

måndag den 6 september 1999

UNIX / LINUX Installation:

INVENTERA HÅRDVARA

Grafikkort & Monitor

Nätverkskort

Nätverksmiljö: TCP / IP

- IP Nummer
- Nätmask
- Gateway
- Namnsserver

(Detta ställer du in med kommandot *ifconfig*)

1. Installera vad från var ???

Gratisvarianter av Linux / Unix:

Slackware Linux

Suse Linux

RedHat Linux

Debian Linux

Best Linux - svensk version

Caldera

Linux består av: Kernel Source = 4,5MB – 20MB
 Kernlar kan vara stabila eller under utveckling (instabila)
 UNIXmiljö: Kommunikation

BSD = Berkley Software Distribution

BSD är en UNIX som ställer hårda krav påhårdvaran.

BSD försöker att anpassa PC'n till UNIX

Detta medför: Stabilare system, långsammare utveckling.

Betalversioner av Linux / Unix:

Solaris, Sun Microsystems UNIX

Intel = 32bit

SPARC = 64, 128, 256bit

AIX, IBM Unix

DIGITAL Unix

AT & T Unix

HP Unix CDE Standard

IRIX

BSD Open UNIX

Installation

Från Disketter

Från Band

Från CD (Vanligast) Ofta med sk. BOOT och ROOT disk

Från Nätverket; NFS, FTP (var som helst ifrån) BOOT och ROOT disk

Från Hårddisk

KONFIGURATION

Hostnamn

Nätverksparametrar

Paket = Miljö, verktyg och applikationer

Servertjänster som önskas

Systemkällkod och Kompilator = Utveckling

PARTITIONERING

Att välja var de olika delarna av operativsystemet skall ligga och hur stort utrymme som skall tilldelas dem:

