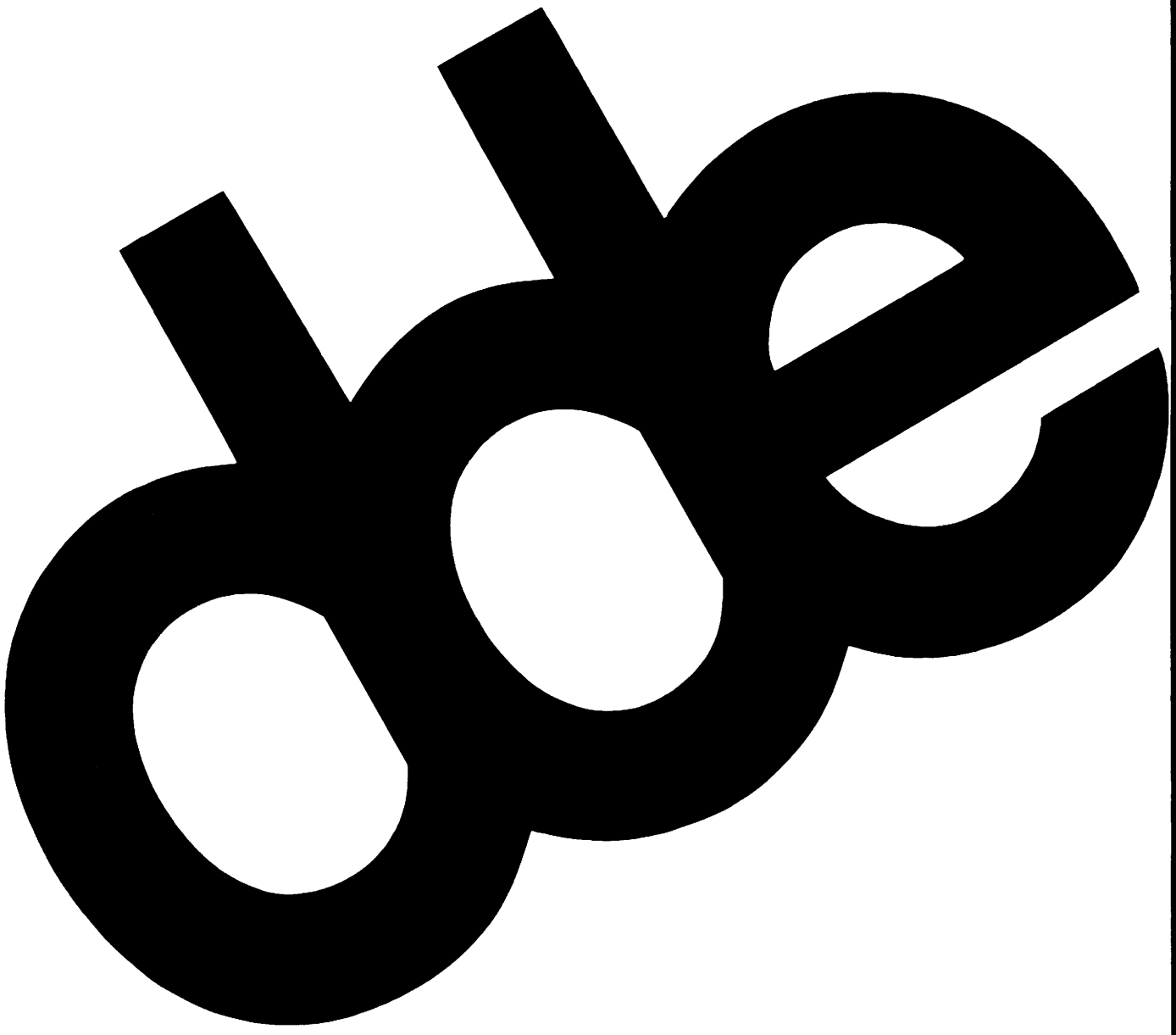
# UNIX SVR4.2

# Systemadministration I




Juli 1996

# Indholdsfortegnelse

# 1 Logbog og dokumentation

Når man skal til at administrere en UNIX server, er der en mængde litteratur samt administrative forholdsregler, som man bør have et godt kendskab til.

I dette indledende afsnit vil der blive fokuseret på at skabe et overblik over det, som er skrevet om UNIX SVR4.2 (udover-dette materiale), samt det som man selv burde skrive.
I et senere afsnit vil vægten blive lagt mere på de arbejds-mæssige aspekter og rutiner ved drift af en UNIX installation som systemadministrator.

## 1.1 Dokumentation

DDE leverer sammen med en Supermax Enterprise UNIX SVR4.2 server en lille guide med titlen "**Supermax Enterprise server ABC**". Som navnet antyder, giver denne bog et kort overblik over maskinen, maskinens funktionalitet samt hvorledes den sættes op ved levering.

"**System Administrator's Quick Reference**" er en anden lille sag, som fortæller om basal systemadministration. Emner som fx systemlog, kerne parametre, printer, mail oa., der adskiller sig fra andre hardwarefabrikanter, er nævnt i denne publika-tion.

Det kan varmt anbefales at læse begge bøger!

## 1.2 Logbog

Sammen med en Supermax Enterprise server levers der en logbog. Det er maskinens "blå bog/dagbog", hvor alt af betydning for drift og vedligehold af maskinen bør arkiveres.

I logbogen er der oprettet følgende afsnit på forhånd:

* **Meddelelser fra bruger til tekniker**
* **Meddelelser fra tekniker til bruger**
* **Servicerapporter**
* **HW oversigt + (følgesedler)**
* **SW oversigt + (følgesedler)**
* **Net konfiguration**
* **DDE-blok, disketter og bånd**

Når man ser under de enkelte faneblade i logbogen, afslører de intet andet, end hvad overskriften foreskriver. Men hver især repræsenterer hvert afsnit vigtige dele i serverens drift og vedligeholdelse. Derfor er det vigtigt at gemme informationen.

Andre afsnit, som med fordel kunne oprettes:

* **Konfiguration**
* **Shell scripts**
* **OS opdateringer**

Husk på, at al dokumentation ALTID skal opbevares i papirform, men meget gerne også i elektronisk form på serveren. For går serveren ned, vil man altid have informationerne ved hånden, hvis uheldet skulle være ude.

## 1.3 Litteratur

Med en Supermax Enterprise server leveres der en del litteratur fra **UNIX PRESS** til UNIX SVR4.2. Køber man selv ekstra litteratur, skal man huske, at den skal have påtrykt betegnelsen **"MP (MultiProcessor)"**, hvilket sikre, at indholdet også omfatter kommandoer og vejledning til brug ved håndtering af flere CPUer.

De enkelte serier fra UNIX PRESS er lavet i forskellige farver. En blå serie er for brugeren, en rød er reference manualer og en grøn er for administratoren.

**Den blå USER'S serie**

User's Guide
Guide to the UNIX Desktop

**Den røde REF (reference) serie**

| | |
|---|---|
| Command Reference (a-l) | 1,1C,1F,1M |
| Command Reference (m-z) | |
| Operating System API Reference | 2,3,3C,3E,3G,3I, 3M,3N,3S,3W,3X |
| System Files and Devices Reference | 4,5,7 |

(numrene yderst til højre er identiske med de henvisninger, som man får fra on-line manualerne, der ligger under kataloget **/usr/share/man/man#/***, og kaldes med UNIX kommandoen **/usr/bin/man** med en UNIX kommando som argument)

**Den grønne ADMIN (administration) serie**

! Basic Systemadministration
System Administration Volume 1
System Administration Volume 2
! Advanced Systemadministration
Network Adminisration
Audit Trail Administration

(! Udgået)

# 2 UNIX-kommandoer

Indledningsvis og som repetition skal nævnes nogle af de vigtigste UNIX-kommandoer og værktøjer, som en systemadministrator kommer til at bruge. Kommandoerne gennemgås med deres syntaks og en kort beskrivelse af, hvad de udfører. Der henvises i øvrigt til Command Reference (a-l og m-z), UNIX SVR4.2MP, UNIX Press.

**cat** SYNTAKS: cat [options] <fil>....
Udskriver indholdet af en eller flere filer. Det er ikke alle filer cat-kommandoen kan håndtere. Filerne skal være af typen ASCII text eller commands text. Dette problem kan dog afhjælpes, hvis man benytter option -v.

**cd** SYNTAKS: cd [<katalog>]
Skifter arbejdskatalog.

**chmod** SYNTAKS: chmod <rettigheder> <katalog>/<fil>....
Ændrer rettigheder på en eller flere filer eller kataloger. Man kan herved styre brugernes rettigheder til få adgang til filer og kataloger. Det er kun ejeren selv og systemadministratoren (superbrugeren), der har lov til at ændre rettigheder.

**chown** SYNTAKS: chown [options] <owner[:group]> <fil>....
Ændrer ejerforhold på en eller flere filer. Kan samtidig anvendes til at ændre filers gruppetilhørsforhold.

**cp** SYNTAKS: cp [options] <fil>.... <fil/katalog>
Kopierer en eller flere filer. Kildefilen nævnes først. Hvis destinationen er et katalog kan flere filer kopieres til dette katalog. Med option -r kan alle filer fra et katalog kopieres til et andet katalog.

**file** SYNTAKS: file [options] <fil>....
Klassificerer filtypen (f.eks. tekst, katalog eller programkode). Man bør være opmærksom på at det ikke altid er muligt for kommandoen at bestemme præcis hvilken type fil,

der er tale om. Kommandoen bør altid benyttes inden man begynder at læse en fil med cat eller pg, idet man herved undgår at læse rene program- eller datafiler, som kan få terminalen til at hænge.

**find** SYNTAKS: find <path liste> <udtryk>
Søger i den opgivne pathliste efter filer, der opfylder det opsatte søgeudtryk. Med find-kommandoen kan man f.eks. søge efter filer med et bestemt navn, med bestemte adgangskoder eller efter filer, der ikke har været læst eller modificeret i et bestemt tidsrum.

**fuser** SYNTAKS: fuser [options] <fil>....
Bruges med option -k og -u til at afbryde en hængende terminal.

**grep** SYNTAKS: grep [options] <udtryk> [<fil>....]
Søger efter et bestemt tekstmønster i en eller flere filer.

**kill** SYNTAKS: kill <-signalnr.> <procesnr.>....
Afbryder processer. Det kan være nødvendigt hvis en terminal hænger eller hvis en eller flere processer får systemet til at hænge. Signalnummer er et mål for hvor kraftigt et stopsignal, der sendes til processen fra kill-kommandoen. kill -9 vil normalt afbryde alle processer, den sendes til.

**lp** SYNTAKS: lp [options] <fil>....
Udskriver en eller flere filer på en tilsluttet printer i spoolersystemet. Når option -d bruges, kan man angive hvilken printer der skal udskrives på: lp -d<printernavn> <fil>. Ellers udskrives på default printeren.

**ls** SYNTAKS: ls [options] <katalog>.../<fil>...
Udskriver filoversigter. Hvor mange oplysninger, der medtages for hver fil afhænger af den eller de options, der er brugt. De vigtigste options er -a (medtager filer, der starter med .), -i (medtager i-node), -l (medtager rettigheder, filstørrelse, ændringsdato, ejer og gruppe).

**mkdir** SYNTAKS: mkdir <katalog>....
Opretter nye kataloger.

**mv** SYNTAKS: mv <fil> <fil> ell. mv <fil1>...<filn> <katalog> ell. mv <katalog1> <katalog2>
Omdøber/flytter filer eller kataloger til andre filer/kataloger.

**passwd** SYNTAKS: passwd [options] [login-navn]
Med passwd-kommandoen ændres eget password. Superbrugeren kan også ændre andres password. Med brug af forskellige options kan der desuden sættes maksimal varighed, minimums ændringsinterval og antal dage for advarsel før tvungen password-ændring.

**pg** SYNTAKS: pg [options] <fil>....
Bruges som cat-kommandoen. Med pg udskrives filen sidevis og man har mulighed for at bladre i filen.

**ps** SYNTAKS: ps [options]
Udskriver processtatus. Med denne kommando fås f.eks. PID (procesnummer), der kan benyttes i kommandoen kill.

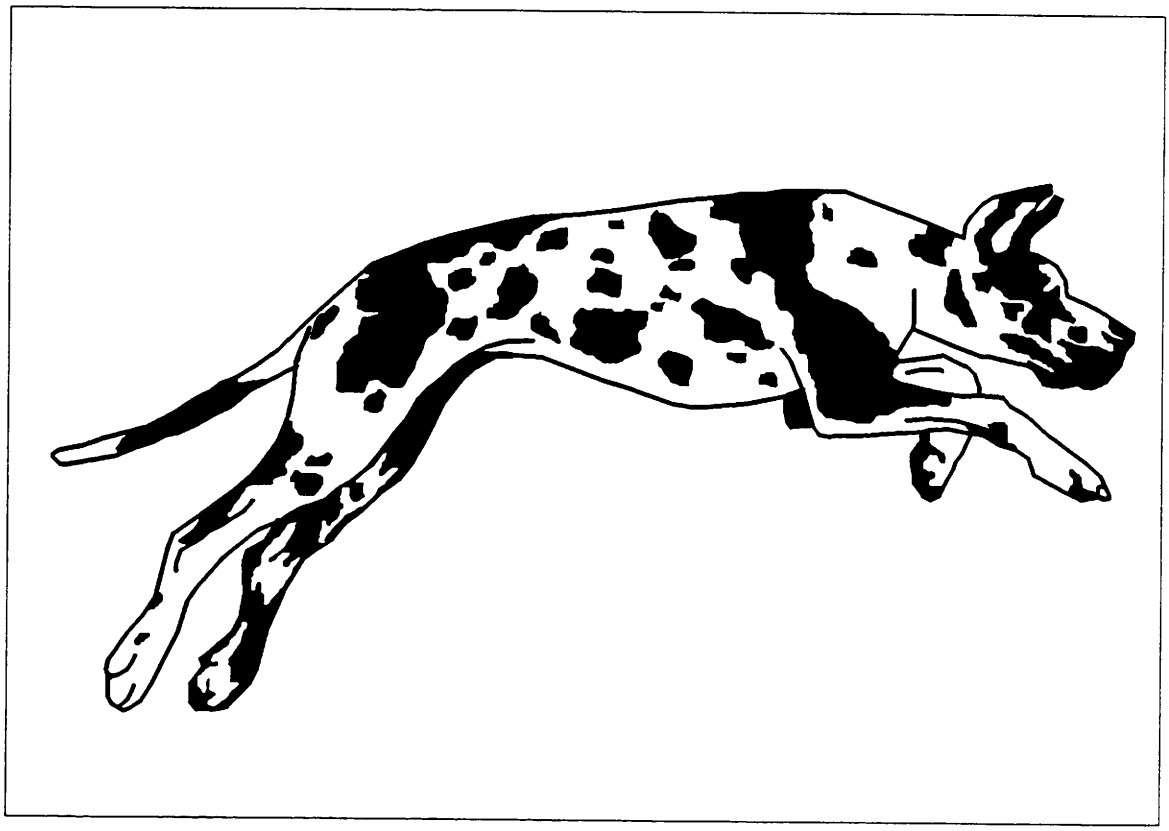
**pwd** SYNTAKS: pwd
Udskriver navn på arbejdskatalog.

**rm** SYNTAKS: rm [options] <fil>....
Sletter en eller flere filer.

**rmdir** SYNTAKS: rmdir [options] <katalog>....
Sletter en eller flere kataloger. Det er en forudsætning at kataloget ikke indeholder nogen filer.

**wc** SYNTAKS: wc [options] [<fil>....]
Tæller linier (option -l), ord (option -w) eller tegn (option -c) i en eller flere filer.

**who** SYNTAKS: who [options] [fil]
Lister forskellige oplysninger om indloggede brugere. Den mest almindelige option er am I (who am I). Optionen bevirker, at kommandoen udskriver den bruger, der er logget ind på den pågældende terminal.



# 2.1 Opgave

*Formål: - at repetere vigtige UNIX-kommandoer*

1. Log ind på systemet. Hvad hedder dit hjemmekatalog? Hvilke filer har du liggende i dit hjemmekatalog? Hvilke af disse filer kan du læse med cat eller pg.

2. Dan et underkatalog til dit hjemmekatalog. Kald det opgave-1. Hvad indeholder det nye katalog?

3. Kopier filerne fra dit hjemmekatalog over i det nye katalog. Slet filerne i dit hjemmekatalog.

4. Flyt alle filerne tilbage til dit hjemmekatalog.

5. Slet det nye katalog opgave-1 igen.

6. Ændr rettighederne på filerne i dit hjemmekatalog, således at der er læse- og skriverettighed for alle.

7. Ændr ejer for en af filerne i dit hjemmekatalog. Står filen stadig i fillisten? Ændre ejer tilbage igen. Hvad sker?

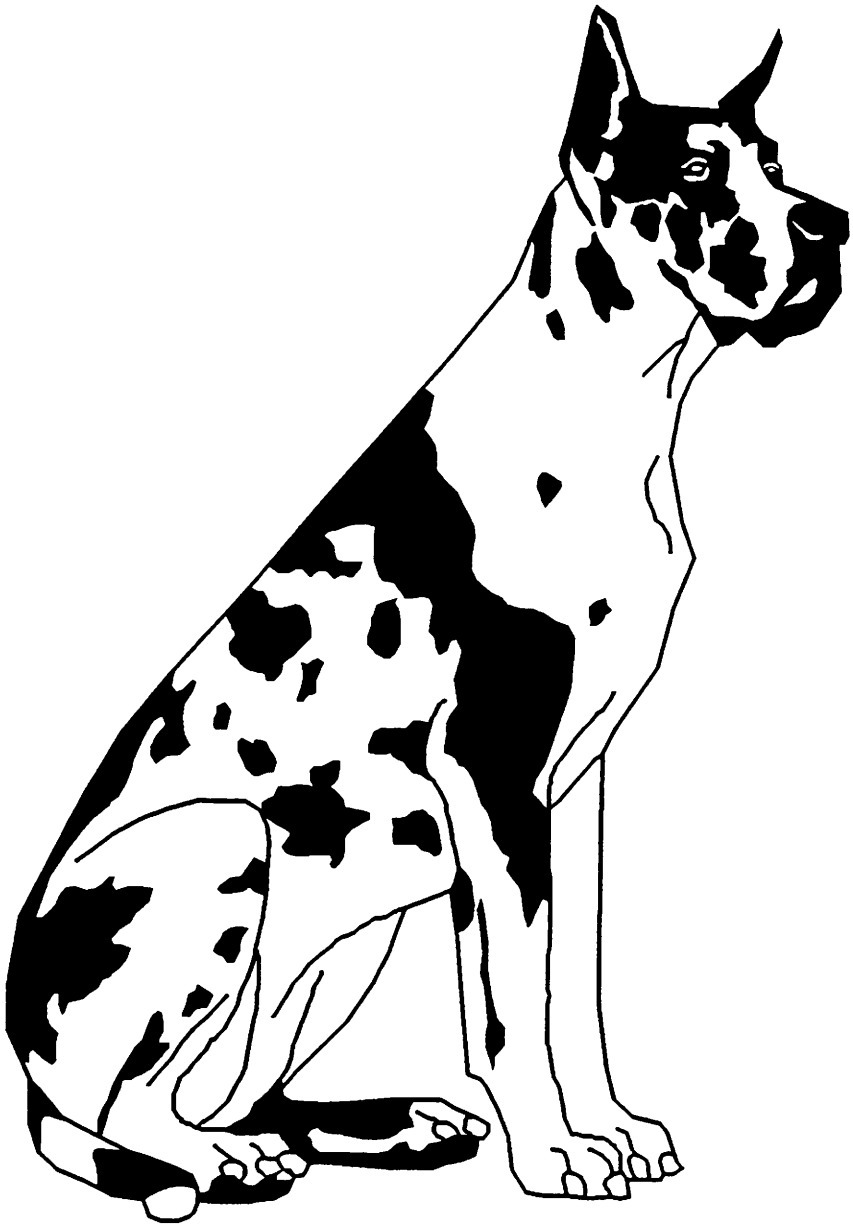
8. Find ud af hvilken terminal du sidder ved og hvilke processer du har kørende på din terminal.

9. Benyt kill til at stoppe dine egne processer. Hvad sker?

10. Log ind igen. Find ud af hvor der findes en fil, som hedder issue. Kan filen læses med pg eller cat? I bekræftende fald prøv at læse hvad filen indeholder.

11. Find ud af hvilke brugere, der er logget ind.

12. Ændr dit password. Husk det!



# 3 En systemadministrators opgaver

En systemadministrators opgaver/ansvar kan opdeles i fire hovedområder:

* **Drifttekniske opgaver og ansvar**
* **Sikkerhedsmæssige opgaver og ansvar**
* **Juridisk ansvar**
* **Økonomisk ansvar**

En ansvarsmæssig opdelingen kan laves på mange måder, og de enkelte områder kan nødvendigvis ikke adskilles. Vigtigt i denne diskussion er ikke, om hvorvidt den ene opdeling er at foretrække frem for den anden, men om opdelingen også dækker de vigtigste arbejdsopgaver, hvor et ansvar kan placeres.

Denne opdeling, som er valgt i dette materiale, dækker de mest fundamentale forpligtigelser, som en systemadministrator bør kende til. Nogle arbejdsopgaver er obligatoriske, hvorimod andre ikke vil eksisterer i en given organisation. EDB afdelingen vil ganske enkelt ikke stå for et evt. juridisk ansvar.

## 3.1 Drifttekniske opgaver og ansvar

At vedligeholde et system indbefatter mange opgaver afhængig af hvor stort systemet er og hvor meget soft- og hardware, der er på det. Men opgavernes omfang afhænger også af, hvor alvorligt systemadministratoren betragter denne del af sin funktion, og hvor meget tid han/hun har til sin rådighed.

### 3.1.1 Installation, vedligeholdelse og fejlretning

Installation og opgradering af både hard- og software er tilbagevendende opgaver. Nye programmer kommer på markedet, nye CPUer fremstilles. Disse opgaver hører til systemadministratorens arbejde, ofte i samarbejde med implementører fra sælgers side.

Udover at ansvaret for at opgaven løses hurtigt og helst uden problemer af nogen art for brugerne, skal systemadministratoren huske på at føre logbogen. Alt hard- og software i maskinen skal være dokumenteret udførligt til senere brug.

Fejl opstår af mange årsager på EDB systemer. Det er en af systemadministratorens fornemmeste opgaver i tilfælde af fejl at få systemet op at køre så hurtigt som muligt. Oppetiden, dvs. en stabil drift og dermed produktion, er en altafgørende parameter for systemadministratoren. Men husk, at fejlsøgningen ikke skal negligeres. Springes fejlsøgningen og dermed diagnostiseringen over, fejler maskinen ofte kort efter igen.

### 3.1.2 Overvågning

Ressourceudnyttelse er et vigtigt aspekt i overvågning. Ved ressourceovervågning forstås bl.a. inspektion af lagerplads som RAM, harddiske, video, streamer eller optiske diske, men også performance hører herunder.

Performance er en finere måde at anskue sit materiel på, men ikke mindre væsentlig, snarre tværtimod. For det er ofte igennem en performance undersøgelse man for et reelt billede af ressourceudnyttelsen. Er det fx nødvendigt med en ekstra CPU, BAIO eller harddisk for at sikre en stabil drift. Systemadministratoren skal medvirke til at få øje på flaskehalsproblemerne, inden de bliver synlige for brugerne på systemet. Performanceanalyser kan naturligvis bestilles.

At overvåge roddisken er ingen spøg. Løber roddisken fuld, vil maskinen ikke kunne fortsætte.

### 3.1.3 Support

En systemadminstrators opgaver omfatter som regel også støtte til de øvrige brugere på systemet. Afhængig af hvordan EDB- funktionen er defineret, kan støtten omfatte alt inden for kontorprogrammer eller hjælp til udviklere.

Vejledning af systemets brugere indbefatter også at holde brugerne underrettet om de vedligeholdelses- og driftrutiner, der foregår på systemet.

Skal brugerne kunne bruge systemadministratoren som en slags "hotline" funktion til løsning af EDB tekniske problemer. Det er et meget diskuteret emne uden entydigt svar, ligesom krav til brugernes EDB færdigheder, samt hvad man som bruger selv skal sætte sig ind i, ofte er et uafklaret spørgsmål.

