen ställs i läge 'normal video'. |
| top | Aktuellt menyfönster flyttas till toppnivån. |
| turn | Skärmen vrids. |
| logout | Utloggning. Exekveras endast om det ej finns några öppna terminalfönster. WSH kontrollerar detta automatiskt. |

Åtminstone en 'menu' måste ges. Om flera ges kommer ett varningsmeddelande att sändas ut. Även då 'menu' ej avslutar listan kommer ett varningsmeddelande att sändas ut.

I följande exempel ändras flaggorna för det aktuella menyfönstret i enlighet med specifikationen i 'flags5'. Därefter öppnas ett nytt terminalfönster såsom beskrivs i 'terminall0'. Till sist stängs menyfönstret och fortsättning sker till 'menu7'.

actionl8:flags5:terminall0:close:menu7

3.5.5 TERMINAL

Terminalaktionen beskriver ett fönster som skall användas till att exekvera ett program. Följande satser kan användas

| SATS | BESKRIVNING |
|-----------|--|
| window | Data för fönstret som skall användas som terminal. Högst ett fönster kan specificeras. Om inget fönster anges kommer satsen att exekveras med '/dev/null' som standard- infil, utfil och felmeddelande-fil. |
| header | Rubrik för terminalfönstret. |
| zoomlist | Zoomlista för terminalfönstret. |
| pointer | Uppbyggnaden av musmarkören då den pekar på terminalfönstret. Om ingen markör specificeras kommer den globala musmarkören att användas. 'amsp'-flaggan för terminalfönstret måste sättas för att göra det möjligt att bygga upp en egen musmarkör. |
| icon | Ikoner som skall definieras innan programexekveringen börjar. |
| string | Strängar som skall skrivas i terminalfönstret innan programexekveringen startar. Strängarna kan användas i både landskap- och porträttmoderna. |
| pstring | Som 'string', men kan endast användas i porträttmod. |
| lstring | Som 'string', men kan endast användas i landskapmod. |
| directory | Bibliotek att flytta till innan programexekveringen börjar. Om inget bibliotek anges kommer det bibliotek som används vid exekveringens början att vara det samma som användes vid WSH-exekveringens början. |

super Om den angetts kommer '*super channel*' öppnas som 'fd' (filtyp, eng: 'file descriptor') tre i programmet.

WSH styr vad som visas när ett nytt fönster startas. Normalt både öppnas och stängs fönster av WSH. Då ett program behöver tillgång till '*super channel*' anges att 'fd' skall motsvaras av nummer tre. Då WSH sedan startar det aktuella programmet blir 'fd' '*super channel*'. Detta är alltså ett sätt att ge flera program tillgång till '*super channel*'.

Eftersom detta är en speciell aktion skall inget nummer anges.

wait Beordrar WSH att vänta till dess att det specificerade kommandot avslutats. Detta är en speciell aktion varför inget nummer skall anges.

environ Modifiering av omgivningen.

penviron Som ovan, men endast giltigt för porträtt-moden.

lenviron Som ovan, men endast giltigt för landskap-moden.

command Specificerar programmet som skall exekveras i terminalfönstret.

I följande exempel beskriver '*window7*' det fönster som skall användas som terminal. Rubriken i fönstret beskrivs i '*header7*'. I '*string5*' - som kan användas i både landskap- och porträtt-mod - beskrivs fönstrets text. I '*directoryll*' anges namnet på det bibliotek dit förflyttning skall ske. '*command23*' specificerar sedan det program som skall exekveras.

`terminall5:window7:header7:string5:directoryll:command23`

3.6 SKRIVNING AV ENSTAKA STRUKTURER PÅ FIL

För att exempelvis skriva en fönsterstruktur ('winstruc') på en fil används optionen '-x'. Om en textfil - meny.wd - som innehåller en fönsterbeskrivning kallad window3 skall utnyttjas för att skriva fönsterdata beskriven av window3 på filen fön3 kan detta beordras genom

```
wshpp meny.wd -x window3 fön3
```

Följande datasatser kan separeras och skrivas på filen:

```
window  
header  
icon  
pointer  
flags  
zoomlist  
substisute  
background
```

3.7 EXEMPEL

EXEMPEL 1

Följande exempel visar hur en infil som bestämmer hur menyfönstret kan se ut vid inloggning. Infilen ges som indata till WSHPD enligt

```
wshpp infil
```

och ger som resultat en utfil med beteckningen '.window'. Filen '.window' läses automatiskt då en användare, som specificerat att fönsterhanteraren skall användas, loggar in.