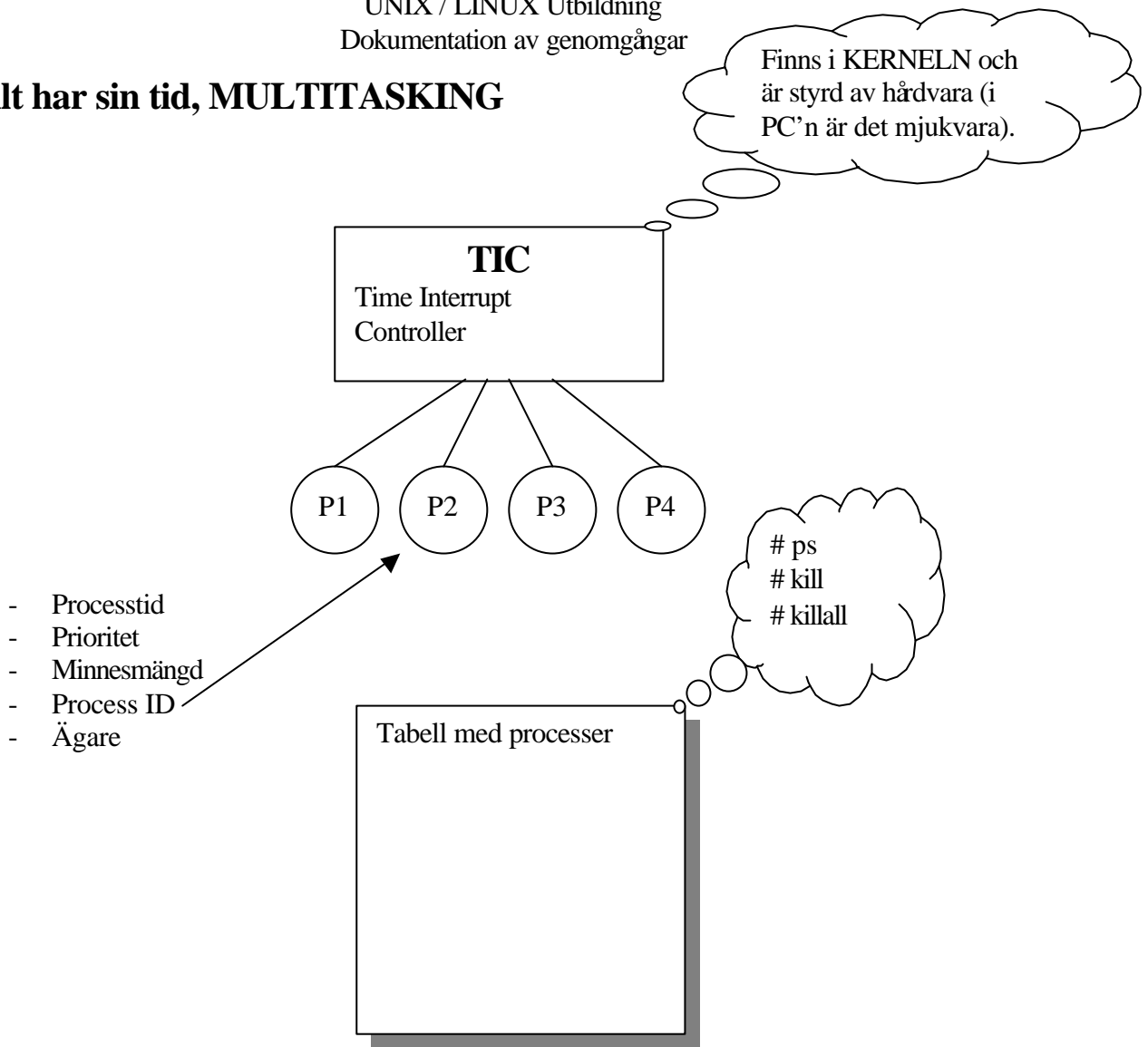
Partitioneringsrekommendation

/	<i>ROOT Filsystem</i>	40MB – 250MB (Rekommenderat 127MB)
/home		250MB +
/usr	<i>Applikationer</i>	minst 512MB
<SWAP>		Samma som minnet i datorn
/usr/local		250MB +
/tmp		127MB
/var		127MB

BOOTNING, väntan och tårar ;))

Inloggning; Kom ihåg ROOT-lösenordet som du angav under installationen!

Allt har sin tid, MULTITASKING



tisdag den 7 september 1999

UNIX Filtråd

ls ~ dir (i DOS)
cat ~ type (i DOS)
cd ~ cd (i DOS)
man <kommando>
 → Manualsidor

more och **less** kommandot:

ls -al | more → sidvis visning av fillista
ls -al | less → bläddringsbar fillista.
(avbryts med **q**)

Var finns kommandona??

/bin	=	Grundkommandon
/sbin (<i>system binaries</i>)	=	Häftigare / kraftfullare kommandon som är tänkt att användas av systemadministratören.
/usr/bin	=	Utökning till Unixmiljön i datorn
/usr/sbin	=	Utökning till Unixmiljön i datorn

← Kan liknas vid C:\WINDOWS katalogen.

ROOT – filträdet

/bin	Körbara binärer för Unix
/sbin	Körbara binärer för Unix
/usr /root	”Nytt” filträd liknande rootfilträdet med ytterligare UNIX kommandon och miljöer Administratörens hemkatalog
/var	”Alla” filer som används frekvent. T ex.: log filer, mail, news, namnservrar
/tmp	Temporära filer
/opt	”Optionally” andra större programpaket. T ex.: Office, CAD, Kalkyl mm.
/proc	SE UPP !! Dators processer, hårdvaruenheter, status och ”minne” i katalogstruktur
/mnt	Monterade filsystem. T ex.: cd rom, floppy, bandstation osv
/etc	Systemets konfigurationsfiler
/home	Användarnas hemmakataloger finns här. (FTP – ROOT)
/lib	”Kataloger” UNIX systemets grund.
/lib/modules	Kernelmoduler som laddas vid behov
/boot	Startfiler för UNIX. T ex.: ”Kernel”, vmlinuz
/vmunix	Samma som /boot. Denna beteckning förekommer främst i icke LINUX system
/dev	”Enheter” T ex.: floppy, portar, hårddiskar, terminaler, nätverksenheter, tangentbord, modem mm.

Övningsuppgifter

Om du inte är root bli det genom att skriva "su" och ange roots lösenord.

Lägg till en användare geno att skriva "useradd <användarens namn>"

Ange den nya användarens lösenord genom att skriva "passwd <användarens namn>"

Vem är inloggad??? Skriv "who" för att se det.

MOUNT

FLOPPY /dev/fd0
(Block Device) /dev/floppy (observera att *floppy* är en länk till *fd0*)

Exempel: mount /dev/fd0 /mnt/floppy

Kolla: ls -al /mnt/floppy (Nu ska du se innehållet pådisketten / floppyn)

CD ROM /dev/cdrom /mnt/cdrom

UMOUNT

(Det ska faktiskt heta **UMOUNT** och inte **UNMOUNT**)

Exempel: umount /mnt/floppy för att frigöra disketten / floppyn

FILFLYTTNING

mv <källa> <destination>

mv -dpR <källträd> <destination>

KOPIERA FILER

cp <källa> <destination>

TA BORT FILER

rm <fil / träd>

rm -r <fil / träd> **Se upp med -r växeln, den är farlig**

rmdir tar bort toma kataloger

Frågar för varje fil /
katalog. -f (force) = inga
frågor.