## 3.2 Sikkerhedsmæssige opgaver og ansvar

Dette område er særdeles vigtigt for systemadministratoren. Går noget galt, skal backup'en være taget, ligesom logbog og generel sikkerhed på systemet skal være i orden.

### 3.2.1 Backup

Det er vigtigt at planlægge og udføre en god backup rutine, således at data på systemet til enhver tid kan reetableres i tilfælde af maskinelle fejl eller brugerfejl. Det er muligt at planlægge backup-'en således at den kører automatisk om natten. I det daglige arbejde vil backup'en derfor ikke være til gene for systemets brugere.

Imidlertid kan reetablering af data være tidskrævende. Derfor bør man i sin backup planlægning tage hensyn til dette aspekt, hvis datatab sker ofte.

UNIX Backup (beskrevet i Supermax Enterprise Server ABC), skal ligeledes tages ved opgradering af operativsystem eller ved væsentlige ændringer af kernen.

Og husk at backup medier slides. Bruger man videobånd til backup, skal de udskiftes med jævne mellemrum. Lad evt. det ældste bånd, som har været brugt i den daglige backup, blive til måneds backup båndet.

### 3.2.2 Logbog

Logbogen er et redskab til brug ved lagring af betydningsfuld viden om EDB systemet. Derfor er det vigtigt at føre logbogen korrekt. Man kan komme ud for, at logbogen er eneste mulighed for at reetablere systemets opbygning (data kan naturligvis kun genskabes ved backup).

På et system med SVR4.2 operativsystem er det en god ide at føre log over:

* Software på systemet
* Hardware på systemet
* Backup herunder UNIX backup
* Tekniker besøg
* Problemer
* Scripts
* Diskkonfiguration (VTOC)
* Maskinkonfiguration (prtconf)
* Spejlede diske

Er virksomheden certificeret, er alt mht. logbog lagt fast.

### 3.2.3 Overvågning

Overvågning blev bl.a. omtalt under de driftstekniske opgaver og ansvar. Men overvågning set ud fra et sikkerhedsmæssigt aspekt kræver væsentligt mere opmærksomhed og spidsfindighed.

Brugerne bør skifte password samt holde deres password hemmeligt. Tager brugerne disketter med til og fra arbejde, er der risiko for virus. Rettighederne på vigtige systemfiler skal checkes, og ubudne gæster på systemet skal findes, inden de laver ravage.

Man kan købe forskellige programmer til at checke for virus og ubud-

ne gæster. De er udmærkede men ikke nødvendigvis tilstrækkelige. Systemadministratoren skal altid være på vakt overfor de udefrakommende farer, eftersom de er de potentielt mest destruktive.

## 3.3 Juridisk ansvar

Det juridiske ansvar tager man ofte lidt for let på, hvis man i det hele taget tager det i betragtning. Ikke desto mindre kan mange forhold mht. software være på kant med loven. Som systemadministrator skal man som minimum være klar over, om de programmer, som brugere i firmaet anvender, også er købt og betalt.

### 3.3.1 Licenser

Software, med mindre det er Public Domain, må kun bruges, når man er i besiddelse af en licens til produktet. Og ofte skal man også have licens til et vist antal brugere. Nu om dage kan mange programmer på UNIX installationer slet ikke køre, med mindre man har betalt for produktet, eftersom softwaren selv checker om den må bruges på den aktuelle maskine.

### 3.3.2 Datatab

Regnskab, kontrakter og andre væsentlige firmadokumenter, karakterblade for skoler etc., skal gemmes i enten lovbestemte tidsintervaller, eller også er det bare en god ide at gemme på dem.

Gemmes data ikke, kan det få alvorlige juridiske konsekvenser, som ofte munder ud i økonomiske påkrav.

### 3.3.3 3. parts ansvar

Med hensyn til den før omtalte sikkerhed på systemet, hvor man skulle overvåge ubudne gæster på sit system, bør man også gøre sig klart, at er man fx koblet op til andre virksomheder i et større netværk, skal man sikre sig, at både egne brugere og ubudne gæster ikke må komme videre.

## 3.4 Økonomisk ansvar

En systemadministrator kan udmærket være pålagt et økonomisk ansvar, selvom det ikke er det mest almindelige. Imidlertid er EDB anskaffelser dyre, og således har systemadministratoren i det daglige ansvaret for anselige summer.

### 3.4.1 Indkøb

Systemadministratoren er ofte den, som ved, hvad der rører sig på det teknologiske markede. Derfor er det naturligt (men ikke altid tilfældet), at systemadministratoren inddrages i beslutningsprocessen vedrørende indkøb af nyt maskinel eller software. Det økonomiske ansvar kan således både være indirekte, fordi man har deltaget i beslutningsprocessen, eller direkte, hvis man har deltaget i hele anskaffelsesprocessen.

### 3.4.2 Vedligeholdelseskonto

I det daglige vedligehold skal der også rutinemæssigt anskaffes nyt udstyr eller reservedele til eksisterende udstyr. Og kontoen skulle gerne holde året ud, så budgettet ikke springes.

Husk, at forebyggende vedligelholdelse er bedre end udbedrende vedligeholdelse.

# 4 Shell

Når en bruger er i kontakt med operativsystemet, sker det igennem en shell. En shell er en kommandofortolker, der udfører kommandoer, som afgives fra en terminal eller et program.

Navnet shell - eller skal på dansk, hentyder til, at shell danner en skal uden om UNIX operativsystemets kerne, dvs. shell skjuler de komplekse operationer, som brugeren ikke behøver at kende til for at bruge systemet.

Efterhånden som UNIX er blevet udviklet, er der lavet mange forskellige shells. For overskuelighedens skyld er de vigtigste shells og deres funktionalitet stillet op i skemaet herunder.

Deres navne er **Bourne** shell eller **shell** (sh), **Restricted** shell (rsh), **DDE** shell (dsh), **C** shell (csh), **Korn** shell (ksh), **TC** shell (tsch) og **Bourne Again** shell (bash).

På **SVR4.2** findes **dsh** ikke. Standardshellen på **SVR4.2** er en **ksh**, men der følger også en **sh** samt en **csh** med. En **restricted** shell ligger under kataloget **/usr/lib/rsh**, og må ikke forveksles med **/usr/-bin/rsh**, som er en remote shell. Imidlertid er **/usr/bin** standard shell kataloget for **SVR4.2**.

| | sh | rsh | dsh | csh | ksh | tcsh | bash |
|---|---|---|---|---|---|---|---|
| kommando historie | nej | nej | ja | ja | ja | ja | ja |
| alias facilitet | nej | nej | nej | ja | ja | ja | ja |
| shell scripts | ja | nej | ja | ja | ja | ja | ja |
| filnavn fuldendelse | nej | nej | nej | ja* | ja* | ja | ja |
| kommando editering | nej | nej | ja | nej | ja* | ja | ja |
| job kontrol | nej | nej | nej | ja | ja | ja | ja |

* Betyder: at det ikke er standardopsætningen for shell'en, men at det understøttes.

Alle shells, men primært **Korn** shell og **Bourne Again** shell, vil blive omtalt senere. Først vil en gennemgang af de vigtigste aspekter ved brug af en shell blive belyst.

## 4.1 Overordnet brug af shell

Alt afhængig af, hvorledes man som bruger er sat op, logger man ind i en shell og bliver der, eller også havner man i en menu eller et program. Starter man op i sidstnævnte, skal man finde en udgang til shellen. Starter man op i en shell som almindelig bruger, vil man normalt se følgende prompt:

**$**

Prompten kan se forskellig ud alt afhængig af den shell man starter op i. Shell variablen **PS1** styre promptens udseende.

Promptens tilsynekomst betyder, at operativsystemet er klar til at modtage kommandoer. Hvis man logger ind som superbruger (root) på konsollen, eller udfører **su** kommandoen for at blive superbruger, vil prompten være:

**#**

Her afhænger prompten igen af den benyttede shell, men normalt afsluttes en superbruger prompt med # tegnet.

Man bør kun arbejde som superbruger, når man skal udføre opgaver som systemadministrator på maskinen

Udlogning fra shell sker med UNIX kommandoen **exit** eller med **^d** (<ctrl> + d).

Hvilken shell man arbejder i, kan man ofte se foran sin prompt. Lad os fx tage **bash**, som vil starte op således:

**bash$**

Udskriver man indholdet af shell variablen **PS1**, når man bruger en bash, vil det være:

```
$ echo $PS1
bash$
```

Man kan også finde ud af hvilken shell man benytter ved at udskrive indholdet af variablen **SHELL**:

```
$ echo $SHELL
/usr/bin/sh
```

som afslører, at den benyttede shell er en **Bourne** shell. Imidlertid kan denne metode fejle, eftersom kun den shell man logger ind i vises. Skifter man shell undervejs uden at ændre **SHELL** variablen manuelt, kan man bruge UNIX kommandoen **ps** til at afsløre shellen med:

```
bash$ ps

  PID  CLS PRI TTY      TIME COMD
17194   TS  70 console  0:01 bash
17221   TS  48 console  0:01 ps
```

Her kunne man allerede ved prompten have gættet på en **bash**, men også UNIX kommandoen **ps** giver klar besked. Processen med **PID** 17194 er en **bash**.

## 4.2 Tastekonventioner

For at ingen forvirring opstår, vil her kort blive gennemgået, hvorledes de forskellige taster kan opfattes. Her er nogle eksempler med tastetryk med forskellig notation:

| ALM. | EMACS | Betydning |
|---|---|---|
| <ESC>F | M-F | tast escape efterfulgt af f |
| <TAB> | | tast tabulator |
| <CTRL> + d | ^D | tast control (og hold den nedtrykket) efterfulgt af d |

## 4.3 Lidt shell historie

Her er en kort gennemgang af forskellige shells gennem tiderne. Nogle kan stadig anvendes, men der er kommet bedre til siden.

* **sh** **Shell** eller **Bourne** shell er den oprindelige UNIX shell skrevet af **Steve Bourne** for Bell Labs, AT&T. Den er tilgængelig på alle UNIX styresystemer. Shellen har ikke interaktive faciliteter som mere moderne shells som fx **C** eller **Korn** shell er i besiddelse af. Af den grund er det svært at skrive shell scripts i denne shell.

* **rsh** **Restricted** shell er en begrænset **Bourne** shell, hvor bl.a. brugerens muligheder for frit at bevæge sig rundt i filsystemet er indskrænket.

* **dsh** **DDE**s egen shell på **SMOS** operativsystemet, som er et UNIX baseret operativsystem med egen shell. **DDE** shell er basalt set en **Bourne** shell med historie mekanisme. **DDE** shell findes ikke på **SMES** operativsystemet dvs. **SVR4.2.**

* **csh** **C** shell blev udviklet i Californien på Universitetet i Berkley. Som navnet antyder, giver det adgang til et **C** lignende shell scripts sprog.

* **tcsh** **TC** shell er kort sagt en public domain **C** shell med mulighed for **EMACS** editering.

* **ksh** **Korn** shell er den officielle shell på **SVR4.2.** Den vil blive gennemgået mere detaljeret senere i dette materiale.

* **bash** **Bourne Again** shell er den umiddelbart mest tilgængelige shell på **SVR4.2** set fra brugerens side. En **public domain** shell fra **Free Software Foundation,** som naturligvis ikke bliver sammen med **SVR4.2.** Man må selv hente den hjem over nettet. Den vil blive gennemgået mere detaljeret senere i dette materiale.

## 4.4 Korn shell

**Korn** shell er skrevet af **David Korn** fra Bell Labs., AT&T. Den er nu blevet til standard shell i stedet for **sh**, som dog stadig leveres med UNIX operativsystemet. Den har samme funktionalitet som **C** og **TC** shells, samtidig med at den har et shell script sprog magen til det i **Bourne** shell.

**Korn** shell kan startes op uden kommando editering og filnavn fuldendelse, men man snyder sig selv for en masse nyttige faciliteter. Man kan vælge mellem **EMACS** og **vi** som hjælpeværktøj. Det betyder, at den funktionalitet som man er vant til fra editorene, også kan anvendes på kommandolinien i shellen. Man starter **Korn** shell op med editorfaciliteter på følgende måde:

```
ksh -o emacs
```

Option **-o** giver med andre ord en lov til at vælge editor. Har man allerede startet **Korn** shell op, kan man vælge editor på anden måde:

```
set -o vi
```

Her er **vi** valgt som editor efter at **Korn** shellen er startet.

### 4.4.1 Kommando historie

Ønsker man at se de gemte kommandoer, kan listen fås frem med kommandoen:

```
$ history
```

De vises nummereret i den rækkefølge, som de er blevet anvendt. Listen kan standard rumme de sidste 128 kommandoer. Ønsker man kun at gemme de sidste halvtreds (50) udførte kommandoer, skal variablen **HISTSIZE** tildeles værdien 50:

```
$ HISTSIZE=50;export HISTSIZE
```

Når man logger ud fra en **Korn** shell gemmes indholdet af kommando historie listen automatisk i en fil i ens hjemmekatalog med navnet

**.sh_history.** Når man logger ind igen, placeres indholdet af filen i kommando historie listen, og man kan således genbruge sine sidst udførte kommandoer fra forrige indlogning.

Den sidste kommando udføres igen på følgende måde:

```
$ pwd
/bruger/hbh
$ r
pwd
/bruger/hbh
```

Her blev UNIX kommandoen **pwd** brugt og derefter genbrugt.

Genbruges en kommando fra kommando historie listen, kan det gøres ved at bruge nummeret fra listen:

```
$ r 15
date
Thu Feb 15 12:46:36 MET 1996
```

Her lå UNIX kommandoen som nummer 15 i kommando historie listen og blev derfor kaldt.

Har man for lidt siden brugt UNIX kommandoen **grep**, og ønsker at bruge den på præcis samme måde igen, kan man taste:

```
$ r grep
grep -n unix0 /etc/passwd
```

hvorefter **grep** kommandoen findes og udføres.

### 4.4.2 Alias facilitet

I **Korn** shell laver man et **alias** ved at skrive:

```
$ del='rm -i'
del brev.txt
rm: remove brev.txt? y
```

Her blev aliaset del tildelt UNIX kommandoen **rm -i**, som interaktivt sletter sine argumenter. Når et alias indeholder blanktegn, skal man huske at indramme UNIX kommandoen med '' eller "". I dette ex. er der et mellemrum (mellem m og -) i '**rm -i**'.

Aliaset kan lægges ned i en fil ved navn **.kshrc**, som udføres lige efter indlogning.

Hvilke alias som er oprettet, kan man se med kommandoen:

```
$ alias
```

Ønsker man oplysninger om et bestemt alias kan man taste:

```
$ alias del
alias del='rm -i'
```

Dette vil give oplysninger om kommandoen del (hvis den findes).

### 4.4.3 Filnavn fuldendelse

Hvis denne facilitet skal kunne virke, skal **EDITOR** variablen sættes til enten:

```
EDITOR=emacs eller EDITOR=vi
```

skriver man fx

```
$ cat h ...
```

hvorefter man ønsker at fuldende navnet (filen hedder haha), kan man gøre det på følgende måde:

```
emacs       vi

<ESC><ESC>  \
```

Eksempel:

```
$ cat h<ESC><ESC>
```

hvorefter der pludselig står:

```
$ cat haha
```

Mulige valg kan man kun få frem i **EMACS** mode. Da må man taste <ESC> efterfulgt af =:

```
$ cat h<ESC>=
1) haha
2) haha_old
3) hahaha.old
```

Her ligger der tre filnavne i kataloget, som alle begynder med h.

### 4.4.4 Kommando editering

Med **EMACS** eller vi emulering kan man ligeledes hente UNIX kommandoer frem fra historie listen samt editere dem uden at udføre kommandoen øjeblikkeligt.

| | EMACS | vi |
|---|---|---|
| Bevæg cursoren til venstre | ^B | <ESC>h |
| Bevæg cursoren til højre | ^F | <ESC>l |
| Bevæg cursor til næste ord | M-F | <ESC>w |
| Bevæg cursor til forrige ord | M-B | <ESC>b |
| Bevæg cursor til slut på ord | | <ESC>e |
| Bevæg cursor til begyndelse af linie | ^A | <ESC>0 |
| Bevæg cursor til slut på linie | ^E | <ESC>$ |
| Slet til venstre for cursor | <BS-DEL> | <ESC>dh |
| Slet tegn under cursor | ^D | <ESC>x |
| Slet til slut af ord | M-D | <ESC>dw |
| Slet til begyndelsen af et ord | M-^H | <ESC>db |
| Slet fra cursor til slut på linie | ^K | <ESC>D |
| Slet en hel linie | ^A^K | <ESC>dd |
| Fortryd | ^Y | U |
| Se tidligere udførte kommandoer | ^P | <ESC>k |

| | | |
|---|---|---|
| Se efterfølgende udførte kommandoer | ^N | <ESC>j |
| Find kommando vha. streng, fremefter | ^Rstreng | <ESC>/streng |
| Find kommando vha. streng, bagud | | <ESC>?streng |

Brug af kommandoen **fc** (fix command) er ligeledes en editeringsmulighed. Når editoren forlades, udføres kommandoen eller scriptet. Her er vist 2 eksempler:

Eksempel 1: Vi ønsker at editere en UNIX kommandoen med EMACS faciliteter fra historie listen. Kommandoen har nummer 32.

```
$ fc -e emacs 32
```

Eksempel 2: Vi ønsker at editerer et shell script bestående af linierne fra 22 til 34 med vi faciliteter.

```
$ fc -e vi 22 34
```

Shell programmering over flere linier uden brug af hverken shell scripts eller **fc** er naturligvis også muligt.

### 4.4.5 Job kontrol

Vi kender alle irritationsmomentet ved at glemme at sætte en proces i baggrunden. Har man kun en shell åben, er man umiddelbart på den, med mindre man arbejder i en shell med job kontrol.

Med **Korn** shell findes den låste shell ikke. Har man startet et job, uden at sætte det til at køre i baggrunden, kan man suspendere jobbet med **^z** (<ctrl>+z). Et typisk forløb kunne se således ud:

```
$ sleep 100    (Tast nu ^z)
[1] + Stopped (user)            sleep 100
$ bg
[1]   sleep 100&
```

Skriver man nu **jobs**, fås følgende oplysning:

```
$ jobs
[1] +  Running                 sleep 100
```

som oplyser om, at processen stadig er aktiv, men nu kører i baggrunden. Bruger man **jobs** med option **-l** fås også oplysning om proces id (PID), ligesom ved brug af UNIX kommandoen **ps**.

Ønsker man at få en proces til at blive en forgrundsproces, kan man bruge **fg** kommandoen:

```
$ jobs -l
[1] + 10472      Running          sleep 100
$ fg 10472
sleep 100
```

Shellen er nu låst, fordi processen fra at være en baggrundsproces er blevet gjort til en forgrundsproces med **fg**.

Med UNIX kommandoen **kill** kan man altid slå en baggrundsproces ihjel.

## 4.5 Bourne Again shell

**Bourne Again** shell er en public domain shell fra **Free Software Foundation** under deres **GNU** (GNU not UN*X) project. **Bourne Again** shell er et "evigt" igangværende forsøg på at lave en fuld implementation af IEEE Posix Shell and Tools specifikationen. Derudover er der kun mulighed for at emulere **EMACS** kommandoer.

**bash** har alle de interaktive faciliteter som **Korn** shell (ksh) og **C** shell (csh) besidder. Shell scripts skrevet i **bash** er kompatible med **Bourne** shell (sh).

Funktionaliteten af **Bourne Again** shell (bash), som er forskellig fra **Korn** shell, vil i det følgende blive gennemgået med nogle eksempler.

### 4.5.1 Kommando historie

Ønsker man at se de gemte kommandoer, kan listen fås frem med kommandoen:

```
$ history
```

De vises nummereret i den rækkefølge, som de er blevet anvendt. Listen kan standard rumme de sidste 500 kommandoer. Ønsker man kun at gemme de sidste halvtreds (50) udførte kommandoer, skal variablen **HISTSIZE** tildeles værdien 50:

```
$ HISTSIZE=50;export HISTSIZE
```

Når man logger ud fra en **Bourne Again** shell gemmes indholdet af kommando historie listen automatisk i en fil i ens hjemmekatalog med navnet **.bash_history**. Når man logger ind igen, placeres indholdet af filen i kommando historie listen, og man kan således genbruge sine sidst udførte kommandoer fra forrige indlogning.

I variablen **HISTFILESIZE** kan man angive, hvor mange kommandoer som skal gemmes i **.bash_history**.

### 4.5.2 Alias facilitet

Ønsker man at knytte en UNIX kommando til et mere sigende navn, eller ønsker man at skyde genvej og slippe for en masse indtastningsarbejde for at få udført en kommando med mange optioner, kan man bruge alias faciliteterne.

Lad os fx tage UNIX kommandoen **ls** med optionerne **-ali**. Et alias til kommandoen **ls -ali** kunne blive dir, som skal erklæres på følgende måde i en fil ved navn **.bashrc**, som skal ligge i ens hjemmekatalog:

```
alias dir="ls -ali"
```

For at kunne bruge sin **.bashrc** fil øjeblikkeligt ved ændringer, udfør da:

```
$ . .bashrc
```

### 4.5.3 Filnavn fuldendelse

I **Bourne Again** shell kan man bruge <TAB> i stedet for <ESC><ESC> til at fuldende fil- eller katalognavne.

### 4.5.4 Kommando editering

Piltasterne kan i **Bourne Again** shell benyttes i stedet for **EMACS** funktionerne.

| bash | EMACS |
|---|---|
| PIL OP | ^P |
| PIL NED | ^N |
| PIL TIL HØJRE | ^F |
| PIL TIL VENSTRE | ^B |

## 4.6 Værd at vide

I dette afsnit står der lidt af hvert, bl.a. dokumenterede fejl i bash shell'en.

### 4.6.1 Fejl i bash shell

Følgende fejl er konstateret:

| | |
|---|---|
| ulimit | håndteres forkert |
| boot | går i stå mht. oracle (ikke undersøgt i detalje) |
| enable | printer kan ikke enables uden fuld stiangivelse til UNIX kommando |
| unix_backup | fejler |

Derfor er benyttelsen af **bash** IKKE anbefalelsesværdig som konfigurationsshell og slet ikke som eneste shell ved systemadministration.

### 4.6.2 bash prompt

**Bourne Again** shell prompten kan konfigureres med avancerede faciliteter. Her er et lille eksempel, som kan placeres i **~/.profile**.

```
PS1=" users on \h:\w \d \t \nCommand nr. \#(\!) \s_\u \$ "
PROMPT_COMMAND="echo -n `who | wc -l`"
export PS1 PROMPT_COMMAND
```

En typisk prompt vil herefter se således ud:

```
6 users on alex:~ Mon Apr  1 12:52:11
Command nr. 3(503) bash_hbh $
```

Prompten vil nu vise antallet af brugere på maskinen, efterfulgt af dato og klokkeslet. Herefter følger et nyt linie skift, hvorefter kommando nummeret efter sidste indlogning vises samt nummeret i kommandolisten (**~/.bash_history**). Til slut vises shell navnet, brugeren samt enten **$** eller **#** tegnet, alt afhængig af brugerens aktuelle **UID**.

Bemærk, at variablen **PROMPT_COMMAND** sættes hver gang at shellen aktiveres.

### 4.6.3 .inputrc (bash)

**Bourne Again** shell kan konfigureres med en fil ved navn **.inputrc**. Ved hjælp af filen, som skal ligge i hjemmekataloget, kan tastaturet sættes op.

### 4.6.4 .login og .logout (C og TC)

Filerne **.login** og **.logout** bruges af C og TC shells til konfiguration ved henholdsvis ind- og udlogning.

### 4.6.5 rc, es og zsh

Der findes også en **rc** shell, skrevet i to tempi af **Tom Duff** og **Byron Rakitzis**, som er blevet videreudviklet til en es shell. Begge er nyskrevne shells, som ikke bygger på de gamle. Den nyeste shell, **zsh** shell, skrevet af **Paul Falstad**, er den mest avancerede af alle. Men fælles for de nye shells er, at de endnu ikke er ret udbredte og gennemprøvede.

Ønsker man at fordybe sig i brugen af shells, er det tilrådeligt at finde sig et godt søgeværktøj og bruge internettet.

Et skift af shell er i grunden en alvorlig sag. Shellen er den proces, som holder en i live på maskinen. Derfor skal man aldrig smide sin gamle shell ud, før man er rimelig sikker i sin sag. Desuden skal en shell compileres ned mod kernen. Er man ikke klar over, hvorledes kode compileres, skal man skaffe sig kvalificeret hjælp, inden man begynder.