Observera att utloggning ej kan göras i nedanstående exempel då detta är en oändlig loop.

```
#
# Detta skall göras vid initieringen
#
init:menul

#
# Menyn har bara ett val
#
menul:windowl:stringl:choicel

#
# Menyfönstret
#
windowl:pxor#350:pyor#400:pxsi#150:pysi#150:ö
        :pvxo#0:pvyo#0:pvxs#150:pvys#150:ö
        :lxor#350:lyor#400:lxsi#150:lysi#150:ö
        :lvxo#0:lvyo#0:lvxs#150:lvys#150:ö
        :colr#1:bord#0:pfnt#65:lfnt#65:ö
        :pmod:lmod:spec

#
# Texten i menyfönstret
#
stringl:öeA2;2HText nr.1öeA7;2HText nr. 2

#
# Ikonen
#
iconl:pxor#0:pyor#0:pxsi#150:pysi#150:ö
      :lxor#0:lyor#0:lxsi#150:lysi#150:ö
      :pmod:lmod

#
# Val 1
#
choicel:iconl:actionl

#
# Vad skall göras vid val 1
#
actionl:menul
```

EXEMPEL 2

Följande exempel visar också (som EXEMPEL 1) hur en infil som bestämmer menyfönstrets utseende vid inloggning kan se ut, men är mer komplext.

```

#
# Detta skall göras vid initieringen
#
init:menul

#
# Menyn har fem val
#
menul:windowl:stringl:choice1:choice2:choice3:choice4:choice5

#
# Menyfönstret
#
windowl:pxor#0:pyor#963:pxsi#768:pysi#61:ö
        :pvxo#0:pvyo#0:pvxs#768:pvys#61:ö
        :colr#0:bord#0:pfnt#65:ö
        :pmod:ncur:nmov:spec

#
# Texten i menyfönstret
#
stringl:öeÅ2;7HSHELLöeÅ2;24HIKON 2öeÅ2;41HIKON 3öeÅ2;ö
58HIKON 4öeÅ2;74HLOGGA UT

#
# Ikon nr 1
#
iconl:pxor#15:pyor#0:pxsi#126:pysi#56:ö
      :pmod:inve:pres

```


Ikon nr 2

icon2:pxor#166:pyor#0:pxsi#128:pysi#56:0
:pmod:inve:pres

Ikon nr 3

icon3:pxor#319:pyor#0:pxsi#128:pysi#56:0
:pmod:inve:pres

Ikon nr 4

icon4:pxor#472:pyor#0:pxsi#128:pysi#56:0
:pmod:inve:pres

Ikon nr 5

icon5:pxor#625:pyor#0:pxsi#128:pysi#56:0
:pmod:inve:pres

Val 1

choice1:icon1:action1

Val 2

choice2:icon2:action2

Val 3

choice3:icon3:action2

```
#
# Val 4
#
choice4:icon4:action2

#
# Val 5
#
choice5:icon5:action3

#
# Vad skall göras vid val 1
#
action1:menul0

#
# Vad skall göras vid val 2, 3 och 4
#
action2:menul

#
# Vad skall göras vid val 5
#
action3:logout:menul

#
# 'rullgardinen' har två val
#
menul0>window10:stringl0:choicel0:choice20

#
# 'rullgardin'-fönstret
#
window10:pxor#5:pyor#902:pxsi#148:pysi#56:ö
      :pvxo#0:pvyo#0:pvxs#148:pvys#56:ö
      :colr#1:bord#1:pfnt#65:ö
      :pmod:ncur
```

```
#
#   Texten i 'rullgardin'-fönstret
#
string10:Ön ProgramÖnÖnÖr Shell

#
#   Ikon nr 10
#
icon10:pxor#0:pyor#0:pxsi#148:pysi#28:ö
      :pmod:inve:pres

#
#   Ikon nr 20
#
icon20:pxor#0:pyor#28:pxsi#148:pysi#28:ö
      :pmod:inve:pres

#
#   Val 10
#
choice10:icon10:action10

#
#   Val 20
#
choice20:icon20:action20

#
#   Vad skall göras vid val 10
#
action10:close:terminal20:menul

#
#   Vad skall göras vid val 20
#
action20:close:terminal30:menul

#
#   'Shell-terminal'
#
terminal20>window20:header20:command20
```

```
#
# Terminalfönstret
#
window20:pxor#24:pyor#588:pxsi#720:pysi#336:ö
      :pvxo#0:pvyo#0:pvxs#720:pvys#336:ö
      :colr#1:bord#2:pfnt#65:ö
      :pmod:mbox:stxt

#
# Rubrik
#
header20:head= Shell :invt

#
# Starta upp 'shell'
#
command20:/bin/sh,sh

#
# Programterminal
#
terminal30>window30>header30>command30

#
# Terminalfönstret
#
window30:pxor#50:pyor#50:pxsi#650:pysi#550:ö
      :pvxo#0:pvyo#0:pvxs#650:pvys#550:ö
      :colr#0:bord#2:pfnt#65:ö
      :pmod

#
# Rubrik
#
header30:head= Program :invt

#
# Starta 'a.out'
#
command30:/usr/kurs/a.out,a.out
```