TEXTEDITORER

e
ed (e och ed är radbaserade)

vi fullskärmstexteditor

emacs lisp i botten och VT100 kompatibel

pico som dos edit eller Notepad

xedit X-windows editor
xemacs X-windows editor

Processtyrning

vad körs ??

ps allt du är upphov till

ps aux allt som du har rätt att se

Olika nivåer för att stänga / kill processer 1 – 9 (9 är ett skott i huvudet)

Du startar om processen med # kill <argument> -HUP

FORKS – BAKGRUNDS PROCESSER

Exempel: # cp -dpR /home/* /backup &
<blinkande markör / cursor>

& ➔ cp kommandot kommer att tilldelas en pid (Process ID) och köras i bakgrunden.

bg <komando>

fg = gåtill bakgrundsprocess

job / jobs = vilka jobb??

Övningsuppgifter:

Utforska följande kommandon:

- # df Summerar fritt hårddiskutrymme
- # du Räknar ut hur mycket utrymme som filer och kataloger tar upp.
- # cp Kopierar filer och kataloger
- # mv Döper om filer, flyttar filer
- # mkdir Skapar en katalog
- # rmdir Tar bort en katalog (tomma kataloger)
- # free Visar minnet (samma funktion som *mem* i DOS)
- # kill Dödar en process
- # killall Dödar alla processer med samma namn
- # userdel Tar bort en användare
- # rm Tar bort filer
- # rlogin (RemoteLogin) Möjliggör inloggning på en annan dator om det finns ett "identiskt användarkonto".
- # telnet Ett program som möjliggör fjärranslutning (till allt som har en IP Adress)

[onsdag den 8 september 1999](#)

INLOGGNINGEN

Lokal inloggning

Övervakningsverktyget för aktivitet vid någon av terminalerna heter *getty*.

Terminaler: (Klassiskt sett) serieportar anslutna till textterminaler.

getty anpassar linjehastighet ev. kommunikation med modem och terminal.
efter *handskakning* fortsätter processen.

(Virtuella Terminaler: Linux & Free BSD handhas också av *getty*

Telnets sessioner / *rlogin* sessioner handhas också TCP / IP sockets (port 23))

Anslutningen överlämnas till *login*. *getty* upprätthåller nu bara "linjen". *Login* sänder ut välkomstmeddelande och frågar efter användarnamn och lösenord.

UNIX / LINUX Utbildning
Dokumentation av genomgångar

Inskrivet lösenord måste krypteras, om lösenord krävs för aktuell användare överhuvudtaget finns i *passwd* filen.
kontrollen mot *passwd* filen utförs. Stämmer allt (username och password) fortsätter *login*.

uid (User ID), gid (Group ID) tilldelas, hemkatalog, SHELL miljö.
OBS!! Allt detta ur passwd filen

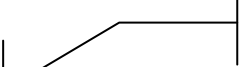
I *passwd* filen kommer först: <username> : <krypterat lösenord> : uid : gid : userinfo : <hemkatalog> : SHELL

Användarna måste erhålla en systemmiljö + grundmiljö + egna inställningar = TOTAL miljö

⇓
sökvägar
system variabler
miljö variabler
makron

OBS !! kolla detta med # set

/etc/profile = grundmiljö
/etc/shellrc
/etc/shell.rc
/etc/shell.env

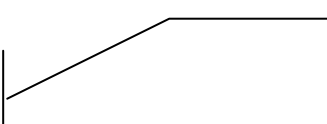


olika SHELL som körs

(system miljö erhålls vid system boot
/etc/rc.d = stor sammling start och stoppfilen)

egna inställningar:

~ <user>/ .profile
~ <user>/ .bash
~ <user>/ .zshrc



Vissa kan läggas in automatiskt vid *useradd*. Körning från viss katalog
/etc/skel (skeleton directory)

Sista steget, prompten eller det program som du ska köra dyker upp.
<blinkande markör>

set = visar alla miljövariabler

Notera: PATH = Sökvägar i systemet för program
MANPATH = Manualsidor
TERM = Linux är ej standard

torsdag den 9 september 1999

BEHÖRIGHETER

chmod = Change Mode
chown = Change Owner
chgrp = Change Group

— v — x r w x r — x orange orange
↑ ↑ ↑ ↑ ↑ ↑
Filtyp | User | Group | Other | User | Group

r = read = 4
w = write = 2
x = execute = 1
d = directory
b = block device
· = hidden
- = "nothing"

tillsammans blir
dessa tre **7**

chmod (Change Mode)

chmod 741 <filnamn> = -rwxr----X

Istället för att använda siffror kan du skriva kommandot så här:

chmod u+rwx <filnamn> nu har user fått ReadWriteExecute rättigheter
lägg märke till att + kan bytas ut mot - eller =

u = user g = group o = other a = all

OBS! Om du vill ändra på owner, group eller other skriver du på följande vis:

chmod u=rwx,g=rwx,o=rwx <filnamn> Nu har User Group och Other rwx behörighet.