## 4.7 Free Software Foundation

**Free Software Foundation** har mange gode produkter på hylden, som gratis kan erhverves. Man kunne foranlediges til at tro, at god software skal koste penge, men har man prøvet nogle **GNU** produkter, ændrer man hurtigt den holdning.

**GNU** produkter udvikles, sammen med dem som bruger produkterne, gennem dialog på internettet. Og mange skriver fx macroer til produktet **TeX**, fordi de synes om produktet og gerne vil bidrage. At internettet benyttes betyder bl.a. også, at man som bruger kan få mange oplysninger om produkterne, hvis man kører fast og har brug for hjælp.

Af gode produkter kan nævnes: **EMACS** (editor), **Bourne Again** shell (shell), **TeX** (tekstbehandler, skrevet af Donald Knuth, med macroen LaTeX, skrevet af Leslie Lamport) og **GNUplot** (grafisk option til TeX).

Internetadressen til **Free Software Foundation** er:

**prep.ai.mit.edu**

# 48 Opgave

*Formål: -at sætte sin shell op og bruge faciliteterne*

1. **Find ud af vilken shell du bruger.**
2. **Udfør nogle kommandoer. Log ud.**
3. **Log ind igen. Har shellen husket kommandoerne.**
4. **Opret et alias (fx ls -ali).**
5. **Bruge filnavn fuldendelse faciliteten.**
6. **Lav kommando editering. Bevæg dig op og ned ad historie listen, editer kommandolinier og søg efter tidligere udførte kommandoer.**
7. **Start en 'sleep 1000' som forgrundsproces.**
8. **Suspender jobbet 'sleep 1000' og sæt det i baggrunden.**
9. **Sæt jobbet 'sleep 1000' i forgrunden igen.**
10. **Slå jobbet 'sleep 1000' ihjel.**
11. **Lav om på din prompt PS1.**

# 5 Maskinkonfiguration

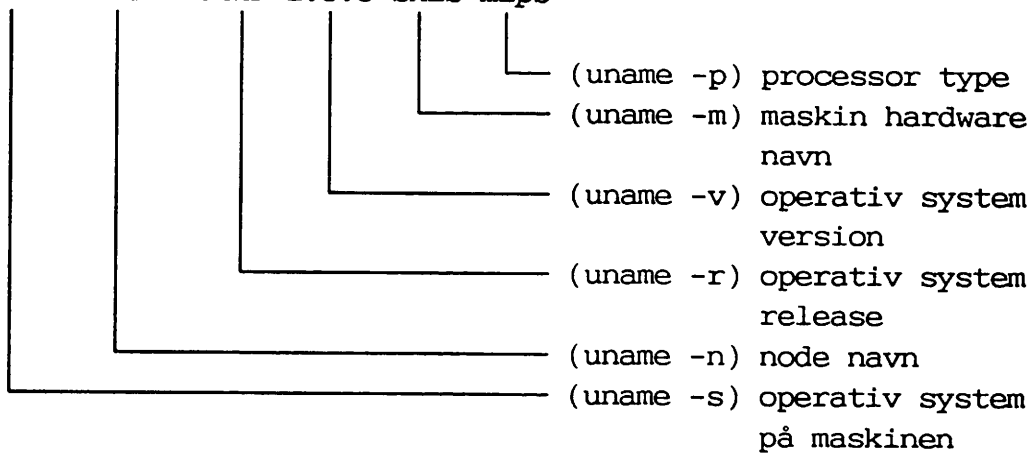
Når man logger ind på en UNIX maskine og ønsker at få et overblik over operativsystem og hardware, er der forskellige kommandoer og databaser, som kan hjælpe en.

Nogle værktøjer giver meget specifik viden om maskinen, andre giver mere overordnet information. **Sysadm** kan til en vis grænse give de samme oplysninger.

## 5.1 Operativsystem

Når man stifter bekendskab med en UNIX maskine, kan man få en overordnet beskrivelse af operativsystemet med '**sysadm machine configuration system**' eller med UNIX kommandoen **/sbin/uname**:

```
$ /sbin/uname -a
UNIX SV alex 4.2MP 1.3.3 SMES mips
```




Bemærk, at **uname -p** er det samme som at give UNIX kommandoen **/usr/ucb/mach**.

Optionen **-a** (all) giver fuld information. De enkelte felter kan også fås frem med andre optioner. Optionerne er vist i parentes med beskrivelsen af feltet umiddelbart efter.

UNIX SVR4.2 operativsystemet anvender katalogerne:

**/**
**/usr**
**/var**
**/stand**

Det er også disse kataloger, som bliver gemt ved en **UNIX Backup**, som laves med UNIX kommandoen **/usr/dde/bin/unix_backup**. Se herom i **"Supermax Enterprise Server ABC"**.

## 5.2 Hardware

En overordnet beskrivelse af hardwaren er det også muligt at generere. Kommandoen **/etc/machid** giver information om maskinens hardware:

```
$ /etc/machid all
bus     spc3
inst    2316
serial  29
bpname  BPL304
fcn                                                    150
cpu     0  0 CPU300                                  7 149
memory  0  0 8192   2000 8192 MEM301-1               8 147
baio    0 -1 ethernet,scsi0,scsi1,uart BAI0302-0 56  156
baio    0  0 service
```

hardware keyword
(kan bruges som argument til machid)

Backplane type — BAIO konfiguration
Backplane serie nummer — Memory konfiguration
Installations nummer — CPU konfiguration
Bus typen (spc2 eller spc3) — Backplane fcn nr.

Argumentet all til kommandoen **/etc/machid** giver fuld information. Men 'hardware keywordet' kan også bruges som argument, hvis der fx kun ønskes specifik information om et bestemt kort.
Hvis maskinen besidder flere cpu-,memory- eller baiokort, skal modulet angives (for baio også submodul).

## 5.3 System konfiguration

UNIX System V Release 4.2 er et "selvkonfigurerende" system. Dvs. at systemet selv opbygger de nødvendige referencer og vælger de rigtige drivere til ydre enheder som terminaler, harddiske, streamere, CD-ROM drev og floppydiske.

En ydre enhed er i UNIX SVR4.2 kendt på to måder:

- Et logisk navn oprettet i Device Databasen (DDB)

- En specialfil under kataloget **/dev** (eller i et af dets underkataloger), som knytter den pågældende enhed til den relevante driver.

Med **'sysadm machine configuration summary'** eller med kommandoen **/usr/sbin/prtconf** kan man få en enkel oversigt over maskinens hardware konfiguration.

## 5.4 Device Databasen (DDB)

Når de forskellige enheder installeres på maskinen oprettes de med et logisk navn i **Device Databasen.** Dette logiske navn kan bruges til at referere til enheden istedet for det specialfilnavn, der ellers kan anvendes.

**DDB's** oplysninger opbevares i en fil, der hedder **/etc/device.tab.** Denne må man ikke rette i direkte, men skal istedet anvende kommandoen **/usr/bin/putdev**, hvis man vil rette manuelt i **DDB.**

Til et navn i **DDB** kan knyttes en række attributter. F.eks. navnet på specialfilen for enheden, navnet på alternative specialfiler, eller forskellige attributter, der kan anvendes af systemadministrative programmer til at få nærmere information om enheden.

En oversigt over enheder kendt i **DDB** fås med kommandoen **/usr/bin/-getdev:**

```
$ getdev
arp
```

```
console
ctape1
disk1
diskette1
dpart11
dpart12
dpart13
.
.
.
udp
```

For den enkelte enhed kan man se, hvilke attributter der er sat med kommandoen **/usr/bin/devattr**, som udover **DDB** gør brug af filen **/etc/-security/ddb/ddb dsfmap:**

```
$ devattr -v ctape1
alias='ctape1'
bklib='SCSI'
bufsize='65536'
cdevice='/dev/rmt/c1t6d0s0'
cdevlist='/dev/erct0,/dev/nrct0,/dev/rct0,/dev/rmt/c0s0,/de
v/rmt/c0s0n,/dev/rmt/c0s0nr,/dev/rmt/c0s0r,/dev/rmt/c1t6d0s
0n,/dev/rmt/c1t6d0s0nr,/dev/rmt/c1t6d0s0r,/dev/rmt/ctape1,/
dev/rmt/ntape1,/dev/xct0'
copy='/bin/dd if=CDEVICE1 of=CDEVICE2 bs=10240'
desc='Tape Drive 1'
display='true'
erasecmd='/usr/lib/tape/tapecntl -e /dev/rmt/c1t6d0s0n'
norewind='/dev/rmt/c1t6d0s0n'
pdimkdtab='true'
removable='true'
retension='/usr/lib/tape/tapecntl -t /dev/rmt/c1t6d0s0n'
rewind='/usr/lib/tape/tapecntl -w /dev/rmt/c1t6d0s0n'
rewindcmd='/usr/lib/tape/tapecntl -w /dev/rmt/c1t6d0s0n'
scsi='true'
type='qtape'
volume='cartridge'
```

## 5.5 Specialfiler til ydre enheder

I UNIX har man adgang til alle ydre enheder vha. specialfiler. Dvs. hvis et program skal skrive til (hhv. læse fra) en ydre enhed angives specialfilen for den relevante enhed som output (hhv. input) fil. Der findes to forskellige typer specialfiler for ydre enheder, der anvendes afhængig af, hvordan "device driveren" skal behandle enheden. Det to typer er hhv. "block special" og "character special". Hvilken type en given fil er ses i output fra en "**ls -l**" kommando. Det første tegn på linien vil ved ovennævnte filtyper være et "b" eller "c". Eksempel:

```
$ ls -l /dev/pts001
crw-rw-rw-    3 root     tty       35,  1 Feb 10 17:56 /dev/pts001
|
└────character special file
```

En "block special file" er en enhed hvortil læsning/skrivning sker blokvis. En "character special file" er en enhed hvortil læsning/-skrivning sker tegnvis.

### 5.5.1 Major/minor numre

En specialfil til en enhed har ingen "størrelse", men har istedet major/minor numre. Se nedenstående eksempel:

```
$ ls -l /dev/pts001
crw-rw-rw-    3 root     tty       35,  1 Feb 10 17:56 /dev/pts001
                                     |   |
                          major nummer┘   └──minor nummer
```

Major nummeret angiver hvilken "device driver" UNIX skal bruge til den pågældende enhed. Minor nummeret angiver hvilken enhed på den pågældende "device driver", der er tale om. Mao. kan man sige, at major/minor numrene er en slags hardware-adresse.

## 5.6 Kataloger med specialfiler til ydre enheder

Alle specialfiler til ydre enheder findes i kataloget **/dev** eller i dettes underkataloger. Hvilken type enhed der er tale om kan således identificeres ved specialfilens beliggenhed. Eksempler:

**/dev/dsk** Indeholder block special filer til harddiske, disk slices og disketter.

**/dev/rdsk** Indeholder character special filer til harddiske, disk slices og disketter.

**/dev/rmt** Indeholder character special filer til de forskellige typer af båndstationer (alm. streamer, video og DAT).

**/dev/term** Indeholder character special filer til terminaler tilsluttet via seriellse linier.

Opbygningen af selve filnavnet for den enkelte enhed siger også noget om, hvilken enhed det drejer sig om. Dette vil blive gennemgået senere i materialet, hvor det er relevant.

## 5.7 Moduler og submoduler

En Supermax Enterprise Server er opbygget af moduler eller kort, hvorpå der kan placeres submoduler. Det mest almindelige kabinet kan rumme 8 indstikskort, men også kabinetter med plads til 3 eller 5 kort fremstilles. De benævnes henholdsvis **8'eren**, **3'eren** og **5'eren**.

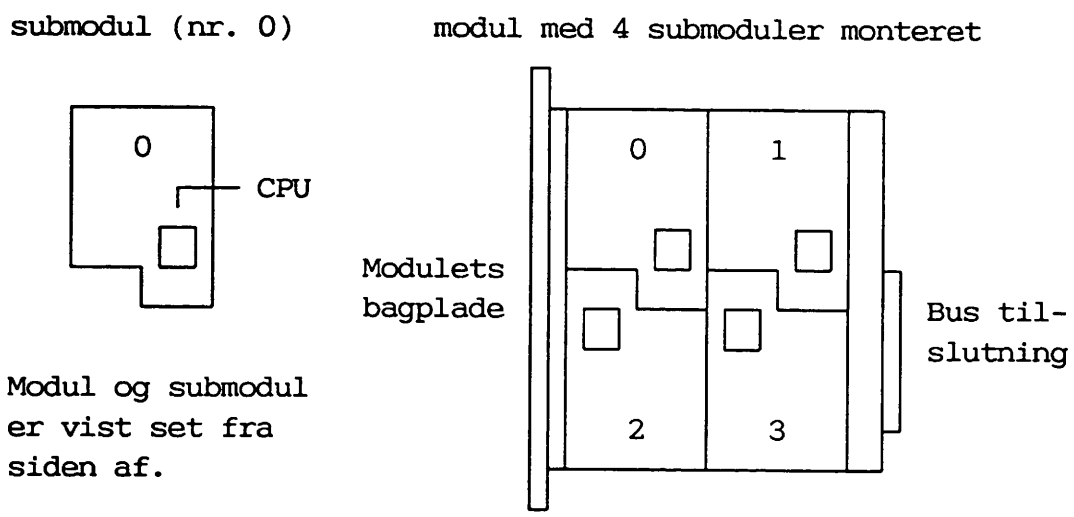
Man omtaler også kortenes placering i forhold til det backplane (bus), hvori kortet placeres. Så termen bus position eksisterer stadig, ligesom man taler om hvilken slot position eller slot nummer, kortet skal placeres i. Kortene slides ind i kabinettet fra bagsiden, efter bagbeklædningen er fjernet.

I **'Supermax Enterprise Server ABC'** kan man læse om moduler og submoduler, samt se flere gode illustrationer af Supermax Enterprise Serveren.

Printkortene er idag sammensat af 14 lag (eller flere), og et slide ind kort kan skiftes meget hurtigt.

## 5.7.1 CPU

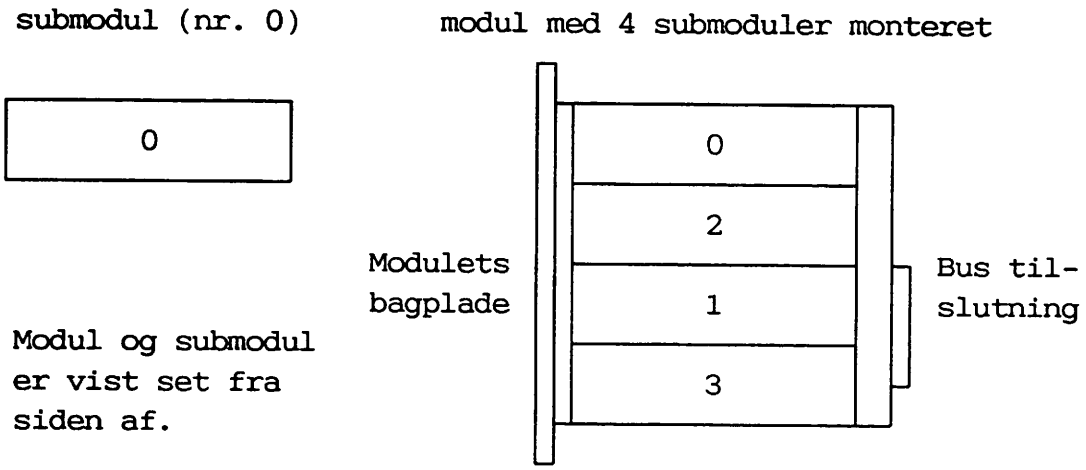
På CPU modulet kan der sidde mellem 2 og 4 submoduler, afhængig af CPU modulets udformning. Selve CPUen sidder på submodulet.




CPUen er af **MIPS 4400** typen, med en clockfrekvens på 200 MHz (de første blev leveret med 150 MHz), dvs. 200 millioner instruktioner pr. sekund. Layoutet er lavet af firmaet MIPS, et datterselskab af SGI (Silicon Graphics Inc.), men selve produktionen af CPU chipen foretages af mange fabrikanter.

## 5.7.2 Memory

Memory modulet kan indeholde op til 4 submoduler. Submodulerne skal monteres parvis, dvs. submodul 0 og 1 skal monteres samtidig, og submodul 2 og 3 skal monteres samtidig.




Submodulernes lagringskapacitet er på henholdsvis 32, 64 og 128Mb. Der kan således maksimalt sidde 512Mb RAM på et memory modul.

## 5.7.3 BAIO

Basic input/output eller kort BAIO er det modul, som klarer kommunikationen mellem maskinen og brugerne, og mellem ydre enheder. På BAIOen kan der sættes 3 submoduler. Submodulerne placeres ikke på selve modulet, men placeres ved siden af modulet i deres eget slot. Submodulerne er derfor kun en trediedel så høje som et modul, så de kan stables oven på hinanden.

En BAIO indeholder 2 SCSI porte, 2 serielle TPE RS232 forbindelser (bruges bl.a. til console på Master BAIO'en), samt 3 muligheder for nettilslutning: TPE (twisted pair cable dvs. PDS), AUI (Ethernet drop cable) og THIN (thin ether eller Cheapernet, dvs. BNC med coax). Derudover indeholder BAIOen altid et submodul, som er servicecomputeren.

## 5.8 Genopbygning af kernen

Efter installation af nyt hardware skal UNIX kernen genopbygges. Dette gøres med UNIX kommandoen **/etc/conf/bin/idbuild**.

# 59 Opgave

*Formål: -at undersøge en Supermax Enterprise Server*

1. **Brug kommandoen uname med forskellige optioner.**
2. **Brug kommandoen machid med forskellige argumenter.**
3. **Brug kommandoen prtconf.**
4. **Undesøg Device Databasen (DDB).**
5. **Undersøg major/minor numre for fx. de serielle linier i /dev og /dev/term kataloget.**
6. **Undersøg Supermax Enterprise Serveren. Find ud af, hvor de forskellige ting sidder set udefra.**

# 6 Harddisken

Harddisken er det sted, hvor data langtidsopbevares. En harddisk kan overordnet deles op vha. to logiske typer kaldet **"partitions"** og **"slices"**. Hver **partition** kan indeholde en bootsektor således, at man eksempelvis kan have een **partition**, der kan boote maskinen med et operativsystem (fx DOS), og en anden, der kan boote maskinen med et andet operativsystem (fx UNIX).

Almindeligvis vil det dog være sådan, at man på en UNIX maskine kun har en disk **partition** på sin harddisk (DDE standard opdeling af harddisk). En **partition** kan igen opdeles i flere logisk adskilte **slices.**

Formålet med at inddele sin disk i flere slices er dels at afgrænse nogle områder i mindre logisk sammenhørende enheder, og dels at have mulighed for at udlægge **slice'ene** til forskellige anvendelser. Det kan være praktisk af hensyn til overskueligheden på maskinen, men inddelingen foretages også af hensyn til backup. De mest almindelige **slices** er:

* **roddisken med filsystemer (/)**
* **vigtige slices med filsystem mounted på rodisken (/usr og /var)**
* **brugerdiske med filsystem (/home og/eller /bruger)**
* **bootdisk med boot-filsystem (/stand)**
* **swapdiske**
* **database diske**

Harddisken(e) inddeles i **slices** første gang UNIX installeres på maskinen, eller en harddisk inddeles i **slices**, når den installeres som ny harddisk på en maskine, der kører UNIX i forvejen.

Hvordan dette gøres kommer vi ikke yderligere ind på her, da kurset først og fremmest handler om administration af allerede konfigurerede maskiner. Det skal dog nævnes, at kommandoen **/sbin/diskadd** (som gør brug af UNIX kommandoerne **/usr/sbin/disksetup** og **/usr/sbin/mkfs**, samt kommandoen **fdisk** under DOS) anvendes til opdeling af nye harddiske.

Swapdisken skal som tommelfingerregel være dobbelt så stor (2x) som den RAM, der er installeret på maskinen. Og husk, at alle CPUerne deler den samme RAM.

## 6.1 Nummerering af diske

Diskene på Supermax Enterprise Serveren er "hardwired", hvilket betyder, at diskens adresse er positionsafhængig. Positionsafhængigheden fås ved at disken aflæser sin fysiske adresse gennem bussen på backplane't. Er diskene ikke hardwired, skal de "strappes" for at adressen bliver sat. Strapping foregår ved at man fysisk ved hjælp af "jumpere" sætter adressen på hardisken.

Hardwiring giver store fordele ved bl.a. **"hot pluged"** diske, hvor udskiftningen af en disk under drift skal foretages på under et minut.

Adresserne fastlægges overordnet vha. et **controller nummer** og et **target nummer. Controller nummeret** fås fra den lodrette position, **target nummeret** fås fra den vandrette position.

Imidlertid bliver de frontbetjente enheder såsom video, floppy, floptical, DAT, CD-ROM og streamer ikke hardwired. De strappes som sædvanligt. Derfor er deres adresse IKKE positionafhængig.

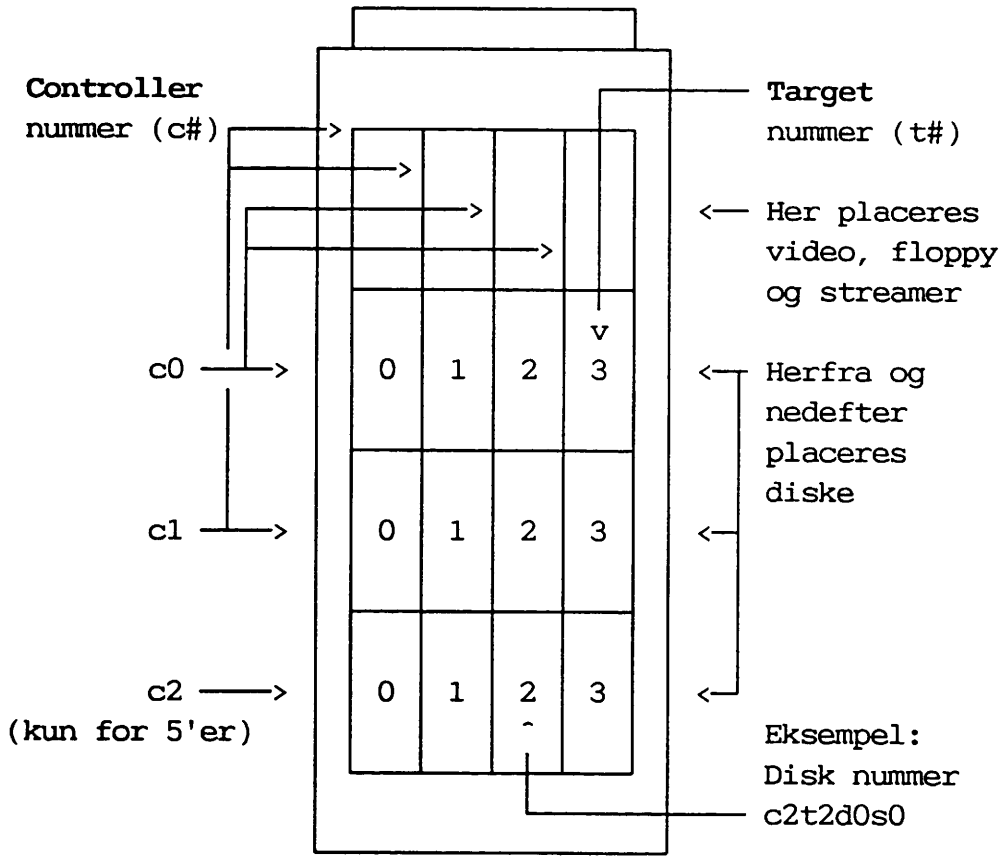
### 6.1.1 Supermax Enterprise 3'er og 5'er

Først gennemgåes en **3'er** og en **5'er**. Navne kommer af, hvor mange kort maskinerne kan rumme. En maskine er pr. definition en **3'er** (dvs. den kan rumme 3 kort), hvis den kun kan rumme en BAIO, og alle kortene sidder helt til venstre bagpå.