4 KOMMANDOTOLKEN - WSH

4.1 INTRODUKTION TILL WSH

För att veta vad som skall utföras börjar WSH med att läsa en datafil. Denna fil har skapats av förprocessorn, WSHP. Beskrivningen av WSHP i kapitel 3 täcker det mesta rörande WSH, varför detta avsnitt endast beskriver WSH:s syntax och ger exempel på hur WSH uppför sig i några olika situationer.

4.2 SYNTAX OCH IGÅNGSÄTTNING

WSH:s syntax är

```
wsh Å-nÅ Å<fil>Å
```

där <fil> är infilen. Om denna inte specificeras kommer WSH att försöka läsa filen '.window' i hemmabiblioteket. Om det inte är möjligt försöker WSH:n slutligen läsa filen '/etc/.window'. Normalt aktiverar WSH fönsterhanteraren efter att ha läst infilen. Emellertid talar optionen '-n' om att så ej skall ske. Istället räknar WSH med att fönsterhanteraren redan har aktiverats och att fönsterhanterarens 'super channel' är av filtyp (eng: 'file descriptor') tre.

WSH kan även aktiveras på fönster under den översta nivån. Förutsatt att menyfönster som tidigare öppnats har stängts innan man fortsätter till en lägre nivå, kan WSH startas på ett godtyckligt antal nivåer. I praktiken är det operativsystemets kapacitet att hantera flera samtidiga processer som begränsar det antal nivåer på vilka WSH kan startas. Aktivering av flera WSH möjliggörs med aktion-satserna 'super' och 'wait' (se 3.5.5 sid 47).

I följande exempel startas den första WSH i hemma-biblioteket (eng: 'home directory'). Vi passerar från hemma-biblioteket till underbiblioteket subdir. Eftersom subdir innehåller en annan fil '.window' med nya fönsterdata ändras bilden på bildskärmen. Då vi återvänder till hemma-biblioteket avslutas den andra WSH som aktiverats i subdir och den ursprungliga WSH som väntat på reaktivering fortsätter automatiskt.

```
commandl0:/usr/window/wsh,whs,-n
directoryll:subdir
terminal2l:wait:super:directoryll:commandl0
```

Om WSH startats från en annan terminal än systemets huvudterminal eller från ett fönster kommer standardtolken (eng: 'ordinary SHell') SH att aktiveras istället.

Från WSH ges normalt felmeddelanden på engelska, men byts till svenska då standardspråket 'LANGUAGE' sätts till svenska ('swedish').

4.3 NÅGRA KOMMENTARER TILL ANVÄNDNING AV WSH

- Då WSH skall hämta ett kommando från ett menyfönster kontrolleras att fönstret är öppnat. Vid behov öppnar WSH fönstret. Därefter kontrolleras att innehållet (rubriker, strängar och ikoner) är det rätta, att gammalt resp nytt innehåll avlägsnats resp skrivits in, samt att de specificerade strängarna skrivs in på menyfönstret.

- WSH ger automatiskt en uppdatering-signal (eng: 'Redraw SIGnal', 'rsig') för alla menyfönster som inte har flaggan 'stxt' satt och sköter uppdateringen av dessa. Om flaggan 'stxt' är satt kommer WSH att sätta 'rsig' till '0' och ger fönsterhanteraren i uppgift att sköta uppdateringen. Observera att samma fönsterbeskrivning ('window') ej bör användas för menyfönster och terminalfönster eftersom WSH manipulerar 'rsig' hos menyfönstren.