chown (Change Owner)

chown <ny ägare> <fil>

chgrp (Change Group)

chgrp <ny grupp> <fil>

PACKA UPP EN FIL

tar = används för att göra om en katalogstruktur till en fil.

gzip = används för att packa filer

```
# tar xzvf <filnamn>.tar.gz
```

extract; gzip; verbose; force

PACKA EN FIL

```
# tar czv <filnamn>
```

Om du inte har *tar* eller *gzip* kan du försöka att lura datorn genom att skriva:

```
# zcat <filnamn> > <destination>
```

(*zcat* visar innehållet i en komprimerad fil)

[fredag den 10 september 1999](#)

Trevlig dokumentation om nästan allt i UNIX hittar du här:

/usr/doc/howto

/usr/doc/faq

/usr/doc/mini

Miljövariabler

- Verktyg för användaren
- Vid shellsript används dessa som lagringsplats för information
- Kommunikation mellan program

Så här går det till:

Att skapa en miljövariabel – deklARATION:

- ```
nizze="Hej hopp i lingonskogen"
```

nizze skapas och "Hej hopp i lingonskogen" lagras i nizze
- ```
# set nizze=" Hej hopp i lingonskogen"
```

som *a*

OBS! Kan användas utan = i cshell

Kolla en variabels värde:

```
# echo $nizze <enter>  
Hej hopp i lingonskogen
```

```
# set          = Visa alla miljövariabler
```

Om jag byter shell tappas alla skapade variabler.

Att byta shell:

```
# ksh          = Korn Shell  
# bash        = Basch Shell  
# sh          = Grund Shell
```

Program som startas körs oftast i ett nytt Shell!

Så här gör man när man vill behålla sina variabler vid Shellbyten:

```
# export $nizze  
Nu finns "nizze" kvar om jag byter shell!!  
Om "nizze" inte existerar kan du göra så här när du skapar honån:  
# nizze=" Hej hopp i lingonskogen";export nizze
```

Vad får jag lov att "ärva" (vid Shellbyten)?

```
# env          = (environment) alla variabler som inte försvinner vid Shellbyten
```

Om du loggar tu så försvinner variablerna som du har exporterat.

OBS!!! STORA bokstäver för miljövariabler
 SMÅ bokstäver för programvariabler

MILJÖVARIABLER MAN VILL HA KVAR

```
/etc/profile (alla shell)  
/etc/bashrc (ett shell)
```

Om du vill göra förändringarna för en särskild användare görs dessa förändringar i hans hemkatalog.

```
# echo $NIZZE          = nizze pekas ut
# echo $PATH           = path pekas ut
# echo $HOME           = Sökvägen till din hemkatalog pekas ut
```

För att "lägga till":

```
# nizze==" nizze', fint väder i dag"
```

[måndag den 13 september 1999](#)

RPM

(Redhat Package Manager)

Generella parametrar kan användas med alla rp komandon

```
-vv                    = Print debugging info
--keep-temps          = Do not remove temporary filez
--quiet               = Print as little as possible
--help                = ;)
--version             = Display rpm verion
--rcfile <file>       = Specify a different (personal) setup file
--root <dir>          = Use <dir> as the top-level directory for all operations
```

Install syntax:

```
# rpm -i <install options> <rpm file>
```

Upgrade Syntax:

```
# rpm -U <uppgrade options> <rpm file>
```

Install / Upgrade options

```
--force                = Force the replacement of a package or file
-h,--hash              = Displays job status #####
--oldpackage           = Replace a new package with an old one.
--percent               = Displays the jobstatus in %
--replacefiles         = Force previous installed files from other packages to be replaced
--replacepkg           = Forces previous installed packages to be removed.
```

RTFM !!!