Det er en **5'er** (kan rumme 5 kort), hvis BAIO'en sidder helt til højre bagpå. En **5'er** kan rumme to BAIO'er. Derfor kan der også installeres harddiske på den nederste række på fronten (benævnt c2), hvilket ikke er muligt på **3'eren.**

En **3'er/5'er** er vist på nedenstående figur. Bemærk, at øverste række, dvs. de frontbetjente enheder, kan nummereres på følgende måde. Hvis det er for c1, kan target nummeret være 4,5 eller 7 (6 er reseveret). Hvis det er for c2, kan target nummeret være 4,5 eller 6

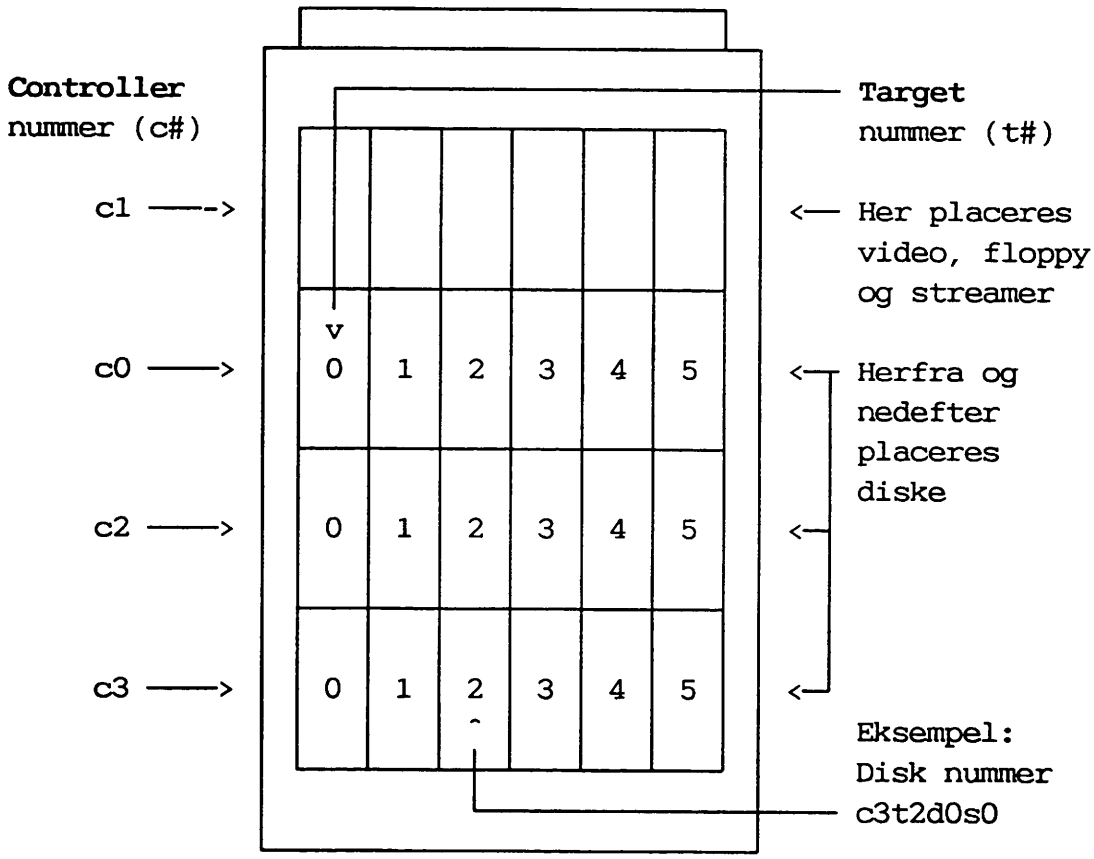
**Front på Supermax Enterprise Server 3'er og 5'er**



### 6.1.2 Supermax Enterprise 8'er

En **8'er** kan rumme otte kort, heraf navnet. For de frontbetjente enheder (controller 1 rækken) kan **target nummeret** være enten 0,1,2,3,4,5 eller 7 (6 er reseveret). For c0 og c2 kan **target nummeret** ligge mellem 0 og 6, og for c3 er det ligesom de frontbetjente enheder.
En **8'er** er vist på nedenstående figur.

**Front på Supermax Enterprise Server 8'er**



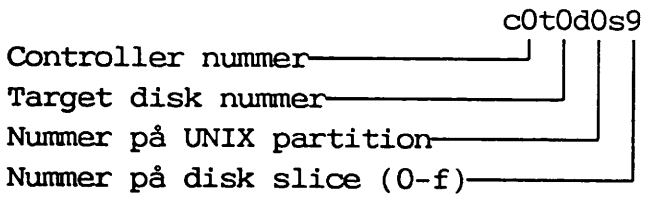
## 6.2 Specialfiler til disk slices

Der findes specialfiler til at referere til disk slices i to kataloger; **/dev/dsk** og **/dev/rdsk**. I **/dev/dsk** findes block special filer og **/dev/rdsk** findes character special filer til de samme slices.

Hvilken specialfil, der anvendes til at referere den enkelte slice afhænger af hvilken operation, der skal foretages på disken. Fælles for begge kataloger er dog navngivningen af de enkelte disk slices.

Navngivningen følger følgende opbygning.

Eksempel på filnavn til disk slice i katalogerne **/dev/dsk** og **/dev/rdsk:**




Disk slice nummer 0 (nul) på hver UNIX partition refererer til hele den pågældende partition. Slice nummereringen foretages hexadecimalt, hvorved man ved et enkelt ciffer (0-9,a-f) kan opdele partitionen i 15 slices.

I **Device Databasen** (**DDB**) oprettes der almindeligvis referencer til såvel hele disken som hver enkelt anvendt disk slice.

**DDE** neddeler ikke diske ved hjælp af partitionering, dvs. der vil altid være et 0 (nul) efter d'et i specialfilen.

# 6.3 Opgave

*Formål: - at undersøge konfigurationen på maskinen*

1. **Undersøg maskinens konfiguration med kommandoen prtconf.**

2. **Kig i /dev/dsk og /dev/rdsk. Hvilke filer med navne af typen "c0t0d0s0" ligger der her?**

3. **Hvilke filer er der med navne, der starter med "f"?**

# 7 Sysadm menuen

De fleste af en systemadministrators opgaver kan varetages ved hjælp af **sysadm**. **Sysadm** består af en række hierarkisk opbyggede menuer og indtastningsvinduer. **Sysadm** startes med kommandoen

```
$ /usr/sbin/sysadm
```

hvorefter man skal indtaste **sysadms** password.

Som superbruger er det almindeligvis nok at skrive kommandoen således:

```
# sysadm
```

Når **sysadm** er startet mødes man af følgende skærmbillede:

```
UNIX SVR4.2MP Operations, Administration and Maintenance
 1                    UNIX System V Administration
>backup_service    - Backup Scheduling, Setup and Control
 ddebkup           - Supermax Backup System
 file_systems      - File System Creation, Checking and Mounting
 machine           - Machine Configuration, Display and Shutdown
 network_services  - Network Services Administration
 ports             - Port Access Services and Monitors
 printers          - Printer Configuration and Services
 restore_service   - Restore From Backup Data
 schedule_task     - Schedule Automatic Task
 software          - Software Installation and Removal
 storage_devices   - Storage Device Operations and Definitions
 system_setup      - System Name, Date/Time and Initial Password Setup
 users             - User Login and Group Administration

Move the cursor to the item you want and press RETURN to select it.

HELP |      | ENTER | PREV-FRM |      | NEXT-FRM | CANCEL | CMD-MENU |
  F1     F2     F3       F4             F5       F6       F7       F8
```

Funktionsrtasternes layout er vist under skærmbilledet (F1 til F8).

Man vælger mellem de forskellige punkter i menuerne vha. piletasterne efterfulgt af et **RETUR/ENTER**.

Man kan også indtaste forbogstaverne på de enkelte punkter, dvs. ønsker man at vælge menu punktet **printers**, taster man først et "**p**", hvorefter cursoren placeres på **ports**, eftersom begge menu punkter begynder med "**p**", og man derfor ikke har foretaget et fyldestgørende valg. Men taster man efterfølgende et "**r**", som er det andet bogstav i menu punktet **printers**, kan **sysadm** vha. "**pr**" indtastningen entydigt vælge punktet. Igen afsluttes valget med **RETUR/ENTER** tasten.

Når man kalder **sysadm**, kan man gå direkte til et menupunkt/undermenu ved at angive menupunktets forkortelse. Fx vil kommandoen

**# sysadm users**

gå direkte til menuen for "**User Login and Group Administration**".

Forneden på skærmbilledet ses en række felter, der angiver funktioner på funktionstasterne **F1** til **F8**. Disse bruges til at bevæge sig rundt mellem vinduer, lukke dem, etc.

Kan man af en eller anden årsag ikke benytte sine funktionstaster (terminalen kan være 'gået i baglås'), skal man ikke fortvivle. Istedet for et funktionstaste tryk kan man taste ^f (**Crtl + f**) efterfulgt af et nummer mellem **1** og **8**, som referer til nummeret på funktiontasten. Ønsker man fx at at taste **F3**, gøres det med tastekombinationen ^**f3**.

Specielt kan fremhæves **F6** (**CANCEL**) som lukker et vindue og **F7** som viser en menu, der bl.a. indeholder kommandoen "**exit**", der er nødvendig, hvis man skal lukke sysadm helt.

**F3** har funktionen **SAVE**, når man skal aktivere/gemme sine indtastninger i et indtastningsvindue.

**F2** skal ofte bruges til at markere valg i en liste, men også mulige valg kan **F2** ofte stille tilrådighed. Ved fx "**ja**" eller "**nej**" spørgsmål togles der mellem disse muligheder så indtastning undgås.

Ser **sysadm** menuen forkert ud allerede ved opstart, kan man forlade

den igen 'med lukkede øjne' ved tastekombinationen ^f7e efterfulgt af **RETUR/ENTER**. Denne tastesekvens er også værd at huske ved udfald af pil- og funktionstaster.

**F1** giver en oplysninger om og hjælp til at bruge **sysadm**. Er man i tvivl om, hvorledes en indtastning skal foretages, kan **F1** med fordel bruges.

Undermenuernes funktionstastlayout er ikke identiske. Således skal man i hver menu under **sysadm** først undersøge, hvilke funktionstaster som betyder hvad, inden man taster.

Et indtastningsvindue kan fx se ud som nedenstående vindue til brugeroprettelse.

```
4                          Add a User

  Comments:
  Login:
  User ID: 100
  Primary group: other
  Supplementary group(s):
  Create home directory? no

  Shell: /usr/bin/sh
  Number of days of login inactivity after which user ....:
  Login expiration date:
```

Bemærk det lille nummer øverst i venstre hjørne. Det fortæller hvor langt ned i undermenuerne man er kommet. I dette tilfælde er det den fjerde undermenu man er gået ind (eller ned) i.

Derudover er der en overskrift for det enkelte vindue, som beskriver den overordnede funktionalitet af vinduet.

Der kan i højre kant af vinduet være placeret en indikator som viser, om man er nederst eller øverst på en side, hvis kun et udsnit af siden vises i det aktuelle vindue. Selve indikatorfeltet er vist på **"Add a User"** vinduet.

# 7.1 Opgave

*Formål: - at bruge sysadm menuen*

1. **Gå ind i sysadm.**

2. **Bevæg dig lidt rundt vha. pil- og funktionstasterne. Brug RETURN/ENTER eller F3 (save) til at bekræfte et valg, og brug F6 (cancel) til at gå et trin tilbage.**

3. **Læg mærke til nummer, indikator i siden og overskrift på vinduerne.**

4. **Brug F1 (help) til hjælp.**

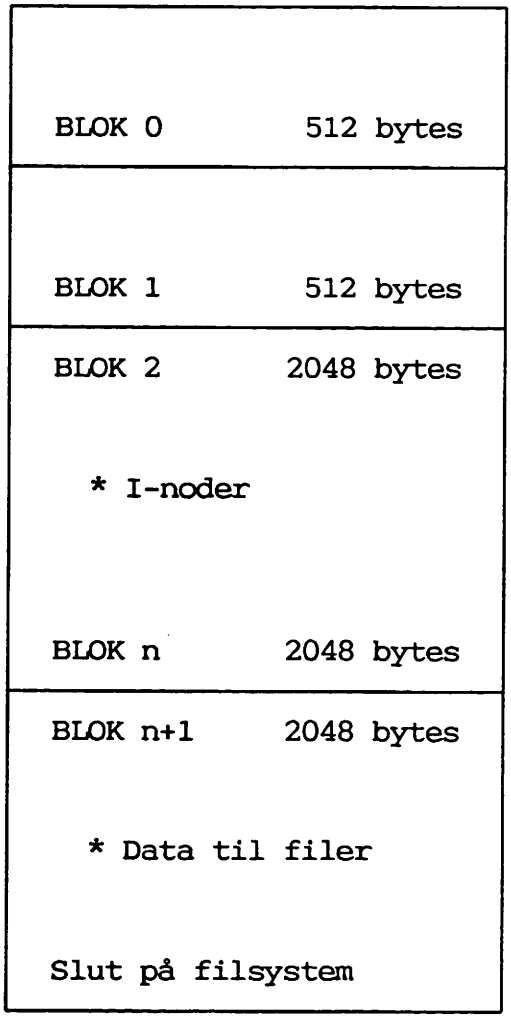
5. **Forlad sysadm vha. "krise tastning", dvs. du må ikke bruge funktionstasterne.**

# 8 Filsystemer

I UNIX SVR4.2 er den samlede katalogstruktur opbygget (vha. mount) af "disk slices", der kan indeholde en række forskellige filsystemer. Det der adskiller de forskellige typer filsystemer er måden, hvorpå disk slicen opdeles i blokke. Årsagen til forskellighederne er, at man ønsker at opnå en bestemt funktionalitet for et bestemt filsystem.

Nedenfor vises et eksempel på det oprindelige UNIX System V filsystem s5's opdeling af en disk.

Blokkene kan inddeles i 4 kategorier, som det fremgår af nedenstående tegning:




* Blok 0 anvendes til bootprocedurer. Bruges kun på roddisken.

* Blok 1 (superblokken) indeholder informationer om filsystemet, dets str. og hægte til frie i-noder.

* Et variabelt antal blokke, der indbefatter i-node liste. Består af blokke på 2048 bytes.

* Et variabelt antal lagerblokke, fortrinsvis indeholdende data. Består af blokke på 2048 bytes.

## 8.1 Filsystemtyper i UNIX SVR4.2

| Filsystemtype: | Kort beskrivelse: |
|---|---|
| vxfs | VERITAS filsystem. Indeholder en log, der forhindrer uregelmæssigheder i filsystemet ved fx crash. fsck ved boot bliver derfor hurtigere og muligheden for datatab mindre. |
| s5 | Traditionelt filsystem som det kendes fra SVR3 på Supermax'en |
| ufs | En type filsystem, der oprindeligt stammer fra BSD UNIX. Det tillader bl.a. brug af quota'er, som sætter en øvre grænse for den enkelte brugers diskforbrug. Quota'en skal startes, når brugeren oprettes. |
| sfs | Secure File System. Opbygningen er identisk med ufs, med tilføjelse af en ekstra i-node for hver fil. Den ekstra i-node indeholder sikkerhedsinformationer. |
| bfs | Boot File System. Et "fladt" filsystem, der mountes på /stand. Filsystemet indeholder bootfiler. |
| memfs | Memory Files System. Er det samme som det man kender som en RAM-disk fra fx MS-DOS. Et filsystem der kun eksisterer i RAM'en. Anvendes fx til temporære filer. |
| virtuelle | Derudover findes der en række virtuelle filsystemer, der afspejler informationer fra UNIX kernen i filform. Disse er /proc, /processor og /dev/fd. |
| nfs, rfs | Network File System og Remote File System. Filsystemer, der mountes fra andre maskine (over TCP/IP). |
| cdfs | CD-ROM File System. Filsystem, som bruges på cd-rom plader. |

## 8.2 I-noden

Et katalog i et filsystem indeholder filnavne og underkatalognavne med reference til deres respektive i-nodenumre.

I-noder indeholder følgende oplysninger:

* **type (almindelig, katalog, block, character eller pipe)**
* **rettigheder (læse-skrive-udføre)**
* **antal links til filen**
* **ejer og gruppe**
* **størrelse (antal bytes)**
* **dato for sidste tilgang**
* **dato for sidste ændring**
* **dato for oprettelse**
* **adresse på datablokke**

Da søgningen efter filer foregår lineært ned igennem et katalog, kan det betale sig at:

* **holde antallet af filer under et katalog lavt**

Herved vil søgetiden efter en bestemt fil kunne nedsættes. Imidlertid må man nok anbefale at placere filer, der logisk hører sammen i deres egne underkataloger for overskuelighedens skyld.

## 8.3 Standardkataloger i UNIX SVR4.2

Under rodkataloget ligger en række standardkataloger. Generelt bør man sørge for, at der kun ligger kataloger og ikke filer på dette niveau af hensyn til den tidligere omtalte søgetid i filsystemet.

En kort beskrivelse af standardkatalogerne:

| Navn | Type (standard) | Beskrivelse |
|---|---|---|
| /bck | mount point | Anvendes som mount point katalog til et evt. backup filsystem ved genindlæsning af fil-backup. |
| /bin | filsystem | Alle almindelige kommandoer under UNIX. |
| /config | tomt katalog | Har ingen anvendelse mere. |
| /dev | katalog | Indeholder specialfiler og underkataloger med specialfiler som normalt er associeret med hw-devices eller streams-devices. |
| /etc | katalog | Kataloger og filer, der indeholder information om maskinkonfiguration samt systemadministrative data. Alle "executables" er nu flyttet til /usr/sbin eller /sbin. (Der findes symbolske links fra /etc). |
| /export | katalog | Kan anvendes som rod-katalog til filsystemer, der skal eksporteres (med f.eks. NFS). |
| /home | filsystem | Filsystem til brugernes hjemmekataloger. |
| /home2 | filsystem | Alternativt filsystem til brugernes hjemmekataloger. |
| /install | mount point | Anvendes som mount point ved installationer. |
| /lost+found | katalog | Bruges af fsck til "tabte" filer/kataloger. |

| | | |
|---|---|---|
| /mnt | mount point | Mount point katalog til midlertidige mounts. |
| /opt | mount point | Mount point til anvendelse ved særlige applikationer (installation). |
| /proc | virtuelt | Indeholder informationer i filform om filsystem for forskellige processer, der kører på maskinen. |
| /save | katalog | Anvendes af sysadm, når der skal gemmes på floppy. |
| /sbin | katalog | Indeholder scripts/programmer der anvendes ved boot samt ved systemfejl |
| /stand | bfs filsystem | Indeholder de programmer og data, der anvendes ved boot af selve operativsystemet. |
| /tmp | memfs filsystem | Til temporære filer. |
| /usr | filsystem | /usr ligner sig selv med visse tilføjelser. Bl.a. indeholder /usr/sbin alle de systemadministrative programmer som tidligere lå under /etc. Under /usr/dde/bin ligger bl.a. udcp (UNIX to DOS copy). |
| /var | filsystem | Indeholder filer, der varierer fra maskine til maskine. F.eks. mail-, crontab- og logfiler. Indeholder desuden visse administrative kataloger, der hidrører fra installerede applikationer. |

# 84 Opgave

*Formål: - at undersøge hvilke standardkataloger, der findes på kursusmaskinen*
*Metode: - opgaven løses to og to*

1. **Undersøg ved hjælp af kommandoen ls, hvilke standardkataloger der findes på kursusmaskinen.**

2. **Diskuter om der er noget I ikke mener hører hjemme direkte under /-kataloget.**

3. **Hvordan vil I forhindre, at jeres brugere lægger filer direkte under /-kataloget?**

## 8.5 Oprettelse af nyt filsystem

Hvis man har en tom disk slice, der skal indgå i det samlede hierarki af kataloger og filer, skal der først oprettes filsystem på den.

Dette gøres med UNIX kommandoen **/sbin/mkfs**. Men inden vi kan oprette et nyt filsystem, skal vi have oplysning om, hvorledes disken er opdelt.

For at få en oversigt over, hvordan en given partition er opdelt i slices kan man anvende kommandoen **/usr/sbin/prtvtoc**. Eksempel (bemærk at character special filen til slice 0 angives som argument):

```
# /usr/sbin/prtvtoc /dev/rdsk/c0t0d0s0
slice 0:        DISK            permissions:    VALID UNMOUNTABLE
        starting sector:        40 (cyl 0)              length:
528040 (942.93 cyls)
slice 1:        ROOT            permissions:    VALID
        starting sector:        78960 (cyl 141)         length:
102480 (183.00 cyls)
slice 2:        SWAP            permissions:    VALID UNMOUNTABLE
        starting sector:        21280 (cyl 38)          length:
57680 (103.00 cyls)
slice 3:        USER            permissions:    VALID
        starting sector:        181440 (cyl 324)
length: 204960 (366.00 cyls)
slice 4:        HOME            permissions:    VALID
        starting sector:        386400 (cyl 690)
length: 30800 (55.00 cyls)
slice 7:        BOOT            permissions:    VALID UNMOUNTABLE
        starting sector:        32 (cyl 0)              length: 34
(0.06 cyls)
slice 8:        ALT SEC/TRK     permissions:    VALID UNMOUNTABLE
        starting sector:        74 (cyl 0)              length: 486
(0.87 cyls)
slice 10:       STAND           permissions:    VALID
        starting sector:        560 (cyl 1)             length:
20720 (37.00 cyls)
slice 11:       VAR             permissions:    VALID
        starting sector:        417200 (cyl 745)
length: 41440 (74.00 cyls)
```

```
slice 12:       HOME            permissions:    VALID
        starting sector:        458640 (cyl 819)
length: 30800 (55.00 cyls)
```

Hvis en disk slice umiddelbart skal kunne anvendes til et nyt filsystem skal "permissions" være **VALID**, og "slice tag" bør være **HOME**. Se også under kommandoen edvtoc i Command Reference a-l.

Syntaksen for mkfs-kommandoen er:

```
mkfs -F <filsystemtype> <character special fil> <størrelse>
```

"Character special fil" for disken findes under **/dev/rdsk** og slice'ns størrelse kan ses af ovennævnte output fra prtvtoc (length).

Eksempel på oprettelse af **vfxs** filsystem på **slice 12**:

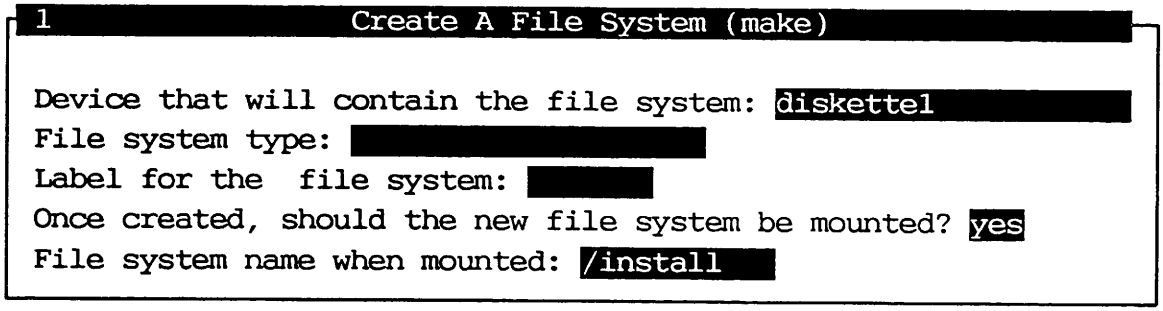
```
# mkfs -F vxfs /dev/rdsk/c0t0d0sc 61600
```

Bemærk, at sidste argument i **mkfs**, som angiver størrelsen af slicen, er dobbelt så stor som angivet ved **prtvtoc**. Dette skyldes, at **prtvtoc** angiver hele kilobytes, og **mkfs** bruger halve.

Filsystemer kan også oprettes fra **sysadm-menuen**. Under punktet "file systems" findes følgende menu:

```
1          Manage File Systems
 check     - Check a File System
 defaults  - Manage Defaults
 diskuse   - Display Disk Usage
 display   - Display Installed Types
 fileage   - List Files by Age
 filesize  - List Files by Size
 identify  - Identify File System Type
 list      - List Mounted File Systems
>make      - Create a File System
 mount     - Mount a File System
 unmount   - Unmount a File System
```

Herunder vælges punktet "make":




Det kan anbefales at anvende **sysadm** til dette og (især under de to øverste punkter) bruge f2 (CHOICES) til at vælge værdier, da man derved ikke fejlagtigt laver filsystem på fx boot-disken.