- Det finns ingen anledning att specificera teckensekvensen (eng: 'Character SEquence', 'cseq') som sänds från ikoner använda i menyfönster eftersom WSH använder sina egna sekvenser. Observera att av samma anledning bör ej heller ikonbeskrivningen ('icon') användas för menyfönster och terminalfönster eftersom WSH manipulerar 'rsig'.

- Signalerna 'tsig' och 'nsig' samt flaggan 'cbox' återställs alltid för menyfönster. Anledningen är att WSH ej kan hantera dessa.

- Markörerna flyttas ej till sina utgångslägen (eng: 'home') förrän de specificerade strängarna visas i ett fönster. Fönstret rensas inte heller förrän strängarna visas. Om nödvändigt måste detta inkluderas i strängarna. Var speciellt uppmärksam med strängar som måste nyskrivas av WSH för att uppdatera menyfönster.

- Strängar skrivs alltid i den ordning som specificerats.

- Samtliga terminalfönster blir definierade som kontrollerande terminaler, dvs '/dev/tty' refererar till fönstret.

- De löpande processerna i olika terminalfönster hör till olika processgrupper.

- Endast filtyperna 0, 1 och 2 (standard-infil, -utfil, och fel-utfil) och ibland även 3 ('super channel') är öppna då ett kommando exekveras.

- Då kommandon exekveras sätts alla signaler till sina standardvärden (eng: 'default') utom de som specificeras av 'tsig', 'nsig', 'rsig', och 'csig', vilka istället ignoreras.

- Hemmabiblioteket för WSH är alltid det bibliotek från vilket WSH startats. Terminalfönster kommer från början att ha samma bibliotek (eng: 'directory') om ej något annat specificeras.

- Om filnamnet i ett kommando ('command') ej börjar med '/' (snedstreck) söker WSH automatiskt i alla bibliotek som specificerats av omgivning-variabeln 'PATH'.

- När WSH hanterar den speciella aktionen 'turn' kontrollerar WSH att inga terminalfönster är öppna. Om så är fallet stängs alla menyfönster och WSH exekverar aktionen 'init' i den nya skärm-moden. Om det finns ett eller flera öppna terminalfönster ignoreras 'turn'.

- När WSH hanterar den speciella aktionen 'logout' ignoreras den om det finns något fönster öppet förutom ett eller flera menyfönster.

5 ORDFÖRKLARINIGAR

Nedanstående begrepp är förklarade speciellt med tanke på användning i datorsammanhang och de tillämpningar denna handledning omfattar. Orden är markerade med *kursiv fetstil* i sakregistret och första gången de uppträder i texten.

- ASCII** American Standard Code of Information Interchange. En standardisering av de tecken som används i informationssammanhang. Standarden innebär att varje tecken har en given plats i en lista, med löpande numrering av tecknen. Här används sju-bitars ASCII vilket ger en lista med tecken från 0 till 127 decimalt. Exempel: 'J'=74 decimalt, '_'=95 decimalt.
- typsnitt** Teckenuppsättning av visst utseende. Stil.
- tecken-
sekvens** Teckenmängd ordnad i linjär följd.
- virtuellt
rum** Rum definierat av koordinatsystem som är oberoende av bildskärmen. (Se figur sid 13).
- fönster** Del av virtuellt rum som avgränsats för visning i form av en ruta på bildskärmen. (Se figur sid 13). Vanligen förekommer två typer av fönster.
- I ett menyfönster ges information och valmöjligheter - t ex öppna eller stänga fönster, fortsätta till andra menyer och starta program.
- I ett terminalfönster emuleras en bildskärmsterminal. Normalt förs dialogen med operativsystem och program från ett terminalfönster.
- rullgardin** Ett område på bildskärmen inom vilket en rad kan markeras. Den rad man pekar på får avvikande bakgrundsfärg. Med hjälp av musmarkören rullar man