Erase (uninstall) syntax:

```
# rpm -e <erase options> <rpm file>
```

Query syntax:

```
# rpm -q <query options> <rpm file>
```

(use the `-a` option to display all installed packages on your system)
(use the `-i` option to display information about the package)

Verify syntax:

```
# rpm -V | -y -verify <verify options> <rpm file>
```

tisdag den 14 september 1999

LÄNKAR

Exempel på hur du länkar "filen" quake till /usr/local/games:

```
# ln -s /usr/local/games quake
```

↑ ↑ ↑
symbolic | Det som ska länkas | länken

SCRIPT

Börja med att ange vilket shell som ska användas: `#!/bin/bash`
Detta ska alltid göras påscriptets första rad.
Därefter skriver du skriptet precis som om det vore en *.bat – fil.

UNIX / LINUX Utbildning
Dokumentation av genomgångar

 less	Bläddringsbar lista avslutas med q (= pipe)
useradd	Lägger till användare
cat	För att se vad som finns i en text fil.
cd	Flyttar dig till hemkatalogen
cd ..	Flyttar dig ett steg uppå
chgrp	Byter grupp
chmod	Ändrar filrättigheter
chroot	Byter rotens plats från "/" till någon annan katalog
chown	Ändrar påägare till filen
clear	Rensar skärmen
cp	Kopierar filer och bibliotek
df	Visar ledigt diskutrymme
du	Visar använt diskutrymme
echo	Visar text påskärmen
env	Visar de variabler som inte försvinner vid shell byten.
exit	Avslutar
find	Sökverktyg
free	Visar ledigt och använt minnes utrymme
grep	Används när du vill specificera ditt sökande.
gzip	Används för att packa filer
kill *1-9	Dödar en process
killall	Dödar alla processer med samma namn
ls	Listar filer och bibliotek
ls /	Listar root biblioteket
ls -al	Listar alla även dolda filer och bibliotek + beskrivning
ls -l	Listar filer och bibliotek + beskrivning
man	Visar manualsidor
mcedit	Text editor som hör till Midnight Commander
mkdir	Skapar kataloger
more	Visar innehållet i en textfil, en skärm bild i taget, (se cat).
mount	Montera enheter t.ex. Floppy, Cd-rom, filsystem
mv	Flyttar eller byter namn påfiler
passwd	Byta lösenord (lösenorden och annan information sparas i filen /etc/passwd)
ps	Visar vad som körs som du är upphov till
ps aux	Visar allt du får lov att se som körs
pwd	Visar hela sökvägen
rlogin	För inloggning på en annan dator om det finns ett "identiskt användarkonto".
rm	Tar bort filer
rmdir	Tar bort tomma kataloger
rpm	Används för att hantera *.rpm paket
set	Visar alla miljövariabler.
tar	Gör om en katalogstruktur till en fil, används även för backup.
telnet	Ett program som möjliggör fjärranslutning med hjälp av IP adresser.
touch	Skapa en fil
umount	Ta bort enheter
userdel *-r	Tar bort användarkonton. * Även deras hemkatalog
vi	Text editor
who	Visar vem som är inloggad
whoami	Visar vem jag är inloggad som
umask	Set file creation mask (subtrahera umasks värdet från standardvärdet → resultat)
cut	Sortera ut fält eller tecken
ln	Skapa länkar

onsdag den 15 september 1999

FTP

ftp tellus.linux.kurs (Du kan även använda IP nummer)

ftp -i tellus.linux.kurs (-i = inga frågor)

get = hämtar en fil
mget = hämtar många filer (multiple)

put = lämnar en fil
mput = lämnar många filer (multiple)

lcd = Att byta katalog på "min" dator
! = Ett nytt Shell startas

TELNET

telnet -l orange tellus.linux.kurs

Syntaxen visar att jag telnetar till *tellus.linux.kurs* . -l visar att jag begär att fålogga in som *orange* på *tellus.linux.kurs* .

X WINDOWS

"GRUND"

Shell, Kernel, Hårdvara

X Server

Grafiska gränssnitt

CDE (Common Desktop enviroment)

Enlightenment

AfterStep

WindowMaker

Besök www.xfree86.org

torsdag den 16 september 1999

JOB MANAGING

Ctrl+z = Pausa ett program och lägga det i bakgrunden.
& = skriv detta tecken efter ett kommando för att köra kommandot i bakgrunden.
bg = Back Ground
fg = Fore Ground
jobs = visa vilka jobb som körs i bakgrunden.

RUNLEVELS (Körnivåer)

0 = halt
1 = single user mode
2 = multiuser mode without NFS
3 = multiuser mode with NFS
4 = empty
5 = X11 (X-windows)
6 = reboot

KERNELKONFIGURERING & KOMPILERING

[/boot = här finns KERNELN
 / /lib/modules = här finns MODULERNA
 / /etc/lilo.conf = detta är LILO's konfigurationsfil
 / /usr/src = här finns källkoderna

OBS!! Ta backup på dessa innan du sätter igång !!!