Menupunktet "identify" kan efterfølgende bruges til at checke hvilken filsystemtype, der er oprettet på en disk slice. Kommandoen **/sbin/fstyp** kan gøre det samme. Med option -v fås tillige information om filsystemets **superblock**.

Eksempler:

```
# fstyp /dev/rdsk/c0t0d0s4
vxfs

# fstyp /dev/rdsk/c0t0d0sc
ufs
```

# 86 Opgave

*Formål: - at oprette et filsystem på en disk slice*
*Metode: - opgaven løses i grupper*

1. **Opret et filsystem på den disk slice I får tildelt af instruktøren. Undlad at mounte disk slicen.**

2. **Check med kommandoen fstyp eller sysadm menupunktet "identify", at filsystemet er oprettet.**

3. **Check hvilke filsystemtyper, der er oprettet på de øvrige disk slices.**

## 8.7 Mountbegrebet

For at kunne benytte et filsystem skal det først mountes, dvs. at der skal skabes en forbindelse mellem et katalognavn og den disk slice, hvorpå der ligger eller skal indlægges fildata. Denne forbindelse skabes med kommandoen **/sbin/mount**. Syntaksen for mount-kommandoen er:

```
mount -F <filsystem type> <block special fil> <mount point>
```

Eksempel:

```
# mount -F ufs /dev/dsk/c0t0d0sc /home2
```

De disk slices, der indeholder filsystemer mountes normalt under opstart (boot) af maskinen. Dette sker gennem kommandoen **/sbin/mountall**, der henter sine oplysninger om, hvad der skal mountes, fra filen **/etc/vfstab**.

Kommandoen **/sbin/mountall** kan naturligvis også bruges under drift, hvis slices hurtigt skal mountes op. Er slices allerede mountet op, vil kommandoen give nogle fejlmeddelelser, som imidlertid kan ignoreres.

Ovenstående syntaks til mount-kommandoen kan skæres
ned til følgende, hvis disk **slicen** og **mount point** kataloget allerede er kendt i **/etc/vfstab**:

```
mount <block special fil> ell. mount <mount point>
```

Eksempel:

```
# mount /dev/dsk/c0t0d0sc
```

eller

```
# mount /home2
```

Gives mount-kommandoen helt uden argumenter fås en oversigt over hvilke disk slices, der er mountet til hvilke kataloger.

Eksempel:

```
# mount
/ on /dev/root read/write/setuid on Wed Mar  1 12:33:14 1995
/proc on /proc read/write on Wed Mar  1 12:33:14 1995
/stand on /dev/dsk/c0t0d0sa read/write on Wed Mar  1 12:33:15 1995
/dev/fd on /dev/fd read/write on Wed Mar  1 12:33:15 1995
.
.
.
/tmp on /tmp read/write on Wed Mar  1 12:33:39 1995
/var/tmp on /var/tmp read/write on Wed Mar  1 12:33:39 1995
```

Forbindelsen mellem en disk **slice** med filsystem og et **mount point** katalog kan ophæves med kommandoen **/sbin/umount**. Som argument angives enten mount point kataloget eller block special filen for disk slicen.

Eksempel:

```
# umount /home2

eller

# umount /dev/dsk/c0t0d0sc
```

Alle disk slices umountes med kommandoen **/sbin/umountall**. Diske der er i brug umountes dog ikke.

# 8.8 Opgave

*Formål: - at opnå fortrolighed med begrebet mount*
*Metode: - opgaven løses i grupper*

1. **Undersøg hvilke disk slices der er mountet på kursusmaskinen.**

2. **Prøv at afmontere en af subdiskene på maskinen. NB Aftal med instruktøren, hvilken subdisk I skal umounte.**

3. **Prøv nu at liste indholdet i det katalog, som subdisken var mountet til. Hvad sker?**

4. **Genskab nu forbindelsen mellem subdisk og filsystem, og check med kommandoen mount, at I har fået mountet korrekt.**

## 8.9 Undersøgelse af fejl i filsystemet

Når en UNIX SVR4.2 maskine bootes undersøges filsystemer automatisk for fejl. Samtidigt med, at der udføres dette såkaldte filsystem-check, rettes eventuelle fejl automatisk, hvis det er muligt. I filen **/etc/vfstab** angives der hvilke filsystemer, man ønsker at få checket under boot-proceduren.

Skulle man selv ønske at undersøge et filsystem for fejl, er det muligt at foretage den samme undersøgelse som under boot med kommandoen **/sbin/fsck**. Kommandoen bør foretages fra konsollen i enkeltbruger tilstand. For at kunne benytte **fsck** skal man først umounte det filsystem, man ønsker at undersøge. Syntaksen for **fsck**-kommandoen er **fx:**

```
fsck -F <filsystem type> <block ell. character special fil>
```

Eksempel

```
# fsck -F vxfs /dev/dsk/c0t0d0sc
file system is clean - log replay is not required
```

**fsck** laver automatisk fejlretning under boot, men udføres kommandoen fra shell, kører den interaktivt.

Hvis kommandoen **fsck** under udførelse møder en 'fejlbehæftet' fil i filsystemet, som det ikke kan rette op på, vil denne fil blive smidt væk. Hvis man i det øverste niveau i filsystemet på den pågældende disk har et katalog lost+found, vil filen blive placeret her. I nogle tilfælde vil man kunne genetablere en ny korrekt opbygget fil herfra ved kopiering, andre gange vil filen være rent kaos uden mulighed for genetablering.

Når man er færdig med undersøgelserne, skal man naturligvis huske at mounte sine filsystemer, så der igen kan arbejdes på systemet.

# 8.10 Opgave

*Formål: - at afprøve fsck*
*Metode: - opgaven løses i grupper*

1. **Umount jeres disk slice fra forrige opgave.**

2. **Udfør et filsystemscheck på den.**

3. **Mount disk slicen igen.**

## 8.11 Filen /etc/vfstab

I filen **/etc/vfstab** kan man angive, hvad der skal ske med de forskellige disk slices, når maskinen bootes.

Formatet på en linie i **/etc/vfstab**:

*special fsckdev mountp fstype fsckpass automount mntops macceiling*

special: Block special fil for disken (eller ressourcenavn ved nfs/rfs)

fsckdev: Character special fil for disken, hvis den findes. Ellers bruges block special filen. "-" hvis der ikke er nogen device fil.

mountp: Navn på mount point kataloget.

fstype: Filsystemtype.

fsckpass: "-" angiver ingen automatisk fsck. Et tal (f.eks. 1) angiver, at automatisk fsck skal udføres. (Se under fsck-kommandoen).

automnt: "yes" ell. "no" for automatisk mount ved boot.

mntops: Relevante options til mount af det pågældende filsystem. (Se under mount-kommandoen).

macceiling: Anvendes kun, hvis "enhanced security" er installeret.

Eksempel på linie i /etc/vfstab:

/dev/dsk/c0t0d0sc /dev/rdsk/c0t0d0sc /home2 ufs 1 yes - SYS_RANGE_MAX

# 8.12 Opgave

*Formål: - at tilrette /etc/vfstab*
*Metode: - opgaven løses i grupper*

1. **Tilret /etc/vfstab således, at jeres disk slice fra tidligere mountes under boot, angives med den rigtige filsystem type og filsystem check udføres automatisk.**

## 8.13 Pladsovervågning

En meget vigtig del af systemadministratorens arbejde er at sikre en fornuftig udnyttelse af systemets ressourcer, herunder også diskpladsen. Hvis man jævnligt checker pladsen på disken, undgår man kedelige situationer, hvor diske løber fulde til stor gene for brugerne på systemet.

Ved hjælp af sysadm kan man få en rapport over udnyttelsen af de mountede subdiske.

*** Vælg punktet "file_systems" fra sysadm hovedmenu**

*** Vælg punktet diskuse fra file_systems menuen**

2 Displays File System Disk Usage Information

FILE SYSTEM USAGE AS OF 03/02/95 10:39:51

| File System | Free Blocks | Total Blocks | Percent Full | |
|---|---|---|---|---|
| / | 65376 | 102480 | 36% | |
| /dev/fd | 0 | 0 | 0 | |
| /home | 17584 | 20720 | 15% | |
| /home2 | 45448 | 50510 | 10% | |
| /proc | 0 | 0 | 0 | |
| /stand | 13867 | 20720 | 33% | |
| /system/processor | | 0 | 0 | 0 |
| /tmp | 16376 | 16384 | 0% | |
| /usr | 73552 | 204960 | 64% | |
| /var | 33696 | 41440 | 18% | |
| /var/tmp | 16384 | 16384 | 0% | |

Der findes også en række UNIX-kommandoer, der kan give en oversigt over diskpladsforbrug. Herunder er en liste med eksempler:

**df** (usr/bin/df) Udskriver en liste over antal frie blokke (á 512 bytes) på de mountede disk slices. Med option t fås også det totale antal blokke. Hvis der angives et navn på en disk slice eller et mount point katalog som argument, udskrives oplysningerne kun for den pågældende disk slice Hvis disk slicen, der er mountet til /home ønskes checket for frie blokke, gives følgende kommando:

```
# df /home    eller
# df -t /home    eller
# df -k /home (viser kilobytes i stedet for blokke)
```

**dfspace** (/sbin/dfspace) Er et script, der anvender df, men konverterer antal blokke til Mb. Der kan angives et navn på en disk slice eller et mount point katalog som argument, eller kommandoen kan gives uden argumenter, hvorved der fås en oversigt over alle disk slices. Eksempler

```
# dfspace eller
# dfspace /home
```

**du** Hvis man ønsker et fingerpeg om, hvor meget et bestemt katalog inkl. underkataloger optager af en disk slice, kan dette fås med kommandoen du -s. F.eks. hvis man ønsker at vide, hvor meget /home optager:

```
# du -s /home
```

Hvis man ønsker overblik over, hvor meget hver enkelt fil optager af plads under kataloget /home, gives kommandoen:

```
# du -a /home
```

# 8.14 Opgave

*Formål: - at afprøve kommandoer til overvågning af diskplads*

1. **Prøv dels ved hjælp af sysadm og dels direkte fra shell at undersøge kursusmaskinen m.h.t. fri diskplads.**

2. **Prøv evt. også at undersøge, hvor meget filerne i dit hjemmekatalog optager af plads.**

3. **Hvor stor er /root på kursusmaskinen?**

## 8.15 Oprydning i filsystemet

Filsystemet indeholder en del filer, som UNIX operativsystemet bruger til at udskrive forskellige meddelelser i. Disse filer kan blive så store som ulimit tillader og derfor fylde godt op på disken. Disse filer er bl.a. systemets logfiler:

| | |
|---|---|
| * /var/adm/log | Katalog med logfiler til diverse utility-programmer fx cs (Connection Server) |
| * /var/adm/lastlog | Log over indlogninger |
| * /var/adm/sulog | Log over su-kommandoen |
| * /var/adm/syslog/alert<br>/var/adm/syslog/crit<br>/var/adm/syslog/emerg<br>/var/adm/syslog/err<br>/var/adm/syslog/info<br>/var/adm/syslog/notice<br>/var/adm/syslog/warning | Katalog med logfiler til selve UNIX systemet |
| * /var/adm/wtmp | Log over login på systemet |
| * /var/cron/log<br>/var/cron/olog | Log over cron aktiviteter |
| * /var/lp/logs/lpNet<br>/var/lp/logs/lpsched<br>/var/lp/logs/requests | Katalog med logfiler til LP Print Service |
| * /var/saf/_log<br>/var/saf/inetd/log<br>/var/saf/inetd/o.log<br>/var/saf/nbt/log<br>/var/saf/nbt/o.log<br>/var/saf/tcp/log<br>/var/saf/tcp/o.log | Logfiler til Service Access Facility |
| * /var/sadm/bkup/logs | Katalog med logfiler til backup |

systemet

En logfil som

```
* /var/adm/shut.log
```

skal ikke afkortes. Den indeholder information om boot, men overskrives - derfor vokser filen ikke.

**Sar** (System Activity Reporter) har et katalog med logfiler under stien **/var/adm/sa**. Disse logfiler skal heller ikke afkortes eller slettes. Det holder **/usr/sbin/sar** selv styr på.

Logfilerne er med undtagelse af **/var/adm/wtmp** almindelige tekstfiler, som nemt kan reduceres i størrelse, fx således at kun de sidste linier i filerne gemmes.

Det er muligt at forkorte filerne ved hjælp af UNIX kommandoer. Skal fx **/var/cron/log** afkortes til kun at indeholde de sidste 20 linier sker det på følgende måde med kommandoen **/usr/bin/tail**:

```
# /usr/bin/tail -20 /var/cron/log > /tmp/logfil
# cp /tmp/logfil  /var/cron/log
# rm /tmp/logfil
```

Denne afkortning af logfiler vil man typisk udføre ved jævnligt at kalde et oprydningsscript fra **cron**.

**/var/adm/wtmp** er en datafil, der indeholder en historisk oversigt over login på maskinen. Denne kan ikke afkortes som en tekstfil, men kan istedet tømmes med passende mellemrum på følgende måde:

```
$ cp /dev/null /var/adm/wtmp  eller
$ > /var/adm/wtmp
```

Kataloget **/var/mail** er systemets postkasse for modtaget post. Det er en god skik at rydde op heri, når man har læst sin post. Systemadministratoren bør checke, at der ikke ligger store filer og fylder op i dette katalog. I værste fald vil man ikke længere kunne modtage post, hvis modtagefilen har nået en størrelse svarende til ulimit.

Af forskellige grunde kan der opstå dump-filer forskellige steder i filsystemet. Disse filer kan blive meget store, hvorfor de med jævne mellemrum bør slettes. "core" indgår som en del af navnet i disse filer, og man kan med en **/usr/bin/find** kommando umiddelbart finde dem således:

```
# find / -name 'core.*' -print
```

Herefter kan navnets sammensætning nærmere undersøges, samt hvor filen er placeret i filsystemet. Man kan nu tage stilling til, hvorvidt filen skal slettes eller ej.

Der bør på det kraftigste advares mod IKKE at anvende en automatisk sletning af disse filer med find, som ved fx at udføre kommandoen

```
# find / -name 'core.*' -exec rm {} \;
```

Har man mistanke om, at systemet indeholder gamle eller store filer, som evt. kan slettes, kan man med sysadm få et overblik herover.

*** Vælg punktet file_systems fra sysadm hovedmenu**

*** Vælg punktet fileage fra file_systems menuen**

```
1    Lists Files Older than a Particular Date

Directory to be searched:
/usr
Age of files to be reported, in days: 90
```

Hvorefter man i et nyt vindue får en liste over filer under /usr, der er ældre end 90 dage.

I UNIX udføres ovenstående kommando ved hjælp af kommandoen **/usr/-bin/find:**

```
# find /usr -mtime +90 -print
```

På samme måde kan man med menupunktet **filesize** søge efter de største filer under et bestemt katalog.

I UNIX arbejder man med forskellige tidsstempler, som alle kan bruges i **/usr/bin/find** kommandoen. De forskellige tidsstempler med find kommandoens option i parentes er oprettelses-, modifikations- (-mtime) og accesstidspunkt (-atime).

Forskellen på access og modifikation er enkel. Har man skrevet i filen er den blevet modificeret. Har man man blot set i filen eller udført den, hvis den er eksekverbar, er filen blevet access'et, dvs. man har haft tilgang til den.

Det er indlysende, at alle UNIX kommandoer fx ikke bør slettes, selvom deres modifikationstidspunkt er af ældre dato. De bruges dagligt, og derfor er deres accesstidspunkt nyt.

# 8.16 Opgave

*Formål: - at undersøge filsystemer med hensyn til logfiler, core-filer og gamle eller store filer.*
*Metode: - opgave 5 løses i grupper*

1. **Undersøg om der på kursusmaskinen findes brugere, der har filer, som ikke er rørt i de sidste 90 dage. Du bestemmer selv, hvilket værktøj du vil anvende.**

2. **Find og udskriv til skærmen en liste over core-filer på systemet.**

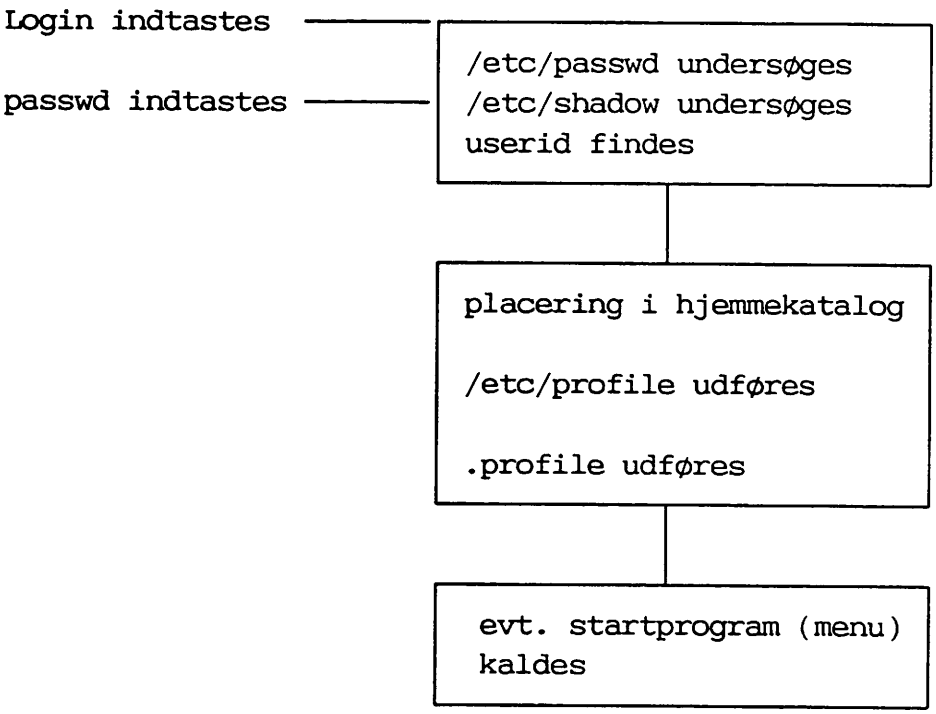
3. **Ved hjælp af sysadm udskriv en liste over de 10 største filer i et katalog f.eks. kataloget /bin.**

4. **Opret en fil der indeholder 20 linier. Prøv at afkorte den til kun at indeholde de sidste 10 linier.**

5. **I får opgivet navnet på en logfil af instruktøren. Afkort denne logfil.**

# 9 Brugeradministration

Når en bruger logger ind læses forskellige filer: **/etc/passwd**, **/etc/shadow**, **/etc/profile** og brugerens **.profile**. Proceduren fremgår af følgende skitse:




Dette betyder omvendt, at en bruger er kendt af systemet ved definitioner i de nævnte filer, og det er i disse filer, at en brugers kendetegn opsættes.

## 9.1 /etc/passwd

**/etc/passwd** er den fil, hvor alle brugere af systemet skal være indskrevet. Et eksempel på filen vises i følgende udskrift:

```
root:x:0:3:0000-Admin(0000):/:/sbin/sh
daemon:x:1:12:0000-Admin(0000):/:
bin:x:2:2:0000-Admin(0000):/usr/bin:
sys:x:3:3:0000-Admin(0000):/:
adm:x:4:4:0000-Admin(0000):/var/adm:
uucp:x:5:5:0000-uucp(0000):/usr/lib/uucp:
nuucp:x:10:10:0000-uucp(0000):/var/spool/uucppublic:/usr/lib/uucp/uucico
nobody:x:60001:60001:uid no body:/:
noaccess:x:60002:60002:uid no access:/:
lp:x:7:9:0000-LP(0000):/var/spool/lp:/usr/bin/sh
listen:x:37:4:Network Admin:/usr/net/nls:/usr/bin/sh
sysadm:x:0:0:general system administration:/usr/sadm:/usr/sbin/sysadm
owner:x:101:1:owner of jahangir:/home/owner:/usr/bin/sh
smtp:x:55:6:SMTP Processes:/var/spool/smtpq:/usr/bin/sh
agn:x:100:1:Anders Gilbro:/home/agn:/usr/bin/sh
mes:x:102:1:Merete Schmidt:/home/mes:/usr/bin/sh
test1:x:103:1:En test bruger:/home/test1:/usr/bin/sh
```

Én linie i denne fil repræsenterer en bruger. Linien følger en bestemt syntaks, idet oplysningerne står opført i nedenstående rækkefølge adskilt af : (kolon):

**login:kodeord:idnr:gruppeid:kommentar:hjemmekatalog:startprog**

Nedenfor forklares de forskellige elementer i /etc/passwd filen:

**login** Indlogningsnavn, der tastes i velkomstbilledet.

**kodeord** I dette felt står der altid "x". I tidligere versioner indeholdt feltet kodeord og oplysninger om forældelse af kodeord. Disse oplysninger figurerer nu i /etc/shadow. Feltet bevares af kompatibilitetsgrunde.

**userid** Nummer som knytter sig til et login.

**gruppeid** Nummer som refererer til filen /etc/group, hvor navnet på den aktuelle gruppe hentes.

**kommentar** Her skrives som oftest forskellige oplysninger om brugeren, fx navn, tlf., etc.

**hjemme-** Det katalog i filsystemet en bruger skal placeres i ved **lokatalog** gin.

**start-** Dette felt skal ikke udfyldes, men ønskes et bestemt **startprogram** program, kan kaldet indsættes her.

Det er meget vigtigt at denne opsætning ikke ødelægges. Fejl i **/etc/passwd** filen kan i værste fald resultere i, at ingen overhovedet kan logge sig på systemet. Man bør derfor altid have en kopi af filen, hvis uheldet skulle være ude. Det vil i så fald kunne spare mange timers arbejde.

## 9.2 /etc/shadow

**/etc/shadow** indeholder oplysninger om den enkelte brugers kodeord samt oplysninger om forældelse af dette, og hvornår brugeren skal (eller kan) ændre det.

```
root:Hg.IvxbdamHS.:9189:0:168:7:::
daemon:NP:6445::::::
bin:NP:6445::::::
sys:NP:6445::::::
adm:NP:6445::::::
uucp:NP:6445::::::
nuucp:NP:6445::::::
nobody:NP:6445::::::
noaccess:NP:6445::::::
lp:*LK*:::::::
listen:*LK*:::::::
sysadm:syLCAKuYnre/k:9190:0:168:7:::
owner:Ed1IiBN7hAbvk:9189:0:168:7:::
smtp:*LK*:::::::
agn:hQGof7RFK6vYw:9190::::::
mes::9191::::::
test1:M5aINrwRMnlag:9192:10:20:3:100:9678:
```

Formatet på en linie i **/etc/shadow** er følgende:

**login:kodeord:sidste ændring:min. ændringsinterval:max. gyldighed:advarsel:max. inaktivitet:udløbsdata:flag**

Forklaring af de enkelte elementer i **/etc/shadow**:

| | |
|---|---|
| **login** | **Indlogningsnavn** |
| **kodeord** | **Det krypterede kodeord angivet ved 13 tegn.** |
| **sidste ændring** | **Dato for sidste ændring givet ved antallet af dage, der er gået siden d. 1. januar, 1970.** |

| | |
|---|---|
| min. ændringsinterval | Det mindste antal dage, der må gå mellem to ændringer af et kodeord |
| max. gyldighed | Det maksimale antal dage en bruger må anvende det samme kodeord |
| advarsel | Antal dage brugeren skal advares før et skift af kodeord |
| max. inaktivitet | Antal brugerens login må stå ubenyttet før kontoen spærres |
| udløbsdato | Dato for kontoens udløb. Angivet ved antal dage efter d. 1. januar, 1970 |
| flag | Et (evt.) flag der angiver en bestemt password generator. |

## 9.3 /etc/profile

Når en bruger arbejder, aktiveres programmer og processer på en UNIX maskine. Alle programmer udføres i et bestemt miljø, som er de shell variable, der gælder for den pågældende bruger. For at fortælle systemet hvilket miljø en bruger arbejder i, sættes disse variable ved indlogning. De variable, der skal gælde for alle brugere sættes i **/etc/profile** filen.