- uppåt eller nedåt i rullgardinen, dvs flyttar den markerade raden uppåt eller nedåt.
- rulla** Kontinuerligt förskjuta fönster i höjdled eller sidled så att nya delar av bilden framträder i rutan - vanligen tecken.
- panorera** Som rulla, men används vanligen då bildskärmen visar grafer och diagram, till skillnad från text.
- markör** Flyttbart märke som för användaren visar aktuellt läge på bildskärmen.
- ikon** Symbol eller grafiskt tecken på bildskärmen som används för start och styrning av programfunktioner. Ikoner aktiveras till exempel då en markör pekar på ikonen och en tangent nedtrycks eller då markören flyttas till eller från ikonen. Ikoner kan även användas för att utmärka en delbild, exempelvis en viss kurva i ett diagram.
- zooma** Samlingsnamn för fortlöpande ändring av tecknens storlek (förstoring och förminskning).
- mus** Till utseendet musliknande positionerare som med handen förs över en yta.
- musmarkör** Med hjälp av musen flyttbart märke som för användaren visar aktuellt läge på bildskärmen.
- markör-placerare** Den stora vita kvadratiska tangenten på tangentbordet.
- adressera** Nå element genom angivelse av dess koordinater eller annan lägesbestämmande information.

- bitmappa** Bestämna utseende på visst element genom att adressera enskilda punkter. Bitmönster.
- 'super channel'** Då fönsterhanteraren aktiveras öppnas en specifik kanal till denna. 'super channel' används för överordnade operationer som ej kan utföras från vanliga fönster, t ex byte av bakgrundsmönster eller övergång från porträttmod till landskapmod. Se även 'super' sid 47.
- 'standard error'** Den fil där felmeddelanden skrivs ut.
- 'infile'** Infilen till WSHPP och WSH. Infilen till WSHPP innehåller text (ASCII-tecken). Infilen till WSH är uppbyggd av binärkod. Denna infil är producerad av WSHPP och är alltså WSHPP's utfil.
- 'outfile'** Den utfil från WSHPP som är infil till WSH. Se 'infile'.
- standard-tolk** Kallas även SH (eng: 'SHell') eller skal. Läser av inmatningar från tangentbordet. Tolken ger en prompt, t ex 'n', varefter det inmatade kommandot exekveras - t ex starta program.
- kommando-tolk** Kallas även WSH (eng: 'Window SHell') eller fönstertolk. Läser av inmatningar från fönster. Tolken rit upp ett fönster varefter den specificerade uppgiften utförs - t ex starta ett program direkt från fönstret. Att låta WSH ersätta SH kan innebära avsevärda arbetsbesparingar.

6 REFERENSLISTA - nyckelord till WSHP

I listan nedan innehåller den vänstra kolumnen nyckelord från kapitel 3 och den högra innehåller dessas motsvarighet i den engelskspråkiga 'USERS MANUAL ABC1600 WINDOW HANDLER'.

Överst anges strukturnamnet för de i strukturen (eng: 'structure') ingående nyckelorden.

Med listans hjälp hittar man snabbt till rätt plats i den engelskspråkiga dokumentationen då mer information önskas.

winstruc

| | |
|-------|-----------|
| pxor# | wp_xorig |
| lxor# | wl_xorig |
| pyor# | wp_yorig |
| lyor# | wl_yorig |
| pxsi# | wp_xsize |
| lxsi# | wl_xsize |
| pysi# | wp_ysize |
| lysi# | wl_ysize |
| pvxo# | wp_vxorig |
| lvxo# | wl_vxorig |
| pvyo# | wp_vyorig |
| lvyo# | wl_vyorig |
| pvxs# | wp_vxsize |
| lvxs# | wl_vxsize |
| pvys# | wp_vysize |
| lvys# | wl_vysize |
| colr# | w_color |
| bord# | w_border |

| | |
|-------|------------|
| pfnt# | wp_font |
| lfnt# | wl_font |
| usrb# | w_uboxes |
| tsig# | w_tsig |
| nsig# | w_ntsig |
| rsig# | w_rsig |
| csig# | w_csig |
| hscr | BX_HSCR |
| vscr | BX_VSCR |
| cbox | BX_CLOS |
| sbox | BX_SIZE |
| mbox | BX_MOVE |
| zbox | BX_ZOOM |
| avis | BX_AVIS |
| bbox | BX_BLOW |
| hbox | BX_HELP |
| pmod | PMODE |
| lmod | LMODE |
| stxt | SAVETEXT |
| sbmp | SAVEBITMAP |
| lock | LOCK |
| novr | NOOVER |
| ncur | NOCURSOR |
| nmov | NOMOVE |
| alls | ALLSCR |
| spec | SPECIAL |
| kscr | KEYSCROLL |
| wscr | WRITSCROLL |
| amp | ALTMPNT |
| rltv | RELATIVE |