- 1 Kartlägg din hårdvara.
- 2 Vad används datorn till?
- 3 SKRIV NER ALLT!
- 4 Packa upp kerneln i en temporär katalog.
- 5 Flytta den temporära katalogen till **/usr/src/** och ge den ett lämpligt namn.
 Till exempel **kernel-2.2.12**
- 6 Gör en symbolisk länk som heter **linux**. Denna länk ska peka till den nya kernelns källkodskatalog.

UNIX / LINUX Utbildning
Dokumentation av genomgångar

7 Skriv **# cd /usr/src/linux** för att förflytta dig till den kernelns källkodskatalog.

8 Nu kan du anropa setupprogrammet genom att skriva:

make xconfig	= grafiskt (X windows)
make menuconfig	= ANSI grafik
make config	= testbaserat

9 När du har tagit dig igenom setuppen skriver du följande:

make dep&&make bzImage&&make modules&&make modules_install&&make install

make dep	= make dependencies
make bzImage	= skapar kerneln (b = kompression)
make modules	= skapa moduler
make modules_install	= installerar moduler
make install	= installerar kärnan

make mrproper = nollställ allt vid fel

måndag den 20 september 1999

NÄTVERKET

195.178.165.245



OKTETT

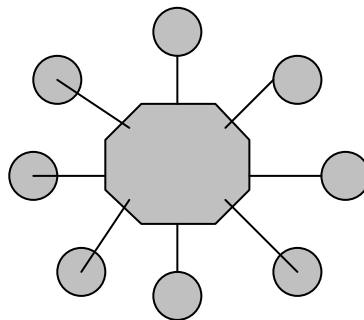
LAN = Local Area Network

WAN = Wide Area Network

LAN nätverket; Ethernet, Token Ring, FDDI

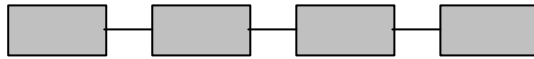
Topologier:

Stjärnnät:

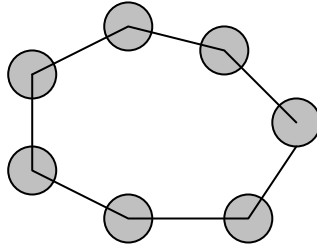


UNIX / LINUX Utbildning
Dokumentation av genomgångar

Busnät:



Ringnät:



Kablar:

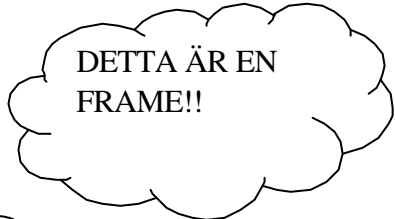
10 Base 2	Thin Koax	185m
10 Base 5	Thick Koax	500m
10 Base F	Fiber	
10 Base T	TP kabel (TP = Twisted Pair)	100m
100 Base X	TP Kabel	
100 Base TX	TP Kabel 2 par (1 - send 1 - return)	
100 Base FX	Fiber Kabel 2 par (1 - send 1 - return)	
100 Base T4	TP Kabel 4 par (3 - send 1 - return)	

Repeater; rensar signalen och återskapar den mellan kabelsegment.

HUB

En HUB är en repeater där varje NOD kopplas in i nätverket.

ETHERNET = 10 Base T Teknologi
Fast ETHERNET = 100 Base T Teknologi



FRAMES (Datapaket)

Varje NOD i ett LAN har en MAC adress (Media Access Controll)

SYNC	MOTTAGARE	AVSÄNDARE	TYP	DATA	FCS
64	48	48	16	344 – 12 000 bitar	(felkontroll)

Regler för datatrafiken!!

Media Access Rules

När man använder Fast ETHERNET kallas reglerna för: CSMA/CD
(Carrier Sense Multiple / Collision Detection)

PROTOKOLL

För att datorer ska kunna skicka och ta emot data såfodras ett protokoll. Ett protokoll talar om hur datapaketet ska vara uppbyggd.

Microsoft kallar sitt protokoll för NetBEUI
Novell kallar sitt protokoll för IPX / SPX
UNIX kallar sitt protokoll för TCP / IP

OSI Modellen

OSI = Open System Interconnection

Består av 7 lager och beskriver den logiska kommunikationen mellan datorer.