Nedenstående vises et eksempel på **/etc/profile**:

```
#ident" (#)/etc/profile.sl 1.1 SVR4.2MP 11/14/93 31562 USL"
#ident"$Header: profile 1.1 91/04/29 $"

trap "" 1 2 3
umask 022# set default file creation mask
export LOGNAME

. /etc/TIMEZONE

case "$0" in
-jsh ¦ -ksh ¦ -rsh ¦ -sh)
# issue message of the day
trap : 1 2 3
echo ""# skip a line
if [ -s /etc/motd ] ; then cat /etc/motd; fi

trap "" 1 2 3
# set default attributes for terminal
stty erase '^h' echoe

# check if it is an at386
/usr/sbin/isat386
if [ $? = 0 ]
then
# it is an at386, check for color
/usr/sbin/adpt-type
if [ $? = 3 ]# VGA color?
then
TERM=AT386# color at386
# work around Dell Fastwrite VGA
```

```
# problem -- force text mode
# to color 80x25
stty VGA-C80x25 1>/dev/null 2>/dev/null
else
TERM=AT386-M# default to monochrome
fi
fi

if [ "$TERMCAP" = "" ]
then TERMCAP=/etc/termcap
fi
export TERM TERMCAP

# check mailbox and news bulletins
mailcheck 2>/dev/null
if [ $LOGNAME != root -a -d /var/news ]
then news -n
fi
#Uncomment this script if you wish to use secure RPC facility
.
.
.
export PATH;
trap 1 2 3
```

En af de variable, der ofte sættes i /etc/profile er umask. Denne maske benyttes til at bestemme hvilke rettigheder brugernes filer vil blive oprettet med. I UNIX oprettes filer standard med rettighederne 666, kataloger med rettighederne 777. Hvis der er sat en umask for den bruger der opretter filer eller kataloger, trækkes umaskens værdi fra standardrettighederne. I ovenstående eksempel skal umasken 022 trækkes fra standardrettighederne:

| | Kataloger: | Filer: |
|---|---|---|
| standardrettigheder | 777 | 666 |
| umask | 022 | 022 |
| rettigheder på fil/katalog | 755 | 644 |

## 9.4 .profile

De variable som er særegne for en bruger sættes i **.profile** filen, der ligger i brugerens hjemmekatalog. Det program som brugeren skal starte op i vil også typisk være defineret i denne fil. Et almindeligt startprogram er hovedmenuen til SM-kontor.

Et eksempel på en **.profile** fil ses nedenfor:

```
MAIL=/usr/mail/${LOGNAME:?}
HZ=100
export MAIL HZ
TERM=`vti -i -v -p int/dde450.t`
PATH=$PATH:/sbin:/usr/sbin:$HOME:$HOME/bin; export PATH
PS1="`uname -n`:$LOGNAME$ ";export PS1
stty erase '^?'
stty kill  '^o'
set -o vi
```

## 9.5 Oprettelse af brugere

Når en bruger oprettes, skal oplysninger om brugeren indskrives i **/etc/passwd** og **/etc/shadow**, og der skal evt. oprettes et hjemmekatalog med tilhørende **.profile**.

Det er muligt at oprette brugere forholdsvis enkelt v.hj.a. en skærmdialog i sysadm menuen. Sysadm vil selv sørge for at definitionen af brugeren indskrives i **/etc/passwd** og **/etc/shadow** i overensstemmelse med de svar, man giver i skærmdialogen.

*** Vælg punktet users fra sysadm hovedmenu**

Herefter vil følgende skærmdialog fremkomme:

```
2       User Login and Group Administration
>add      - Add Users or Groups
 defaults - Define Defaults for Adding Users
 list     - List Users or Groups
 modify   - Modify Attributes of Users or Groups
 password - (Re-)define User Password Information
 remove   - Remove Users or Groups
```

Dette er hovedmenuen for brugere og gruppe håndtering. Man kan sætte standardværdier til brugeroprettelse vha. defaults indgangen i menuen.

Disse standardværdier gemmes, sammen med andre standardværdier, nede i /etc/default kataloget under filnavnet useradd. Nede i **/etc/default** kataloget ligger der også standardopsætninger til password (passwd) og brugersletning (userdel). Standardopsætningerne kan også ændres vha. UNIX kommandoen **/sbin/defadm**.

Man kan udmærket oprette, ændre og slette brugere og grupper ude fra shell'en med UNIX kommandoerne **/usr/sbin/useradd**, **/usr/sbin/usermod** og **/usr/sbin/userdel** samt **/usr/sbin/groupadd**, **/usr/sbin/groupmod** og **/usr/sbin/groupdel** for henholdsvis brugere og grupper.

Inde fra sysadm vil den fortsatte skærmdialog ved brugeroprettelse

se ud på følgende måde:

*** Vælg punktet add fra users menuen**

```
3                          Add a User

Comments: _____________________________________________
Login: ___________
User ID: 104
Primary group: other
Supplementary group(s): _______________________________
Create home directory? yes
Home directory: _______________________________________
Shell: /usr/bin/ksh
Number of days of inactivity after which user cannot login:
Login expiration date: ________
```

hvorefter følgende skærmdialog fremkommer:

```
4                  Define User Password Information

Password status: lock
Maximum number of days the password is valid: 24
Minimum number of days allowed between password changes: 0
Number of days for warning message: 1
```

# 9.6 Opgave

*Formål:* - *at oprette gruppe ved hjælp af sysadm*
- *at oprette brugere ved hjælp af sysadm*

1. **Kald sysadm.**

2. **Gå ind i punktet users.**

3. **Gå ind i punktet add og tilføj en ny gruppe som hedder dit fornavn.**

4. **Prøv menupunktet list for at se grupper og brugere.**

5. **Gå ind i punktet add og opret 2 nye brugere inkl. dig selv på systemet. De nye brugere skal tilhøre den gruppe, du netop oprettede. Giv nogle af dine brugere password. NB Du får tildelt userid af instruktøren.**

6. **Log ud og prøv at logge ind som de nyoprettede brugere og undersøg, hvad navnet på hjemmekataloget er, samt hvilke filer der ligger her.**

7. **Prøv ved hjælp af pg at undersøge /etc/passwd og /etc/shadow for at se hvorledes dine nyoprettede brugere ser ud her.**

## 9.7 Ændring og sletning af brugere

Der er to menupunkter i users menuen, der anvendes til at ændre på specifikationerne for en bruger eller en gruppe.

Under menupunktet **modify** kan man ændre på de elementer, der specificeres i billederne "Add a user" (se ovenfor) og "Add a group".

Under menupunktet **password** kan man ændre på de elementer, der specificeres i billedet "Define User Password Information".

Menupunktet **remove** kan anvendes til at slette en bruger eller gruppe.

Når man sletter en bruger vha. sysadm, skal man være klar over, at bruger ID'et for den slettede bruger ikke kan anvendes et år frem i tiden. Er dette imidlertid utilfredsstillende, kan man enten slette brugeren ude i shell ved UNIX kommandoen **/usr/sbin/userdel**

# userdel -r -n0 <bruger_login>

eller også skal man slette brugerens ID inde i filen **/etc/security-/ia/ageduid.**

# 9.8 Opgave

*Formål:* - *at ændre en brugers login oplysninger*
- *at slette en bruger*

1. **Prøv at ændre kodeord på dit eget login.**

2. **Prøv at sætte dit login til en maksimal varighed på 30 dage.**

3. **Kodeordsforældelse skal nu gælde for dig. Sæt min. og maks. varighed af kodeord til hhv. 1 og 8 uger.**

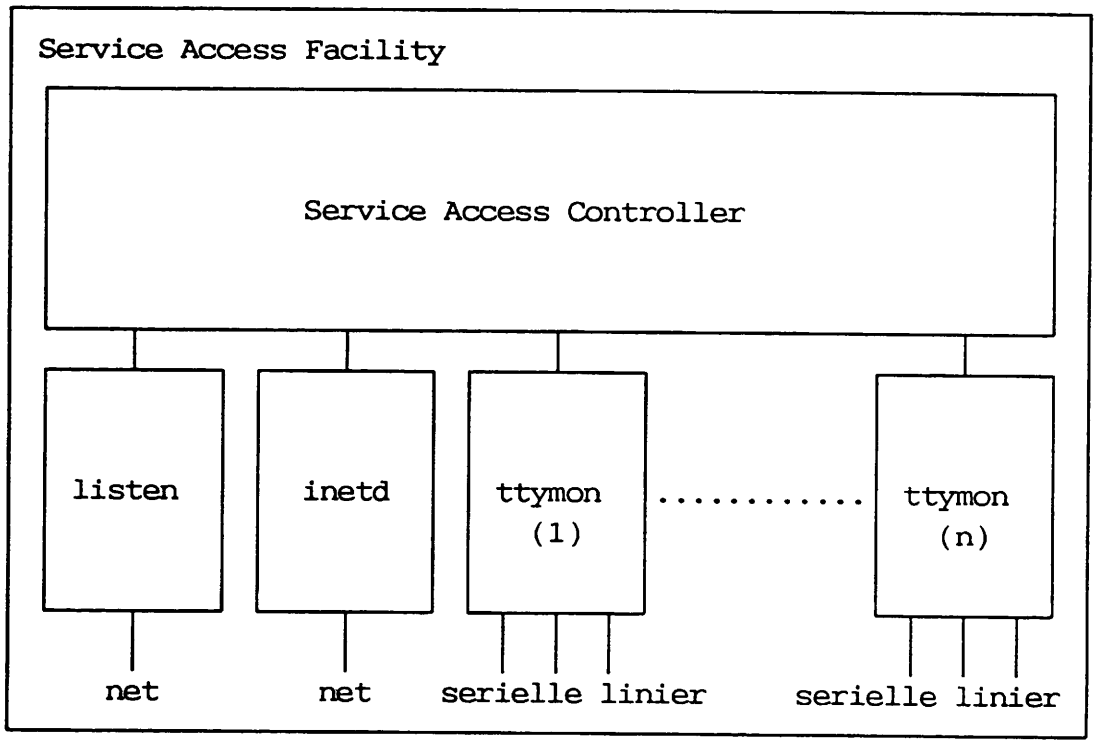
4. **Prøv at ændre hjemmekatalog for en af de brugere du oprettede i sidste opgave.**

5. **Prøv ved hjælp af pg at undersøge /etc/passwd for at se hvordan ændringerne ser ud.**

6. **Slet en af de brugere du oprettede i sidste opgave.**

# 10 SAF

I UNIX SVR4.2 styres terminalopkoblinger og netværksservices af et samlet system, der hedder "Service Access Facility" (SAF). Nedenfor vises en figur der skitserer SAF's opbygning:




**Service Access Controller** (**SAC**) er et overordnet program, der starter og stopper forskellige portmonitorer, der håndterer indkommende opkald til maskinen fra eksterne linier. Det være sig serielle eller netlinier. Udfra konfigurationsfilerne **/etc/saf/_sysconfig** og **/etc/saf/_sactab** finder SAC ud af hvilke portmonitorer, der skal startes. Programfilen for SAC hedder **/usr/lib/saf/sac** og startes selv fra **/etc/inittab** vha. følgende linie:

```
sc:234:respawn:/usr/lib/saf/sac -t 300
```

**listen** er en type portmonitor, der overvåger om der kommer indkommende "requests" fra nettet. Listen portmonitoren kan stille forskellige "services" til rådighed. Fx kan den lytte efter print-requests fra eventuelle klienter, der vil skrive ud gennem maskinens LP Print Service eller det kan være UUCP opkald fra andre maskiner via nettet.

**inetd** er portmonitor for alle services, der kan ydes i henhold til TCP/IP kommunikationsprotokollen. Fx telnet- eller FTP-opkald. "inetd" startes som en del af SAF, men læser sine konfigurationer (hvilke services den skal starte) fra filen **/etc/inet/inetd.conf.**

**ttymon** er en type portmonitor, der kan overvåge serielle linier. Den kan overvåge flere linier på een gang. Man kan have flere ttymon portmonitorer, der "deler" de serielle linier imellem sig.

## 10.1 Betingelser for terminalopkoblinger

Der gælder forskellige betingelser for, at en terminalopkobling kan fungere/etableres.

**For serielt opkoblede terminaler gælder følgende:**

- Der skal køre en ttymon portmonitor, der yder login-service på den pågældende port (fx **/dev/tty01**)
- Portens stty-parametre skal stemme overens med terminalens setup mht. kommunikation

For terminaler opkoblede via net gælder følgende:

- inetd portmonitoren skal være startet af SAC
- inetd portmonitoren skal i **/etc/inet/inetd.conf** være konfigureret til at yde telnet og login (rlogin) service. Dvs. følgende linier skal være tilstede i filen:

  telnet stream tcp nowait root **/usr/sbin/in.telnetd in.telnetd**
  login stream tcp nowait root **/usr/sbin/in.rlogind in.rlogind**

**stty-parametre:**

Terminalens stty- eller kommunikationsparametre dækker over mange forskellige værdier, der bestemmer hvorledes maskinen kommunikerer, bl.a. hastighed, paritetscheck, om <ctrl> + s, <ctrl> + q virker etc.

En terminals stty-parametre aflæses:

```
$ stty -a                          (aflæse for egen terminal)
# stty -a < /dev/tty23             (aflæse for tty23)
# stty ispeed 19200 ospeed 19200   (sæt parametre)
```

Man kan kun sætte stty-parametre for sin egen terminal. Dette kan imidlertid også gøres af SAF systemet, idet man man definerer hvilke stty-parametre, der skal gælde for en port, når der oprettes en service på porten.

**Terminalens setup:**

Terminalens interne setup bestemmer bl.a. hvorledes terminalen kommunikerer med maskinen m.h.t. hastighed, paritetscheck, o.l. stty-parametrene og terminalens setup skal derfor stemme overens på disse punkter. I setuppen bestemmes også, hvordan tastaturet og skærmen virker m.h.t. sprogkode, markør, skærmfarve etc.

## 10.2 Terminfo og TERM variablen

Når et program i UNIX vil foretage en eller anden form for skærmhåndtering sendes en række escape-sekvenser til terminalen. Det kan f.eks. være escape-sekvenser, der får tegn til at blinke, terminalen til at bippe eller rykker markøren et bestemt sted hen på skærmen.

De forskellige typer terminaler har ofte forskellige escape-sekvenser til de forskellige operationer. Men for at programmerne ikke skal kende alle terminalers escape-sekvenser eksisterer der i UNIX en database, der indeholder de forskellige escape-sekvenser til en lang række terminaler. Denne kaldes for terminfo databasen.

De forskellige programmer skal blot vide hvilken terminal i terminfo databasen, man arbejder ved for at kunne sende de rigtige escape-sekvenser. Denne oplysning får de fra shell variablen TERM. Er TERM=vt100 vil alle skærmoperationer fra programmet af UNIX blive oversat til de korrekte escape-sekvenser ud fra oplysningerne i terminfo databasen.

De forskellige "konverteringstabeller" ligger under kataloget **/usr/share/lib/terminfo**. Her findes en række underkataloger, hvis navne er forbogstaverne på de terminaltyper for hvilke der opbevares konverteringstabeller i det pågældende katalog.

Fx findes konverteringstabellen til vt100 terminaler i kataloget **/usr/share/lib/terminfo/v** og konverteringstabellen til ansi terminaler i kataloget **/usr/share/lib/terminfo/a**. Ved en gennemgang af filnavnene i katalogerne under **/usr/share/lib/terminfo** kan man såle-

des se hvilke gyldige værdier, der er gældende for shell variablen TERM.

## 10.3 Fejlretning på terminaler

Af forskellige grunde kan der opstå problemer med kommunikationen mellem maskinen og terminal. Nogle hints til fejlsøgning på terminaler ved forskellige problemer:

Symptom: Der skrives ikke på skærmen, når der tastes.

Årsag: Der er tastet <ctrl>+s
Løsning: Tast <Ctrl> + q.
Sluk og tænd igen for enheden.

Symptom: Der skrives ikke på skærmen, når der tastes.

Årsag: Fejl på stik eller ledninger.
Løsning: Check stik og ledninger for forbindelse/brud.
Skriv direkte ud til terminalen:
cat /etc/group > /dev/enhed (er der hul igennem?)

Symptom: Der skrives ikke på skærmen, når der tastes.

Årsag: En process har "spærret"/looper på porten
Løsning: Afbryd processerne på terminalen:
/etc/fuser -ku /dev/enhed.

Symptom: Der skrives ikke på skærmen, når der tastes.
Årsag: "Snavs" i portens input-buffer (sjælden)
Løsning: Prøv at rense porten med kommandoen:
cat < /dev/enhed.

Symptom: Tegn ser "forkerte" ud, taster reagerer forkert.
Årsag: Forkert setup i kommunikationen.
Løsning: Check enhedens setup
Check enhedens stty-parametre.

Symptom: Skærmen er lige tændt og der gives intet login-billede (serielle linier)
Årsag: ttymon portmonitor yder ingen service for porten eller portmonitoren er stoppet.
Løsning: Start portmonitoren eller opret en ny service til eksisterende portmonitor.

Symptom: Skærmen er lige tændt. Der gives startbillede (/etc/issue), men intet login-prompt (serielle linier).
Årsag: ttymon portmonitoren, der yder service for porten er disabled.
Løsning: Portmonitoren enables.

Symptom: Der forsøges rlogin ell. telnet fra anden maskine/terminal--server, men intet sker. Maskinen kan "ping'es". (ved netopkobling).
Årsag: Forkert konfiguration af inetd portmonitoren i /etc/inet/inetd.conf
Løsning: Stop inetd portmonitoren.
Ret konfigurationen i /etc/inet/inetd.conf
Start inetd portmonitoren igen.

Symptom: Udskrift på skærmen bliver placeret "forkert", piletaster virker ikke, funktionstaster virker ikke.
Årsag: TERM variablen har en forkert værdi, så der slås op på den forkerte konverteringstabel i terminfo databasen.
Løsning: Sæt TERM variablen til den rigtige værdi (husk at exportere TERM efterfølgende).

## 10.4 Styring af portmonitorer via sysadm.

Følgende kommandoer kan anvendes til administration af SAF:

**sacadm** Bruges til at konfigurere SAC, og definere hvilke portmonitorer, der skal startes. Konfigurationer gemmes i **/etc/saf/_sactab**.

**pmadm** Bruges til at konfigurere de enkelte services, som hver port monitor skal udføre. Konfigurationer til de enkelte services lagres i en fil med navnet **/etc/saf/<pmtag>/pmadmin**. <pmtag> er et slags alias, man selv giver sin portmonitor, når den oprettes med sacadm.

**ttyadm** Er en kommando, der kan bruges i forbindelse med pmadm til at sætte specifikke parametre for en service. Bruges fx til at sætte standard stty-parametre til serielle linier.

Det er dog lettere at andvende **sysadm ports** menuen:

```
1                Service Access Management
>port_monitors   - Port Monitor Management
 port_services   - Port Service Management
 quick_terminal  - Quick Terminal Setup
 tty_settings    - Terminal Line Setting Management
```

**Hvordan man opretter en ttymon gennem sysadm og tilknytter en service, der overvåger en seriel linie.**

Oprettelse af portmonitor ttymon (giver options til sacadm-kommandoen):

```
sysadm
   ports
      port_monitors
         add
            Port monitor tag:  <pmtag> Et slags alias, der
                               navngiver den specifikke port-
                               monitor, når den kører.
            Port monitor type: ttymon
            Command to start the port monitor:
```

/usr/lib/saf/ttymon
Version number: 2
Start port monitor immediately? 'yes' eller 'no'
Start state: 'ENABLED' eller '**DISABLED**'. Processen startes i begge tilfælde. Ved DISABLED kan den lytte på en port og udskrive en meddelelse om, hvorfor man ikke kan logge ind.
Restart count: <tal>, hvor mange gange skal der forsøges genstart, hvis port monitoren ikke starter.
File name of the port monitor configuration script: <filnavn> Konfigurationsfil, der kopieres til /etc/saf/<pmtag>/_config
Comments: Kommentar, der beskriver porten.
F3 (save)

Oprettelse af service, der overvåger en seriel linie (Page 1 definerer options til pmadm-kommandoen, page 2 definerer options til ttyadm-kommandoen, der giver værdier til pmadm's -m option):

port_services
add
add_to_one
<pmtag> (Givet i ovenstående)
Service tag: **<svctag>** Et navn, der kan identificere den specifikke service, fx tty00, hvis det drejer sig om overvågning af /dev/tty00.

Identification & authentication scheme:
login
Service invocation identity:
Port/service state: **DISABLED** (ENABLE, hvis servicen skal startes med det samme)
utmp entry to be created for this service?
yes
Version number: 2
File name for the port service configuration
script: Evt. navn på konfigurationsfil, der skal anvendes. Kopieres til /etc/saf/<pmtag>/<svctag>
Comments: Evt. kommentar til denne service.
Fx "Lytter på tty00".

F3 (save) Page 2:
Name of tty device: **/dev/tty01**
ttylabel: <indgang i /etc/ttydefs> (ofte 9600NP)
Service command: /usr/bin/shserv (giver en shell efter login)
Hangup: yes (hvis linien skal reinitialiseres efter hangup)
Connect-on-carrier: no ("yes" vil aktivere servicen uden at spørge om login først)
Bidirectional: no ("yes" hvis linien skal kunne bruges til at "ringe" ud på)
Wait-read: no ("yes" samt et <antal>, hvis linien skal modtage <antal> newlines før den skal reagere
Timeout: 0 (Et <antal>, hvis linien skal lægges, hvis der ikke er modtaget data i <antal> sekunder)
Prompt message: Fx "alex login:"
Modules to be pushed: ldterm
Disabled response message: (Meddelelse, der skal udskrives på linien, hvis den er DISABLED, og nogen alligevel forsøger at bruge den)

F3 (save)

# 105 Opgave

*Formål: - at foretage fejlsøgning på terminaler*
*Metode: - fra delopgave 2 løses opgaven* i grupper

1. Prøv at ændre hastigheden i terminalens set-up, og check hvilken virkning det har. Ret op på fejlen igen.

2. Gå sammen med din sidemand, og prøv nu at ændre hastigheden på den ene terminalport ved hjælp af stty-kommandoen. Se hvilken virkning det har. Ret terminalen op igen. NB anvend ovenstående hints.

3. Find en programfil. Prøv at inspicere den med cat eller pg. Hvis terminalen hænger, kan den afbrydes fra den anden terminal.

4. Prøv at stoppe en ttymon portmonitor, og forsøg derefter login fra en terminal, som portmonitoren overvåger. Start portmonitoren igen bagefter,

5. Prøv at disable en ttymon portmonitor, og forsøg derefter login fra en terminal, som portmonitoren overvåger. Husk at enable portmonitoren igen.

---

Ekstra:

6. Opret en ny ttymon portmonitor og tilknyt en service, der giver mulighed for login på en seriel port.

# 11 Printere

I UNIX SVR4.2 findes der tre muligheder for tilslutning af printere til en maskine:

- En printer kan være forbundet via en seriel linie
- Maskinen kan have adgang til en printer, der sidder direkte på et netværk eller på en terminalserver på et netværk (samme konfigurering). En sådan printer kaldes i UNIX SVR4.2 også for en "dial-up printer".
- Maskinen kan via et netværk få adgang til (serielle) printere på andre maskiner.