| | |
|------|---------|
| ncpi | NOCPIN |
| ncpo | NOCPOUT |
| text | TXTSIZE |
| | |
| wgrp | WGROUP |
| rulc | REL_ULC |
| rurc | REL_URC |
| | |
| rllc | REL_LLC |
| rlrc | REL_LRC |

winicon

| | |
|-------|------------|
| pxor# | ip_xorig |
| lxor# | il_xorig |
| pyor# | ip_yorig |
| | |
| lyor# | il_yorig |
| pxsi# | ip_xsize |
| lxsi# | il_xsize |
| | |
| pysi# | ip_ysize |
| lysi# | il_ysize |
| cseq= | i_cmdseqAA |
| | |
| pmod | I_PMODE |
| lmod | I_IMODE |
| pres | I_PRESS |
| | |
| rlse | I_RELEASE |
| inve | I_INVERT |
| entr | I_BNTER |
| | |
| leav | I_LEAV |
| rmov | I_REMOV |
| rqst | I_RQST |
| | |
| schk | I_SETCHK |
| lzer | I_LZERO |
| text | I_TEXT |

npstruc

| | |
|-------|----------|
| xsiz# | np_xsize |
| ysiz# | np_ysize |
| xpnt# | np_xpnt |
| ypnt# | np_ypnt |
| and# | np_andAA |
| or# | np_orAA |

headstruc

| | |
|-------|---------|
| head= | h_hdrAA |
| invh | H_INVHD |
| invt | H_INVT |

flgstruc

| | |
|------|------------|
| lock | LOCK |
| novr | NOOVER |
| ncur | NOCURSOR |
| nmov | NOMOVE |
| alls | ALLSCR |
| kscr | KEYSCROLL |
| wscr | WRITSCROLL |
| ncpi | NOCPIN |
| ncpo | NOCPOUT |
| rulc | REL_ULC |
| rurc | REL_URC |
| rllc | REL_LLC |
| rlrc | REL_LRC |

zoomlist

| | |
|-------|-----------|
| plst= | zp_listAA |
| llst= | zl_listAA |
| pmod | Z_PMODE |

lmod Z_LMODE

substit

| | |
|-------|-----------------|
| init# | c_initflg |
| onof# | c_keysÄS_ONOFFA |
| mpup# | c_keysÄS_MPUA |
| mpdo# | c_keysÄS_MPDA |
| mple# | c_keysÄS_MPLA |
| mpri# | c_keysÄS_MPRA |
| mpul# | c_keysÄS_MPULA |
| mpur# | c_keysÄS_MPURA |
| mpdl# | c_keysÄS_MPDLA |
| mpdr# | c_keysÄS_MPDRA |
| lpup# | c_keysÄS_IMPUA |
| lpdo# | c_keysÄS_IMPDA |
| lple# | c_keysÄS_LMPLA |
| lpri# | c_keysÄS_LMPRA |
| lpul# | c_keysÄS_LMPULA |
| lpur# | c_keysÄS_LMPURA |
| lpdl# | c_keysÄS_LMPDLA |
| lpdr# | c_keysÄS_LMPDRA |
| pcmd# | c_keysÄS_PCMDA |
| cwin# | c_keysÄS_CHWINA |
| mtxt# | c_keysÄS_MCAA |
| step# | c_step |
| lstp# | c_lstep |

chbgstruc

bmap# cb_bitmap

7 SAKREGISTER

De ord som markerats med *kursiv fetstil* i förteckningen nedan återfinnes i kapitel 5 ORDFÖRKLARINGAR.

| | |
|--|---------------------------|
| ABC1600 terminalemulator..... | 8, 14 |
| <i>adressera</i> | 25, 60 |
| användardefinierade rutor - se rutor | |
| <i>ASCII</i> | 7, 19, 59, 61 |
| automatisk mod - se moder | |
| bibliotek..... | 16, 33, 34, 46, 47, 55-58 |
| <i>bitmappa</i> | 61 |
| bitmönster - se även bitmapdata, bitmappa..... | 31, 32, 40 |
| 'Character SEQUENCE'..... | 57 |
| CR - se kontrolltecken | |
| DEC VT100..... | 8, 14, 18, 33, 34 |
| 'default'..... | 57 |
| '/dev/null' - se filer '/dev/null' | |
| '/dev/tty' - se filer '/dev/tty' | |
| 'directory' (directoryNN)..... | 18, 57, (47) |
| 'disabled'..... | 37 |
| ELLER-mask..... | 29-32 |
| emulera - se ABC1600 terminalemulator | |
| 'enabled'..... | 37 |
| 'environment'..... | 15, 17, 33, 34 |
| ersättningstangenter för musen..... | 17, 37-39, 41, 42, 44 |
| '/etc/.window' - se filer '/etc/.window' | |
| faderfönster..... | 24, 25, 35, 36 |
| felmeddelande..... | 15, 56, 61 |
| felmeddelandefilen - se filer 'standard error' | |
| 'file descriptor'..... | 47, 55 |
| filer | |
| '/dev/null'..... | 46 |
| '/dev/tty'..... | 57 |