7	A	Application	Fil och Srvartjänster, WWW och FTP
6	P	Presentation	Data encryption and decryption (Secure Socket Layer)
5	S	Session	
4	T	Transport	CRC controll
3	N	Network	
2	D	Data Link	Styr när en nod kan sända och ta emot data
1	P	Physical	Hur man lägger ut elektroniska signaler påkabeln

All People Seem To Need Data Processing

och så tar vi det segnälkaB

Please Do Never Throw Sausage Pizza Away

IP Nummer

IPv4 (består av 4 oktetter)

IP nummret är den unika logiska identifiering av en NOD I ett nätverk.

Internt nätverk, enbart unika nummer för detta nätverk.

Om nätverket ska kopplas till internet såbehövs det IP nummer som är unika påinternet.

IP adresser består av 4 oktetter. Segmenten är mellan 0 – 255 ”stora”.

00000000.00000000.00000000.00000000 = 0.0.0.0
11111111.11111111.11111111.11111111 = 255.255.255.255

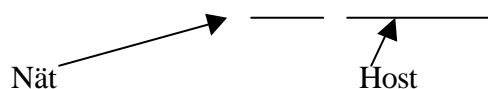
↑
OKTETT

Alla tillgängliga IP nummer har man delat upp i 3 segment: Klass A, B och C. Varje klass identifieras av värdet påden första oktetten i IP numret.

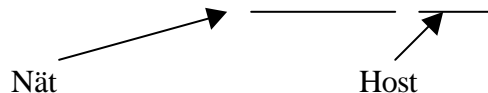
Klass	Område	Antal Hostar
A	1 –126	ca 16 milj
B	128 – 191	ca 65 000
C	192 – 223	254

127.0.0.1 Loop Back adress !!!

Klass A nät **126.1.28.76**



Klass B nät **191.125.20.1**



Klass C nät **195.178.165.245**



tisdag den 21 september 1999

IFCONFIG (småbokstäver # ifconfig)

Detta verktyg används för att sätta upp och konfigurera nätverkskortet i datorn.

Nätverkskonfigurationen görs tidigt i startsekvensen så att andra nättjänster som startas ska veta hur de ska kommunicera med omvärlden.

Syntax:

```
# ifconfig <interface> <IP adress> <options>
```

ifconfig behöver inget "--" framför operanden.

Interface är den nätverksenhet som skall sättas upp. Om du använder Ethernet så blir det **eth**, följt av det nummer som visar vilket Ethernetkort du avser. Numreringen startar på 0 (noll). Det första kortet heter alltså **eth0**.

IP adress är den adress som du vill ge ditt nätverkskort.

ifconfig använder default nätmask och broadcast som baseras på din IP adress. Du kan givetvis ställa in dessa manuellt efter egna behov.

I detta fall använder du parametrarna netmask och broadcast följt av den adress som du vill ge dem.

exempel:

```
# ifconfig eth0 192.168.42.2 netmask 255.255.255.0 broadcast 192.168.42.255
```

Networking

UTP/STP Category	Speeds
Cat 3	16 mbps
Cat 4	20 mbps
Cat 5	100 mbps

Ethernet specifications

Type	Cable types	Connection type	Max lenght
10Base2	RG-58 thinnet coax	BNC T-Connector	185 m
10Base5	Thicknet coax	DIX/AUI (15pin)	500 m
10BaseT	Cat 3, 4, or 5 UTP	RJ-45	100 m
100BaseT	Cat 5 UTP	RJ-45	100 m

TCP/IP

Split	Mask	Nät	Host (A)	Host (B)	Host (C)
Net/Host			Host + 16bit	Host + 8bit	Host + 4bit
1/7	128	2			126
2/6	192	4	4096K	16382	62
3/5	224	8	2048K	8190	30
4/4	240	16	1024K	4094	14
5/3	248	32	512K	2046	6
6/2	252	64	256K	1022	2
7/1	254	128	128K	510	0
8/0	255	256	64K	254	0

Class	Range	Nät.Host
Class A	1-126	X. <u>X</u> .X.X
Class B	128-191	X.X. <u>X</u> .X
Class C	192-223	X.X.X. <u>X</u>

128 64 32 16 8 4 2 1

onsdag den 22 september 1999

10 megabit = 10 000 000 bps

Dämpning (störningar)

IMPENDANS:

Resitiv = Signalen bromsas / störs pga motstånd i kabeln.

Kapacitiv = störningar pga "restspänning"

Induktiv = Störningar av magnetfält

Störningar:

EMP = Electro Magnetic Pulse

Kablar som ligger för nära varann. Elmotorer som startas och stängs av. Blixten. Solen.

Kommunikation

Simplex = Kommunikation som endast kan ske åt ett håll.

Duplex = Dubbelriktad kommunikation.

Halv Duplex = Endast en i taget kan prata.

Full Duplex = Bägge kan prata samtidigt.