## 11.1 Betingelser for printere

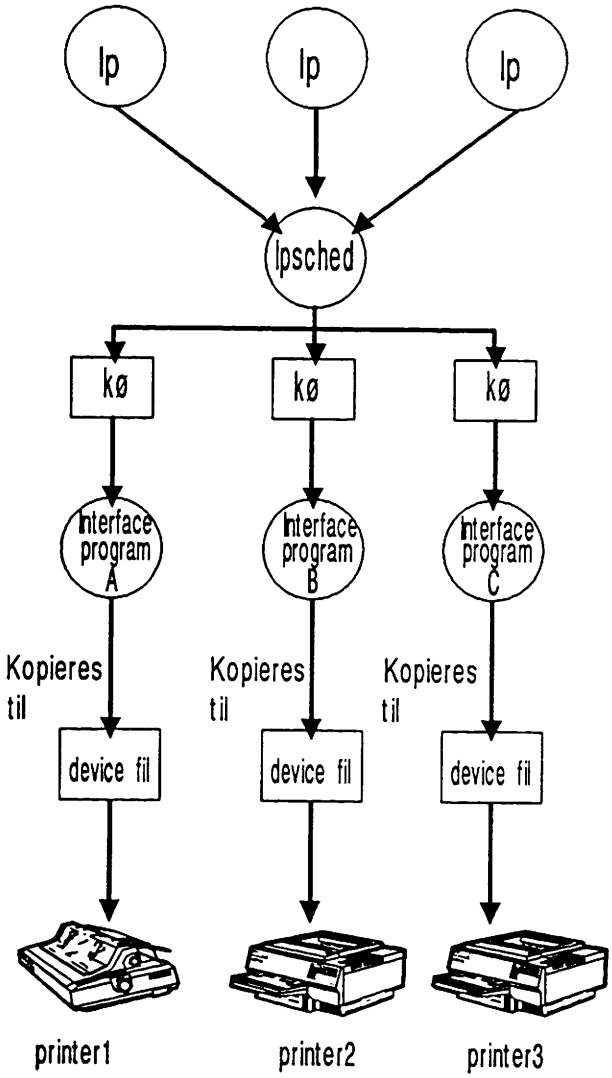
Ligesom for terminaler er der også en række betingelser, der skal være opfyldt for, at der kan skrives ud på en printer. For selve "linien" til printeren gælder følgende:

**For serielle printere gælder at:**

- Portens stty-parametre skal stemme overens med printerens setup
- Specialfilen for den serielle port skal have lp som ejer
- Rettighederne for specialfilen for porten skal være 600 (rx-------)

## 11.2 LP Print Service's opbygning

Printere er en af de ressourcer, der ikke kan deles. Det betyder at brugere, der vil skrive ud til samme printer, må vente på hinanden. Bl.a. for at undgå generende ventetider anbefales det altid at skrive ud gennem LP Print Service. LP Print Service er den facilitet i UNIX, der styrer udskrivning til printere, således at brugeren kan arbejde videre uanset om printeren er ledig. Udskriftsanmodninger bliver stillet i kø til den pågældende printer, hvorefter det er LP Print Service's ansvar at starte udskrivning, når printeren bliver ledig. LP Print Service's struktur fremgår af nedenstående skitse:



Figuren angiver, at der er oprettet et antal køer, der står i forbindelse med en række device-filer, hvortil udskriften sendes. En device fil kan i dette tilfælde også være et dial-up token fra /etc/uucp/Systems.

Printerkøer kan lukkes og åbnes efter behov. At en kø er åben (enabled) betyder, at der er forbindelse fra printerkøen til den fysiske printer. Hvis køen er lukket (disabled) kan der stadig lægges anmodninger i kø, men disse bliver i køen indtil denne igen åbnes.

Hvis en printer "rejecter" jobanmodninger kan der ikke lægges job i køen. Hvis en printkø "accepterer" jobanmodninger, kan der godt lægges job i køen og enable/disable status vil da være afgørende om der udskrives på printeren.

## 11.3 Udskrivning via LP Print Service

**lp <filnavn>** Udskriver <filnavn> på systemets default printer.

**lp -dprint0 <fil>** Udskriver <fil> på print0.

**cancel print0-12** Fjerner printjobbet print0-12 fra køen. Jobid - her print0-12 - findes med **sysadm printers/requets** eller **lpstat** kommandoen.

## 11.4 Administration af LP Print Service

Administration af spoolersystemet gennem **sysadm** vil i det følgende blive gennemgået.

Det første man altid skal gøre, er:

*** Vælg printers fra sysadm hovedmenu**

Herefter er der en del valgmuligheder. Udvalgte menu indgange med tilhørende skærmdialog vil blive gennemgået udfra den følgende skærmdialog, som kommer frem efter det første valg som vist ovenfor.

```
   2  Line Printer Services Configuration and Operation
   >classes     - Manage Classes of Related Printers
    filters     - Manage Filters for Special Processing
    forms       - Manage Pre-Printed Forms
--> operations  - Perform Daily Printer Service Operations
    printers    - Configure Printers for the Printer Service
--> priorities  - Assign Print Queue Priorities to Users
--> requests    - Manage Active Print Requests
--> status      - Display Status of Printer Service
    systems     - Configure Connections to Remote Systems
```

De med pil markerede indgange vil blive gennemgået!

**Status**

**printers** Kan vise om printere er disabled/enabled.

**requests** Kan vise om der ligger printjobs i kø.

**Requests**

Skærmdialogen ser ud som følger:

```
3              Manage Active Print Requests
>cancel   - Cancel Print Requests
 hold     - Place Pending Print Requests on Hold
 move     - Move print Requests to a New Destination
 release  - Release Held Print Requests
```

Alle indgange i skærmdialogen efterfølges af endnu en skærmdialog, hvor yderligere information skal indtastes. HELP og CHOICES kan med fordel bruges.

**Priorities**

Denne indgang giver mulighed for, at brugere tildeles prioritet mht. udskrivning. Standardværdien er 20, og kan ligge i intervallet 0-39. En bruger med et printjob med et lavt tal bliver udskrevet før en bruger med et job, som har mindre prioritet (større tal).

**Operations**

Skærmdialogen ser ud som følger:

```
3                Perform Daily Printer Service Operations
>accept       - Allow Class(es) and/or Printer(s) to Accept Print
                Requests
 control      - Stop the Printer Service
 disable      - Disable Printer from Printing
 enable       - Enable Printer for Printing
 lp_copy      - Set Lp to Always Copy Print Requests Files
 mount        - Mount a Form or Font On a Printer
 reject       - Stop Printer from Accepting Print Requests
 set default  - Set Default Print Destination
 unmount      - Unmount a Form or Printerwheel from a Printer
```

Bemærk, at indgangen control, som stopper og starter spooleren, kan "toggle's". Dvs. at vælges indgangen control, hvorved spooleren stoppes, vil indgangen efterfølgende kunne starte spooleren igen. Hvis spooleren stoppes, vil de øvrige indgange blive blokeret, hvilket visuelt resulterer i en "shading" af indgangene.

**UNIX kommando**

Ude fra shellen kan man give kommandoen **/usr/bin/lpstat** med option **-t**. Med option -t gives en fuld oversigt over status for LP Print Service og de forskellige printerkøer. Kommandoen kan bruges med andre options, der udtager forskellige delmængder af de totale oplysninger.

## 11.5 Fejlfinding på printere

I lighed med terminaler findes der også nogle tips til fejlfinding på printere, der af en eller anden grund ikke vil skrive ud. For printere må fejlsøgning tage udgangspunkt i først at diagnosticere om problemet ligger i LP Print Service eller på linien til printeren.

**Er problemet i LP Print Service?**

1. Skriv direkte ud til printeren:
   seriel linie: cat /etc/group > <device fil>

   Skrives filen ud, ligger fejlen sandsynligvis i LP Print Service. (Der fortsættes med punkt 2 ellers hoppes ned til "Fejl på linien eller printeren").

   dial-up linie: telnet til terminalserverens IP-adresse/-portnummer (angiv evt. hostnavn istedet for IP-adresse). Eks.

   ```
   $ telnet ntc 2060
   ```

   Skriv lidt tekst og terminér telnet. Tryk formfeed på printeren. Hvis teksten kommer ud er linien i orden.

2. Check om spooleren kører.

3. Check at køen er 'enabled' og 'accepting requests'.

4. Check at alle filer under /etc/lp er ejet af lp.

**Fejl på linien eller printeren:**

1. Sluk og tænd for printeren.

2. Check stik og ledninger for forbindelse/brud.

3. Afbryd eventuelle processer på printeren:
   /etc/fuser -ku <device fil>

4. Prøv at rense porten med kommandoen:
   cat < /dev/enhed.

5. Check printerens statusværdier (på printerens display).

6. Sammenlign værdierne med stty-parametrene.

Ved dial-up printere:

7. Check om der er de rigtige konfigurationer i /etc/uucp/Systems og /etc/uucp/Devices.

# 11.6 Opgave

*Formål:* - *at anvende LP print Service*
- *at opnå fortrolighed med at fortolke uddata fra lpstat -t kommandoen og sysadm printers*

*Metode:* - *fra delopgave 4 løses opgaven samlet*

1. **Undersøg om LP Print Service kører.**

2. **Prøv at udskrive en fil til en printerkø. Kønavn opgives af instruktøren.**

3. **Prøv at få en statusoversigt over spooleren. Brug sysadm eller lpstat kommandoen med forskellige options.(Se Reference Manualen).**

4. **Disable en printerkø. Prøv at skrive til køen. Ligger der noget i den? Kommer det ud på printeren?**

5. **Enable køen igen. Kommer der noget ud på printeren nu?**

6. **Reject en printerkø. Prøv at skrive til køen. Hvad sker der? Sæt printkøen til at acceptere jobanmodninger igen.**

7. **Stop LP Print Service. Prøv at skrive en fil ud på printeren. Hvad sker der? Start spooleren igen.**

# 12 Elektroniske meddelelser

Der findes flere værktøjer, der giver brugerne mulighed for at kommunikere med hinanden, ligesom systemadministratoren kan bruge mulighederne til at henvende sig til brugerne.

Nogle værktøjer er interaktive, andre er hurtige. Valget af kommunikationsform bør bero på informationen, som ønskes afsendt.

Hvis fx systemet skal lukkes på et tidspunkt, hvor det normalt er i drift, bør brugerne orienteres. Et eller flere værktøjer kan bruges til at informere med. Valget af værktøjer afhænger af i hvor god tid, før en nedlukning, man er i stand til at informere brugerne.

## 12.1 finger

UNIX kommandoen **/usr/bin/finger** giver information om en bruger. Uden argumenter giver **finger** information om én selv. Angives et eller flere logins, får man information om de enkelte brugere.

Syntaksen er:

```
$ /usr/bin/finger brugernavn(e)
```

Indholdet af de to filer, **~/.project** (kun første linie vises) og **~/.plan**, vises af **finger**.

Kommandoen **/usr/bin/finger** skal bruges koordineret af en organisation for at være et godt kommunikationsværktøj.

## 12.2 mail

En besked sendt med **/usr/bin/mail** lægges i modtagernes elektroniske postkasse. Næste gang brugerne logger ind, vil der komme en meddelelse om, at der er indgået post. **mail** egner sig derfor bedst til beskeder, der ikke haster.

En besked sendes således:

*$* **/usr/bin/mail [options] brugernavn(e)**

**mail** læser normalt fra standard input. Det er imidlertid også muligt at forberede post i en fil og så sende denne. Syntaksen er:

*$* **/usr/bin/mail [options] brugernavn(e) < <filnavn>**

Kommmandofortolkeren har en standard shell variabel der hedder **MAILCHECK. MAILCHECK** angiver hvor ofte, det skal undersøges, om der er post (målt i antal sekunder). Default er 600 sekunder, altså hvert 10. minut.

Skal man læse sin post skriver man blot

*$* **/usr/bin/mail**

Et **?** giver en oversigt over mulige kommandoer.

Et alternativ til **/usr/bin/mail** er **/usr/bin/mailx. mailx** har samme formål som **mail**, men indeholder en række muligheder som blandt andet at svare på et brev m.m.

## 12.3 write

Kommandoen **/usr/bin/write** kan bruges til at sende beskeder til indloggede brugere. Syntaksen er den samme som beskrevet for **mail**, idet der kan skrives en besked direkte fra terminalen eller der kan sendes en fil.

## 12.4 talk

UNIX kommandoen **/usr/bin/talk** kan bruges ved interaktiv kommunikation mellem to brugere. Syntaksen er den samme som beskrevet for **/usr/bin/mail** og **/usr/bin/write**, bortset fra at kun to brugere kan tale sammen af gangen.

## 12.5 /var/motd

Indholdet af filen **/var/motd** udskrives på alle skærme hver gang der logges på. **/var/motd kan** altså passende bruges, hvis man vil henvende sig til alle brugere.

**/var/motd** er en tekstfil, som den systemansvarlige kan skrive i fx med **vi** editoren.

## 12.6 news

Hvis man mener, at det er nok at en bruger læser en besked een gang, er **/usr/bin/news** det rigtige værktøj. Hver besked skal skrives i en fil, der ligger i kataloget **/var/news**.

Når en bruger logger på, gives en meddelelse om, at der er nyheder. Nyhederne kan da læses ved i shell blot at skrive

```
$ /usr/bin/news -a
```

Når en bruger har læst de nyheder, der ligger, vil der i brugerens

```
~/.news time
```

fil blive registreret dato og klokkeslet for, hvornår nyhederne blev læst.

Denne fil holder styr på, om brugeren har læst de sidste nyheder.

## 12.7 wall

Hvis man meget hurtigt skal informere alle brugere, der er logget på, kan **/usr/sbin/wall** bruges. **wall** virker som en **write** til alle aktive brugere.

**wall** bruges blandt andet af kommandoen **/usr/sbin/shutdown** i forbindelse med en nedlukning og den kan kun afgives af superbrugeren.

## 12.8 mesg

UNIX kommandoen **/usr/bin/mesg** bruges til at forhindre andre i at kunne kommunikere med én. Ved at skrive

```
$ /usr/bin/mesg -n
```

kan hverken **/usr/bin/write** eller **/usr/bin/talk** bruges af andre til at komme i kontakt med én.

Med

```
$ /usr/bin/mesg -y
```

kan man åbne op for kommunikationen igen.

Bruges **mesg** uden option, får man oplyst, om man kan modtage kommunikation med **write** eller **talk**.

# 12.9 Opgave

*Formål: - at sende besked til brugerne på systemet*

1. **Overvej hvilket værktøj du vil bruge til at informere brugerne i følgende situationer:**

   a. **Du har aftalt med DDE's tekniker at han kommer om 14 dage og installerer en ny disk.**

   b. **I morgen kommer teknikeren med en ny BAIO, der skal installeres.**

   c. **Du har lige installeret et nyt program, der kun skal bruges af et par få mennesker.**

   d. **Det er sent, og du skal foretage en total backup af systemet. Du tror, at Anders And har glemt at logge ud.**

   e. **På grund af større fejlrettelser bliver du nødt til at boote SUPERMAX'en. Hvordan vil du informere dine brugere om at maskinen skal lukkes ned nu.**

2. **Afprøv de forskellige meddelelsesmetoder.**

# 13 Nedlukning og opstart

Det er ind imellem nødvendigt at lukke en UNIX maskine ned og starte den op igen. Årsagen kan være rutinemæssig som backup eller installation af programmer, men nedlukningen kan også skyldes fejl eller omkonfigurering. I det følgende forklares hvad der sker under nedlukning og opstart, og der gennemgås forskellige metoder.

## 13.1 Driftsniveauer

En UNIX maskine kan befinde sig på forskellige driftsniveauer. Undertiden anvendes også følgende betegnelser for "driftsniveau": Run-level, run-mode, init-state og init.

Kommandoen **/sbin/init** anvendes til skift af driftsniveauer. Følgende gives en kort forklaring på, hvad der sker ved de enkelte init-kommandoer:

**init 0** Fuld shutdown. Scripts i kataloget **/etc/rc0.d** udføres.

**init 1** Enkeltbruger tilstand. Alle filsystemer er mountet, men kun de mest nødvendige processer er startet. Scripts i **/etc/rc1.d** udføres.

**init 2** Flerbruger tilstand. Scripts i **/etc/rc2.d** og **/etc/dinit.d** udføres. Brugere kan logge på både serielt og via net. Dette styres af SAC (Service Access Controller), der kaldes fra **/etc/inittab** ummiddelbart inden scripts i **/etc/dinit.d.**

**init 3** Netværks tilstand. Ressourcer der via nettet skal stilles til rådighed for andre maskiner startes. Dvs. NFS (Network File System) og RFS (Remote File Sharing). Scripts i **/etc/rc2.d**, **/etc/rc3.d** og **/etc/dinit.d** udføres.

**init 5** Firmware mode.

**init 6** Genstart til driftsniveau angivet i **/etc/inittab** (initde-

fault). Scripts i **/etc/rc0.d** udføres, hvorefter der bootes normalt.

**init s** Virker forskelligt afhængigt af om der er bootet helt init S fra start med "s" som initdefault (eller ved init 6) eller om init s er givet fra et højere driftsniveau. Ved "s" som initdefault vil kun en begrænset del af slicene være mountet og kun de mest nødvendige processer vil være aktive.

Ved "init s" fra højere driftsniveau vil processer, der ikke skal køre i flerbrugertilstand være stoppet. Diske vil stadig være mountet og øvrige processer vil stadig køre.

## 13.2 Kommandoer og kataloger til nedlukning og opstart

Via oplysninger fra **/etc/inittab** udføres **/sbin/rc0**, **/sbin/rc1**, **/sbin/rc2** og **/sbin/rc3** ved hhv. **/sbin/init 0,1,2** eller **3**. **/sbin/init 5** og **6** udfører **/sbin/rc0**. De forskellige rc-kommandoer kalder scripts i en række tilsvarende rc-kataloger under **/etc**. Fx vil en **/sbin/init 2** medføre kald af **/sbin/rc2** som kalder scripts i kataloget **/etc/rc2.d**

Når man udfører kommandoen

# init n (hvor n er lig 0,1,2 eller 3)

bliver scripts i det tilsvarende **/etc/rcn.d** således udført.

Scripts med navne som 'K##<name>' bliver udført for at standse processer. Man kan også kalde dem "kill-scripts".

Scripts med navne som 'S##<name>' bliver udført for at starte processer. Man kan også kalde dem "start-scripts":

Hvis n er lig 2 eller 3 bliver scripts i **/etc/dinit.d** med tilsvarende navne også udført.

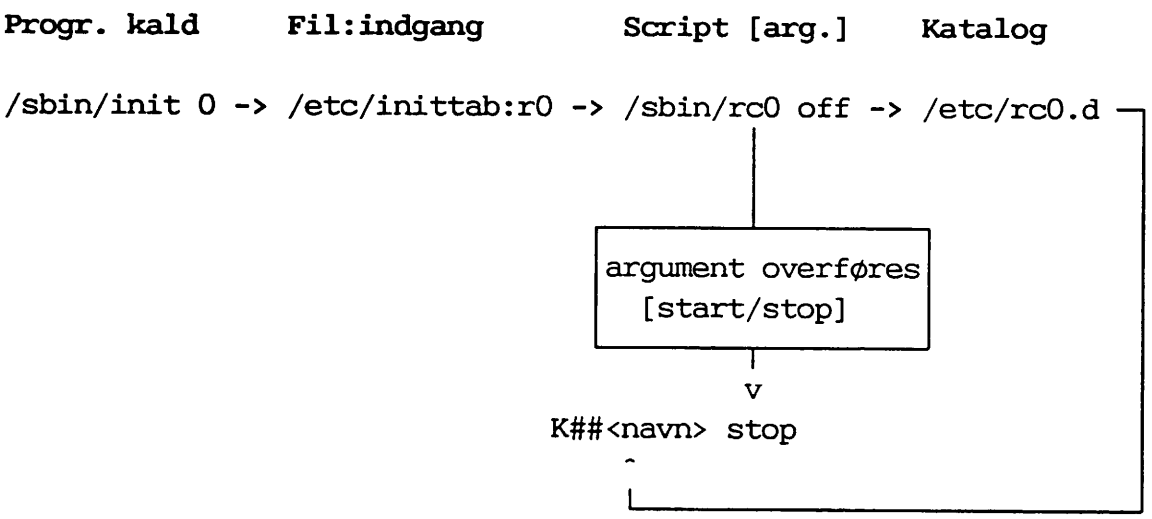
Alle S- eller K-scripts i ovennævnte kataloger er linket til scripts i **/etc/init.d** med navnet <name> (i henhold til navnet på S- eller K-scriptet).

Det samme script i **/etc/init.d** kan have både et S- og et K-script linket fra et af de ovennævnte kataloger. Således vil de samme scripts både tage sig af start og stop af processer. Dette gøres ved at disse scripts checker første argument, som de blev kaldt med.

Kald af S- og K-scripts udføres af **/sbin/rc0**, **/sbin/rc1**, **/sbin/rc2** eller **/sbin/rc3**, der leder **/etc/rc**n**.d** katalogerne igennem. Hvis et script her begynder med "S" vil "start" blive brugt som første argument, hvis scriptet begynder med "K" vil "stop" blive brugt som første argument.

## 13.3 Oversigt over nedlukning

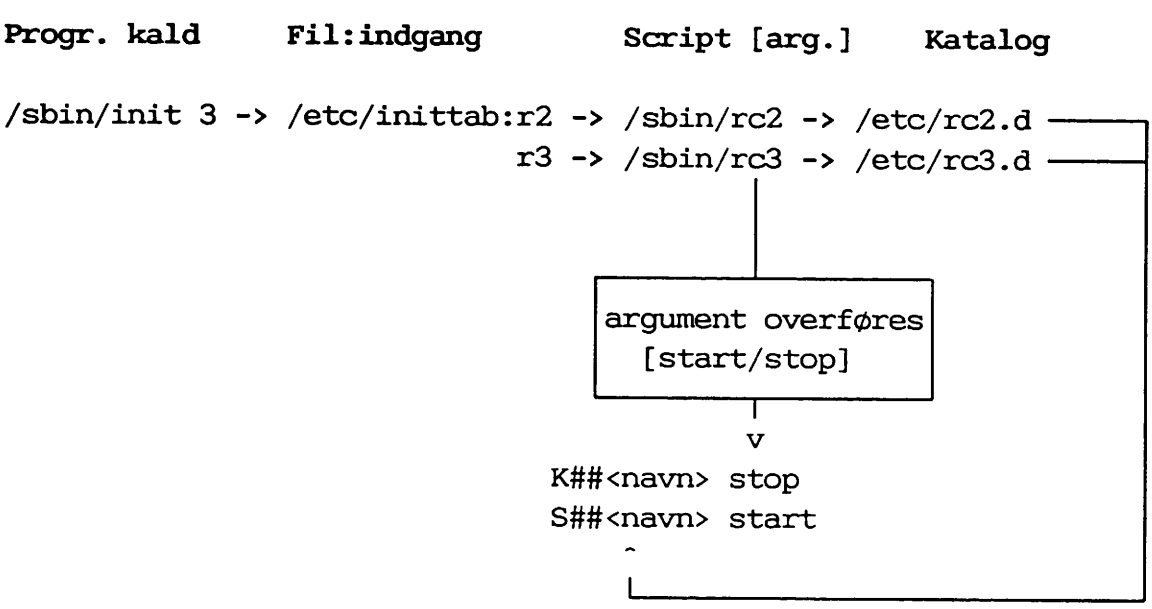
Ved nedlukning kan forløbet kort skitseres således:




Først gives UNIX kommandoen **/sbin/init 0**. Denne kommando går ned i filen **/etc/inittab** og udføre linien med indgangen **'r0'**. Denne linie indeholder et kald af scriptet **/sbin/rc0 off**. Dette script går ned i kataloget **/etc/rc0.d**, hvor alle scripts eksekveres med et af argumenterne start eller stop. Eftersom maskinen lukkes ned, ligger der kun "kill" scripts her.

## 13.4 Oversigt over opstart

Ved opstart kan forløbet kort skitseres således:




Først gives UNIX kommandoen **/sbin/init 3**. Denne kommando går ned i filen **/etc/inittab** og udføre linierne med indgangene **'r2'** og **'r3'**. Disse linier indeholder kald af scriptene **/sbin/rc2** og **/sbin/rc3**. Disse scripts går ned i katalogerne **/etc/rc2.d** og **/etc/rc3.d**, hvor alle scripts eksekveres med et af argumenterne start eller stop. Eftersom maskinen startes op, udføres både "kill" og "start" scripts.

## 13.5 Opbygning af boot script

Et boot script skal kunne håndtere en parameter, som angivet ovenfor. En parameter i et UNIX shell script angives med et dollar tegn ($) efterfulgt af et nummer mellem et og ni (1-9).

Et eksempel på et boot script kunne være:

```
$ cat /etc/init.d/kursus

# Tue Jan  2 15:52:31 MET 1996
# script til brug i dde kursus
case "$1" in
'start')
     echo `who -r` ": start" >> /kursus.init
     ;;
'stop')
     echo `who -r` ": stop" >> /kursus.init
     ;;
*)
     echo "Brug (start/stop)"
     ;;
esac
```

Når der foretages et skift til fx driftsniveau 2, skriver dette script tidspunktet ned i en fil ved navn **/kursus.init**.

## 13.6 Linkning af boot scripts

Efter at boot scriptet er lavet og lagt i enten **/etc/init.d** eller **/etc/dinit.d** kataloget, skal der foretages følgende to ting, som følger eksemplet ovenfor:

```
1) chmod u+x /etc/init.d/kursus
2) ln /etc/init.d/kursus /rc2.d/K11kursus
   ln /etc/init.d/kursus /rc3.d/S11kursus
```

Det første sikrer, at scriptet kan eksekveres. Det andet linker scriptet til de respektive kataloger, hvor det skal udføres. I dette tilfælde udføres et "kill" i driftsniveau 2 og en "start" i driftsniveau 3.