| | |
|--|--|
| '/etc/.window'..... | 55 |
| 'standard error'..... | 15, 46, 57, 61 |
| 'termcap'..... | 18, 34 |
| '.window'..... | 15, 48, 55, 56 |
| filtyper | |
| 0 | standard-infil.....14, 15, 46, 57 |
| 1 | standard-utfil.....14, 15, 41, 57 |
| 2 | standard-fel-utfil...15, 46, 57, 61 |
| 3 | 'super channel'.....47, 55, 57, 61 |
| flagga..... | 16-19, 23, 26, 32, 35, 36, 43, 45, 46, 56 |
| fönster..... | 59 |
| - se även toppnivåF, speciellt F, moderF, grupp av F | |
| fönstermeny - se menyfönster | |
| fönsterskal - se kommandotolk | |
| fönsterram - se ram | |
| förklaringar av ord..... | 59 |
| förprocessor..... | 5, 8, 9, 14, 15, 19, 43, 48, 55, 61, 62 |
| grupp av fönster..... | 25 |
| handledningen..... | 5 |
| hemma-bibliotek..... | 55-57 |
| 'home'..... | 57 |
| 'home directory' - se hemma-bibliotek | |
| ikon.... | 8, 14, 16, 23, 24, 26-28, 37, 42, 44, 46, 56, 57, 60 |
| ikoner | |
| 'blow up'..... | 24 |
| 'close box'..... | 23, 26 |
| 'help box'..... | 24 |
| 'move box'..... | 23, 26 |
| 'size box'..... | 23, 26 |
| 'zoom box'..... | 23, 26 |
| infil..... | 5, 9, 14, 15, 43, 46, 48, 50, 55, 57, 61 |
| infile..... | 14, 15, 61 |
| inloggning..... | 48, 50 |
| 'inverse video'..... | 42, 45 |
| kanalen 'fd'..... | 47 |
| kommandotolk (WSH)..... | 5, 9, 55, 56, 61 |
| kommentarer..... | 56, 57 |

| | |
|--|---|
| kontrollkoder - se kontrolltecken | |
| kontrolltecken | |
| CR..... | 7 |
| CTRL-SHIFT-PF15..... | 37, 39 |
| LF..... | 29 |
| kopiera text..... | 25, 35 |
| '-l'..... | 34 |
| 'landscape mode' - se moder | |
| landskapmod - se moder | |
| 'LANGUAGE'..... | 15, 56 |
| LF - se kontrolltecken | |
| 'ls'..... | 34 |
| 'ls -l'..... | 34 |
| lång steglängd - se steglängd | |
| läsbegäran..... | 27 |
| <i>markör</i> | 8, 14, 60 |
| - se även hårkorsM, musM, teckenM | |
| <i>markörplacerare</i> | 8, 39, 60 |
| mask - se även ELLER-mask, OCH-mask..... | 29-32 |
| menyfönster..... | 4, 5, 7, 17, 23, 24, 41-45, 49, 50, 55-59 |
| moder | |
| landskap-mod..... | 16, 19-22 |
| 24, 26-28, 36, 37, 43, 46, 47, 61 | |
| porträtt-mod..... | 16, 19-22 |
| 24, 26-28, 36, 37, 43, 46, 47, 61 | |
| skärmmod..... | 58 |
| teckenmarkör - se teckenmarkör | |
| 'mouse substitute keys'..... | 17, 37, 41, 42, 44 |
| <i>mus</i> | 60 |
| <i>musmarkör</i> | 16, 24, 27, 29-32, 41, 43, 44, 46, 59, 60 |
| 'normal video'..... | 42, 45 |
| nyckelord..... | 6, 9, 62-66 |
| OCH-mask..... | 29-32 |
| omgivning-lista..... | 33 |
| omgivning-modifierare..... | 17, 18, 34, 47 |
| omgivning-sträng | |
| environ..... | 16 |
| lenviron..... | 16 |

| | |
|---|--------------------|
| penviron..... | 16 |
| omgivning-variabeln 'LANGUAGE'..... | 15 |
| omgivning-variabeln 'PATH'..... | 33, 58 |
| option..... | 15, 43, 48, 55 |
| '-n'..... | 15, 55 |
| '-o'..... | 15 |
| '-x'..... | 15, 48 |
| 'options' - se option | |
| ordförklaringar..... | 59 |
| 'ordinary SHell' (SH) - se standardtolk | |
| osynlig..... | 35, 36 |
| 'outfile'..... | 14, 61 |
| <i>panorera</i> | 60 |
| parametrar | |
| <struc>..... | 14 |
| 'path name'..... | 16 |
| 'PATH' - se omgivning-variabeln 'PATH' | |
| 'pixels'..... | 10 |
| 'portrait mode' - se moder | |
| porträttmod - se moder | |
| processgrupp..... | 57 |
| prompter..... | 61 |
| 'pull down menus'..... | 8 |
| ram, ramtyp..... | 12, 21-23, 33 |
| 'read request'..... | 27 |
| 'redraw signal'..... | 56 |
| referenslista..... | 62 |
| <i>rulla</i> | 60 |
| <i>rullgardin</i> | 8, 42, 59 |
| rullning - se rulla | |
| rutor..... | 22 |
| 'screen'..... | 6 |
| 'SH' - se standardtolk | |
| SHell - se standardtolk | |
| signaler..... | 22, 23, 37, 56, 57 |
| skal - se standardtolk | |
| skriva strukturer på en fil..... | 48 |

| | |
|--|---------------------------|
| skärmmod - se moder | |
| speciellt fönster..... | 24, 35 |
| 'standard error' (standard-fel-utfil) - se filer, filtyper | |
| 'standard input' (standard-infil) - se filtyper | |
| 'standard output' (standard-utfil) - se filtyper | |
| standardfil - se filtyper | |
| standardspråk 'LANGUAGE' - se 'LANGUAGE' | |
| <i>standardtolk</i> (SH)..... | 56, 61 |
| standardtypsnitt..... | 37 |
| steglängd..... | 37-39 |
| 'structure'..... | 62 |
| struktur - se 'structure' | |
| 'super channel' - se filtyper | |
| 'swedish'..... | 15, 56 |
| syntax..... | 5, 15, 55 |
| tecken | |
| ' , '..... | 19 |
| '*'..... | 16, 18 |
| ', '..... | 30, 40 |
| ': '..... | 17, 19, 41 |
| '='..... | 19 |
| 'ö'..... | 19 |
| teckenmarkör..... | 24, 35, 36, 39 |
| <i>teckensekvens</i> | 59 |
| 'TERM' - se filer 'termcap' | |
| 'termcap file' - se filer 'termcap' | |
| terminalfönster..... | 17, 41, 44-47, 54, 56, 59 |
| terminalen DEC VT100 - se DEC VT100 | |
| 'text cursor'..... | 24, 35 |
| tolk - se kommandotolk, standardtolk | |
| toppnivå..... | 24, 33, 45 |
| toppnivåfönster - se toppnivå | |
| <i>typsnitt</i> - se även standardtypsnitt..... | 22, 36, 37, 59 |
| typsnittlista..... | 36 |
| underbibliotek..... | 56 |

| | |
|---|-------------------------------|
| uppdatering-signal..... | 56 |
| utfil..... | 14, 43, 46, 48, 57, 61 |
| utloggning..... | 45, 48 |
| varningsmeddelande..... | 43, 45 |
| <i>virtuellt rum</i> | 12, 19-23, 25, 28, 34, 35, 59 |
| 'window' - se filer 'window' | |
| 'window shell' (WSH) - se kommandotolk | |
| 'window shell preprocessor' (WSHPP) - se förprocessor | |
| WSH - se kommandotolk | |
| WSHPP - se förprocessor | |
| <i>zooma</i> | 8, 60 |
| 'zoomlist' - se zoomlista | |
| zoomlista..... | 17, 18, 36, 37, 46, 48 |

Vi förbehåller oss rätten att justera produkten utan avisering samt full ansvarsfrihet för följden av tryckfel eller felaktig användning, Luxor Datorer AB.

produktion: IFS DATA AB Linköping och Luxor Datorer AB Motala