## 13.7 /etc/inittab

Når init kommandoen kaldes fører dette til opslag i filen **/etc/inittab**, der angiver, hvad der skal ske ved de forskellige init-kommandoer. Herunder kald af de relevante rc-kommandoer.

Ændringer i **/etc/inittab** må ikke foretages direkte i filen, da denne kan blive rebuildet i forbindelse med boot og derved kan ændringerne gå tabt. I stedet foretages ændringer i **/etc/inittab** på følgende måde:

- Foretag de nødvendige ændringer i **/etc/conf/init.d/kernel**, som har samme format som **/etc/inittab**.

- Udfør **/etc/conf/bin/idmkinit**, som opbygger en ny inittab, der lægges som **/etc/conf/cf.d/inittab**.

**Eksempel på /etc/inittab:**

```
# WARNING: THIS FILE IS AUTOMATICALLY GENERATED.
# Any changes made directly to this file may be overwritten
# at the next system reboot.
# Permanent changes should also be made to files in the
# /etc/conf/init.d directory.
# See Init(4) and idmkinit(1M) for more information.
#
swap::sysinit:/sbin/swap -a /dev/dsk/0s2 >/dev/sysmsg 2>&1
cr::sysinit:/sbin/ckroot >/dev/sysmsg 2>&1
mi::sysinit:/sbin/sh -c '[ -x /sbin/macinit ] && /sbin/macinit'...
ck::sysinit:/sbin/setclk >/dev/sysmsg 2>&1
mm::sysinit:/etc/conf/bin/idmodreg -c `/etc/conf/bin/idkname -c...
ldmd::sysinit:/etc/conf/bin/idmodload >/dev/sysmsg 2>&1
ap::sysinit:/sbin/autopush -f /etc/ap/chan.ap
ak::sysinit:/sbin/wsinit >/etc/wsinit.err 2>&1
bchk::sysinit:/sbin/bcheckrc </dev/console >/dev/sysmsg 2>&1
key::sysinit:/sbin/sh -c '[ -x /sbin/keyadm ] && /sbin/keyadm -...
onl::sysinit:/sbin/psradm -n -a
bu::sysinit:/etc/conf/bin/idrebuild reboot </dev/console >/dev/...
me::sysinit:/etc/conf/bin/idmkenv >/dev/sysmsg 2>&1
xdc::sysinit:/sbin/sh -c 'if [ -x /etc/rc.d/es-setup ] ; then /...
ia::sysinit:/sbin/creatiadb </dev/console >/dev/sysmsg 2>&1
metr::sysinit:/sbin/sh -c '[ -x /sbin/metreg ] && /sbin/metreg'...
is:3:initdefault:
bd:56:wait:/etc/conf/bin/idrebuild </dev/console >/dev/sysmsg 2>&1
```

```
r0:0:wait:/sbin/rc0 off >/dev/sysmsg 2>&1 </dev/console
r1:1:wait:/sbin/rc1 >/dev/sysmsg 2>&1 </dev/console
r2:23:wait:/sbin/rc2 >/dev/sysmsg 2>&1 </dev/console
r3:3:wait:/sbin/rc3  >/dev/sysmsg 2>&1 </dev/console
r5:5:wait:/sbin/rc0 firm >/dev/sysmsg 2>&1 </dev/console
r6:6:wait:/sbin/rc0 reboot >/dev/sysmsg 2>&1 </dev/console
sd:0:wait:/sbin/uadmin 2 0 >/dev/sysmsg 2>&1 </dev/console
fw:5:wait:/sbin/uadmin 2 2 >/dev/sysmsg 2>&1 </dev/console
rb:6:wait:/sbin/uadmin 2 1 >/dev/sysmsg 2>&1 </dev/console
li:23:wait:/usr/bin/ln /dev/systty /dev/syscon >/dev/null 2>&1
sc:234:respawn:/usr/lib/saf/sac -t 300
co:12345:respawn:/usr/lib/saf/ttymon -g -v -p "Console Login: ...
d2:23:wait:/sbin/dinit >/dev/sysmsg 2>&1 </dev/console
┬ ┬ ──┬ ──────────────────┬──────────────────
a b   c                   d
```

a. Entydig identifikation af en linie

b. Angiver ved hvilke driftsniveauer kommandoen skal udføres

c. Kommandomåde:

| | |
|---|---|
| off | Linien bliver ikke udført. |
| respawn | Hvis processen ikke eksisterer startes den. Hvis den dør, bliver den automatisk startet. |
| wait | Ved skift til andet driftsniveau, startes processen og venter til den er færdig. |

Disse tre kommandomåder er de vigtigste at kende til.

d. Et program eller shell-script.

# 13.8 Opgave

*Formål: At skifte driftsniveau og se hvornår et script i /etc/rc2.d kaldes. Opgaven løses samlet fra konsollen.*

1. Opret et script i /etc/init.d, der udskriver en meddelelse om, at scriptet er blevet kaldt. Link scriptet til /etc/rc2.d/S10<scriptets navn>.

2. Udfør en **init 2** kommando og se om scriptet bliver kaldt.

3. Udfør en **init s** kommando. Hvilke processer kører der herefter? Hvilke diske er mountet?

4. Udfør en **init 3** kommando. Bliver scriptet fra før kaldt?

5. Udfør en *init 1*. Hvilke processer kører nu? Hvilke diske er mountet?

6. Udfør en **init 3**.

# 14 Backup

En af de vigtigste forudsætninger for et driftsikkert EDB-system er, at man altid har mulighed for at genetablere systemet ved systemnedbrud. Et sådant systemnedbrud kan fx skyldes fejl på harddisken, men i de fleste tilfælde skyldes det menneskelige fejl: dataområder der slettes ved en fejltagelse, filer som overskrives af andre filer m.m. For at kunne genetablere systemet således at mindst mulig data går tabt, er det nødvendigt, at man lægger en backupstrategi, der passer til ens system.

I den forbindelse skal følgende overvejes:

**1) Hvad skal kopieres?**

**2) Hvor ofte skal kopiering foretages?**

**3) Hvordan skal kopiering foretages?**

ad. 1 Der skal foreligge en backup af alt, hvad der ligger på maskinen. De områder hvor data udskiftes mest dynamisk, dvs. bruger- og databasediske bør kopieres meget hyppigt, medens områder med statiske data ikke skal kopieres så ofte.

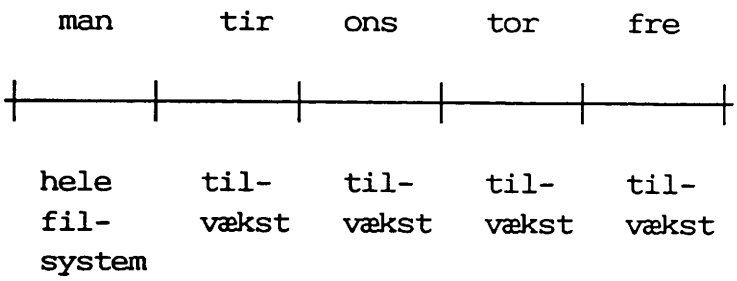
ad. 2 Der bør tages en totalbackup af systemet med passende mellemrum, eksempelvis en gang om måneden og efter større tilretninger og installationer.

Bruger- og databasediske bør kopieres hyppigere, eksempelvis dagligt eller et par gange om ugen afhængig af hvor meget, der arbejdes på systemet.

ad. 3 En totalbackup bør omfatte:

* diskbackup af bootområdet
* diskbackup af databasediske (dbs og redologdiske) og kontrolfiler
* filbackup af filsystemer på maskinen
* backup af roden

Kopieringen af dynamiske data kan organiseres som kopiering af hele filsystemer suppleret med tilvækstbackup'er af de samme filsystemer. Kopieringen kan fx forløbe efter følgende cyklus:




**MAN BØR ALTID KONTROLLÆSE SIN BACKUP!!**

Man har hørt grufulde historier om installationer, der har kørt backup i flere år på SKRIVEBESKYTTEDE bånd. Derudover slides bånd betydeligt ved brug. Bånd bør IKKE bruges mere end 20-30 gange førend de udfra et sikkerhedsmæssigt synspunkt er slidt op.

Når man har fastlagt sin strategi, vil man opdage, at der er utallige muligheder for at foretage kopiering ud på disketter og bånd. Generelt kan det dog siges, at UNIX skelner mellem sikkerhedskopiering af filer/filsystemer og diske.

## 14.1 Forberedelse til backup - datasæt

I sysadm menuen er det muligt at definere forskellige datasæt, der hver kan løse en kopieringsopgave. Et datasæt består af et antal subdiske, som skal kopieres samlet. Eksempler på datasæt kunne være:

* **brugerdisk(e)** (dynamiske data - filer)
* **programdisk(e)** (statiske data - filer)
* **databasediske m. kontrolfiler** (rå diske + filer)
* **bootdisken** (rå disk)
* **alle ovenstående 4 typer til totalbackup af maskinen**

Der bør eksistere datasæt til at klare totalbackuppen og den jævnlige backup af brugerdata. Datasættene defineres en gang for alle fra **sysadm** menuen.

* **Vælg ddebkup fra sysadm hovedmenu**
* **Vælg setconf fra s Supermax backup system**
* **Vælg add fra 3 Setup the backup configuration**

```
3                Setup the backup configuration
>add     -       Add a dataset to the backup system
 delete  -       Delete a dataset from the backup system
 modify  -       Modify a dataset in the backup system
 show    -       Show the backup configuration
```

Herefter kan man indtaste nogle dataset.

```
4              Add dataset to the backup system

New dataset: bruger

FIles:       /dev/dsk/c0t0d0s3
             /dev/dsk/c0t0d0s4
             /dev/dsk/c0t0d0s5
             /dev/dsk/c0t0d0sc
             /bruger/unix1/fil
```

## 14.2 Forberedelse til backup - labels på bånd

Under forberedelsen til backup skal man også sætte labels på sine bånd. Inden kopieringen startes, vil labelens navn blive opgivet og man får mulighed for at fortryde kopieringen. Denne facilitet skal sikre, at man ikke ved en fejltagelse får overskrevet et bånd. Der sættes labels på båndene en gang for alle således:

* **Vælg ddebkup fra sysadm hovedmenu**
* **Vælg tapemgmt fra 2 Supermax backup system**
* **Vælg putlabel fra Manage tape information**

Ved den efterfølgende skærmdialog skal man opgive labelens navn og det drev hvor båndet befinder sig.

## 14.3 4 typer backup med datasæt

Når et datasæt er defineret, kan det i princippet benyttes til at tage 4 forskellige typer backup:

* **filbackup**
* **diskbackup**
* **tilvækstbackup**
* **blandet fil- og diskbackup - mixback**

Det bør bemærkes, at kun datasæt der refererer til subdiske med filsystemer kan bruges til alle 4 typer backup. Det er således ikke muligt at tage filbackup af databasediske og bootdisken, da data på disse diske ikke er organiseret i filer.

## 14.4 Totalbackup - filsystemer og logiske diske

En totalbackup indbefatter filbackup af alle subdiske, der indgår i filsystemet samt backup af databasen og bootdisk(e). Hvis der på systemet befinder sig et videodrev, vil en totalbackup med fordel kunne tages som en 'mixback' af ét datasæt, hvor alle relevante subdiske indgår. Er backupmediet en streamer, vil en totalbackup normalt sprede sig over adskillige bånd (og datasæt), hvor der ligger hhv. filsystemer og rå subdiske.

Der bør under alle omstændigheder være så lidt aktivitet som muligt på systemet, når denne backup foretages, hvorfor en totalbackup normalt tages i enkeltbruger tilstand.

Man får systemet i enkeltbruger tilstand ved kommandoen:

```
# init 1
```

Backuppen kan nu udføres fx som vist nedenfor, hvor der startes en totalbackup som en 'mixback':

*** Vælg ddebkup fra sysadm hovedmenu**

*** Vælg backup fra 2 Supermax backup system**

*** Vælg mixback fra 3 Backup dataset**

I den efterfølgende skærmdialog vælger man det relevante datasæt fra listen over tilgængelige sæt. Der vælges også backup medium.

Kopiering af filsystemer sker med UNIX kommandoen **/usr/bin/cpio** - kopiering af logiske diske med kommandoen **/usr/dde/bin/dskback** (DDE specifik kommando), der tager en image kopi af hele disken.

# 14.5 Opgave

*Formål:* - *at lægge en backup strategi*
- *at oprette datasæt*
- *at sætte labels på bånd*

*Metode:* - *delopgave 2 og 3 løses opgaven samlet ved konsollen*

1. **Undersøg konfigurationen på kursusmaskinen for at bestemme hvad en totalbackup skal omfatte.**

2. **Definer et datasæt til brug for en totalbackup af maskinen.**

3. **Definer et datasæt til brug for en backup af brugerdata.**

4. **Prøv at sætte en label med et passende navn på et bånd.**

## 14.6 Jævnlig backup af brugerdata

Den jævnlige backup af alle brugerdata sættes op til at køre automatisk på et tidspunkt, hvor der er lidt aktivitet på systemet. Dette gøres en gang for alle således:

* **Vælg ddebkup fra sysadm hovedmenu**
* **Vælg setplan fra 2 Supermax backup system**
* **Vælg add fra 3 Setup the automatic backup procedure**

Herefter indtastes de nødvendige oplysninger i felterne.

```
4      Add command to the backup plan

Month(s):          all
Day(s):            1-5
Time:              22:00

Backup command:
Dataset(s):        All-valid-datasets

Label:
Device:            ctape1
```

Brug **F2 (CHOICES)**, hvis information om de enkelte felter ønskes. Fx får man under feltet '**Backup command**' følgende muligheder frem:

```
5 Choices
>diskback
 incrback
 totalback
 mixback
```

## 14.7 Check af automatiske backup procedure

Man kan checke de automatiske backup procedure i sysadm på følgende måde:

* **Vælg ddebkup fra sysadm hovedmenu**

* **Vælg bkplan fra 2 Supermax backup system**

```
3                Show the automatic backup procedure

Current backup plan:

Day / Date / Month / Start / Type / Drive / Label - Data

1-5  *  * 22:00  mixback  ctape1 "man-1" - bruger
```

En stjerne (*) markerer, at alle gyldige værdier er valgt. For fx måneder vil det betyde, at alle måneder fra januar til december er valgt.

## 14.8 Tilvækstbackup af brugerdata

En tilvækstbackup består alene af filer, der er ændret siden sidste backuptidspunkt. Denne form for backup supplerer backuppen af alle brugerdata og tages de resterende dage. Rutinen kan sættes op til at køre automatisk. Dette gøres ved i **sysadm** at gå ned i **'ddebkup setplan add'** og i feltet **'Backup command'** at vælge **"incrback"**, ligesom man naturligvis skal angive korrekte ugedage i forhold til den fuldstændige backup.

## 14.9 Jævnlig backup af Oracle databasen

Det er muligt at tage backup af Oracle databasen på flere forskellige måder. Fra **sysadm** menuen er det muligt at definere en **'mixkopiering'** af de størrelser, der indgår i databasen, dvs. en diskkopiering af redologdiskene, databasedisken(e) samt en filkopi af Oracles kontrolfiler. Når et datasæt er oprettet med disse

størrelser, defineres backuppen i '**sysadm ddebkup setplan add**' hvor der i feltet '**Backup command**' vælges "**mixback**" på de ønskede tidspunkter.

Databasen skal være lukket ned før det er muligt at tage '**mixback**' af den.

## 14.10 Indlæsning af sikkerhedskopi

Når man skal indlæse en sikkerhedskopi er det vigtigt at vide, hvordan kopien er taget. Sikkerhedskopier taget som diskbackup skal også kopieres ind fra menupunktet '**diskrestore**' under '**sysadm ddebkup restore**'.

Skærmdialogens valgmuligheder er vist her:

```
3                          Restore data
>diskrestore - Restore a rawdisk from a removable medium
 filerestore - Restore files and directories from removable media
 findtape    - Find tapes that contains the backup data
 mixrestore  - Restore raw disks, files and directories from
               removable media
```

Ligeledes skal man være opmærksom på, at enkelte filer kun kan indlæses fra kopier taget som filbackup, idet filerne ved denne form for kopiering bliver organiseret filvis på båndet.

# 14.11 Opgave

*Formål:* - *bestemme fremgangsmåde for backup af brugerdata*
- *at sætte backup af brugerdata til at køre automatisk*
- *at tage backup*

*Metode:* - *opgaven løses samlet ved konsollen*

1. **Findes der et datasæt til brug for backup af brugerdata?**

2. **Hvor ofte vil du tage backup af alle brugerdata hhv. tilvækstbackup? På hvilket tidspunkt?**

3. **Definer en backupplan således at backuppen kører automatisk på de tidspunkter du bestemte i forrige delopgave.**

4. **Definer endnu en backupplan, der starter totalbackup af brugerdata om 5 minutter.**

# 15 Programadministration

Installation (og afinstallation) af programmer foretages nemmest vha. **sysadm** menupunktet **software**. Drejer det sig om **"carrier bånd"** fra DDE, dvs. software, som kun kan installeres med licensnøgle, skal man anvende **'sysadm software'**. Dette kapitel vil derfor tage udgangspunkt i brugen af sysadm.

Der findes dog også en række UNIX kommandoer, der kan anvendes til installation af software. Der vil blive givet en oversigt over disse UNIX kommandoer sidst i kapitlet.

Programmer kan leveres i form af et **"set"** eller en **"package"**. En **package** er et eller flere programmer, filer eller scripts. Tilsammen, hvis der er tale om flere programmer, udgør de en helhed, der er nødvendig for installation af en ny facilitet på maskinen. Et **set** indeholder en række **packages**, der er nødvendige for installation af en større overordnet funktion.

Man kan godt vælge at lægge **sets** eller **packages** ind på sin harddisk uden at installere dem. I dette tilfælde lægges de "rå" installationsfiler blot i et særligt katalog, og ligger klar til senere at blive rigtigt installeret. En sådan løsning vil fx være velegnet i netværk, hvor andre maskiner blot kan installere fra den maskine, der har **"spool'et"** de relevante **sets** og **packages**.

Med **'sysadm software'** har man alle de funktioner til rådighed, der er nødvendige til administration af **sets** og **packages**. Både hvad angår installation og afinstallation såvel som **spooling**.

```
 2 Software Installation and Information Management
>check     - Checks Accuracy of Installation
 defaults  - Sets Installation Defaults
 install   - Installs Software Packages
 interact  - Stores Interactions with Package
 list      - Displays Information about Packages
 read_in   - Stores Packages Without Installing
 remove    - Removes Packages
```

Forklaring til de enkelte punkter:

**check** Med dette punkt kan man checke om alle filer til en package er tilstede i henhold til oplysninger om, hvad en package bør indeholde

**defaults** Under dette punkt kan man konfigurere forskellige ting såsom, hvor ny software som default skal installeres. Oplysninger, der redigeres under dette punkt opbevares almindeligvis i filen **/var/sadm/install/admin/default.**

**install** Punkt til installation af sets og packages. Hvis der angives navnet på et set, bliver de tilhørende pacgages automatisk installeret.

**interact** Med dette punkt kan man sætte default svar til installation af konkrete packages, hvis der skal besvares spørgsmål under installationen.

**list** Under dette punkt kan man se oversigter over installeret, spool'et eller endnu ikke installeret software.

**read_in** Under dette punkt kan man indlæse software uden at installere det. (Softwaren spooles og kan installeres senere). I Oracle terminologi kaldes det for et "stage area", der hvor softwaren placeres, og på en UNIX server kaldes det for et "spool area".

**remove** Under dette punkt kan man slette installeret software.

## 15.1 Installation fra carrier bånd

Med **carrier båndet** følger der altid udførlig produktinformation samt installationsvejledning. Båndet kan sagtens indeholde flere programmer end dem, som er blevet bestilt. Imidlertid kan hvert program kræve hver sin installationsnøgle, hvilket bevirker, at kun de produkter, som man har bedt om, kan installeres.

Inden man installerer et program, kan man vha. **'sysadm software list'** få information om, hvad der ligger på båndet. Under punktet **'Location of the software'** kan man bruge **'choices'** (F2) og vælge mediet, hvorpå man har modtaget programmet.

Et vindue med titlen **'Packeges on <medium>'** vil komme frem på skærmen, hvor information om navn på packages (mnemonic), om der kræves licensnøgle (L), varenummer (Stockno.) og produktforklaring (Description) vil blive vist.

Navnet på den konkrete **package**, som man ønsker at installere, har man således sikret sig fra 'mnemonic' feltet, og man kan nu installere fra **sysadm** ved **'sysadm software install'**. Under vinduet med titlen **'3 Install a Software Package'**, kan man bruge **'choices'** (F2) til valg indlæsningsmedium, som skal angives i feltet **'Package location'**. **Package** navnet skal indtastes i feltet **'Package name(s)'**.

Nu vil et vindue med titlen **'4 License Information'** blive vist, hvis programmet kræver licensnøgle. På linien **'License key'** skal man indtaste sin licensnøgle, som har følgende format:

```
4                    License Information

License key: XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XX
```

dvs. 8 blokke af 4 store bogstaver adskildt af bindestreg, efterfulgt af 2 store bogstaver adskildt af bindestreg (i alt 34 store bogstaver).

Skriver man forkert, må man gentage hele indskrivningen fra begyndelsen igen.

Licensnøglen, som låser op for installationen, er genereret vha. installationsnummeret, og det er derfor umuligt at installere på anden hardware end den maskine, som programmet er blevet bestilt til.

Brug af licensnøgle sikre, at software ikke umiddelbart misbruges, samtidig med at det er en smidig måde at distribuere software på. Man behøver ofte ikke at skaffe et program først, eftersom man allerede har det. Man skal blot rekvirere en licensnøgle.

## 15.2 Check og spooling

**Carrier bånd** kan ikke spooles, og man kan ikke udføre check af programmer på **carrier bånd.** Når et program fra et **carrier bånd** er installeret, kan man godt undersøge det for fejl.

## 15.3 Afinstallation af programmer

En **package,** som er installeret fra et **carrier bånd,** kan afinstalleres både fra shell ved UNIX kommandeon '**pkgrm <pkg name>**' eller vha. '**sysadm software remove**'.

Fra '**sysadm software remove**', hvor man i vinduet med overskriften '3 **Remove a Software Packages**', skal indtaste navnet på den **package,** som skal fjernes, på linien med teksten '**Package name(s) to be removed**'. Man kan vha. '**choices**' (F2) få oplyst, hvilke **packages** der installeret.

## 15.4 UNIX kommandoer til administration af programmer

Kommandoer til installation af programmmer:

Under **/usr/bin** findes:

**pkginfo** Der kan vise informationer om packages og set, der er installeret, spooled eller ligger på et eksternt medie.
Eksempel: pkginfo -d ctape2

**pkgparam** Der kan vise parameterværdier for de installerede packages.
Eksempel: pkgparam -d ctape2 <pkg_name>

Under **/usr/sbin** findes:

**pkgadd** Som bruges til installation eller spooling af set og packages.
Eksempel: pkgadd -d ctape2 <pkg_name>

**pkgask** Giver mulighed for at forhåndsdefinere svar, der skal gives under installation, og gemme disse på en fil.
Eksempel: pkgask -d ctape2 -r <file_name>

**pkgchk** Kan bruges til at checke om installation er OK.
Eksempel: pkgchk -d ctape2

**pkgrm** Bruges til afinstallation af set og packages.
Eksempel: pkgrm <pkg_name>

# 15.5 Opgave

*Formål:* - *at installere et nyt program*
- *at fjerne en programpakke*
- *at checke versioner på allerede indlagte programmer*

*Metode:* - *opgaven løses samlet*

1. **Et carrier bånd udleveres af instruktøren.**

2. **I får oplyst et programnavn af instruktøren. Undersøg om det ligger på de/det udleverede bånd.**

3. **Undersøg, om programmet allerede er installeret (det er det, men undersøg alligevel).**

4. **Afinstallér programmet og check, at det er blevet fjernet.**

5. **Installér programmet igen.**

6. **Check med list menupunktet, at programmet er blevet indlæst.**

7. **Check med check menupunktet, at programmet er korrekt installeret.**

# Stikordsregister

Dansk Data Elektronik A/S
Herlev Hovedgade 199
DK 2730 Herlev
Tel.: ( + 45) 42 84 50 11
Fax: ( + 45) 42 84 52 20