IEEE = Institute of Electrical and Electronic Engineers

IEEE 802.3u = Standard för Fast Ethernet

IEEE 802.5 = Standard för Token Ring

torsdag den 23 september 1999

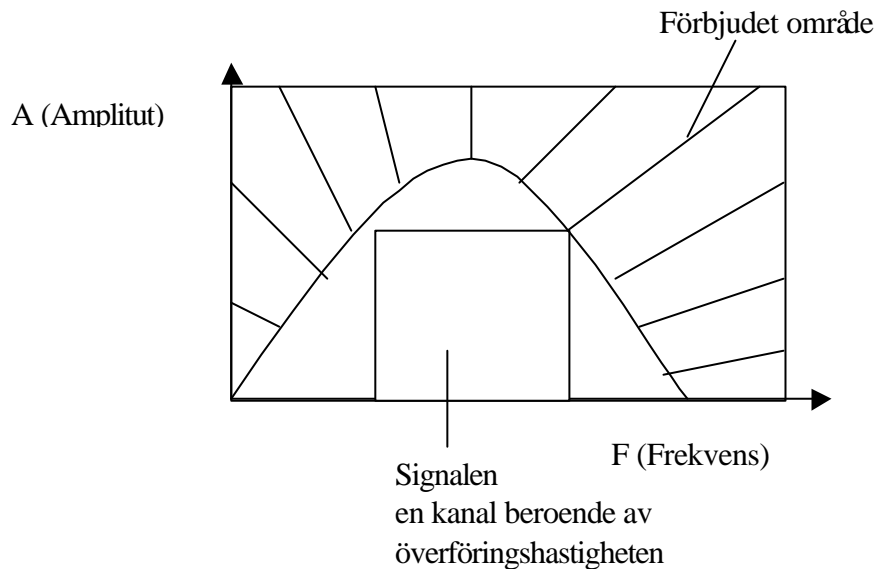
VERKTYG I NÄTVERKET

- NIC (Network Interface Card)
- WAN Adapter (WAN = Wida Area Network)
- Bryggor
- Hubbar
- Switchar
- Routrar
- Gateway
- Repeater

TP Kabel (TP = Twisted Pair)

10 BASE T

Hastigheten som kabeln eller NICet är specad för
Baseband. Hela kabeln utnyttjas enbart för datatrafik. En kanal.
Twisted Pair

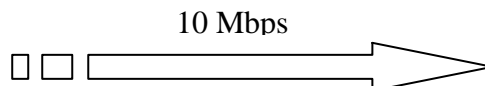


Full Duplex (alla måste ha stöd för det om det ska fungera)

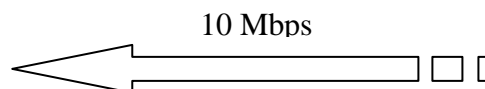
Dock bara 2 enheter å gången.

Flera par

TX par



RX par



Tänk på att det är tvinnade par!!

Twisted Pair

Twisted Pair finns som: Skärmad, Oskärmad, Foliaskärmad och Dubbelskärmad

Observera att kapaciteten är den samma under idealiska förhållanden.

STP = Shielded Twisted Pair

UTP = Unshielded Twisted Pair

FTP = Foil Shielded Twisted Pair

Skärmen ska hindra yttre signaler från att tränga in i kabeln och vice versa.

OBS!!! Det är olagligt att dra Spännings och Signalkablar tillsammans ;)

JORDNING

Jordning vid användning av skärmade kablar och jordning i allmänhet.

- Allt ska jordas på samma ställe.
- Jordpunkten sätts lämpligt där HUBBAR eller SWITCHAR står därför att allt strålar samman där.
- Jordplanet som det också kallas har också som uppgift att vara nollplan för signalen.

Observera att det är speciellt viktigt med jordning då du använder Koaxial kabel.

TP

Antalet par i en kabel kan vara 2 eller 4.

4 par = 2 st TX, 2 st RX eller alla TX , alla RX

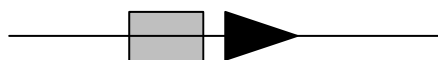
100 VG AnyLAN = 4 par Voice Grade kabel. Kräver att alla kablar används för kommunikation ≈ Full Duplex 50 Mbit

REPEATER

- Rena signalförstärkare som har som uppgift att förlänga nätverkets räckvidd. Störningar förstärks också



- Signalförstärkare med störningsfilter. Störningar filtreras till viss del bort.



- Signaluppfreskande Repeater



D/A = Digital / Analog omvandlare

DSP = Digital signal processor som uppfreskar den digitala signalen

A/D = Analog / Digital omvandlare

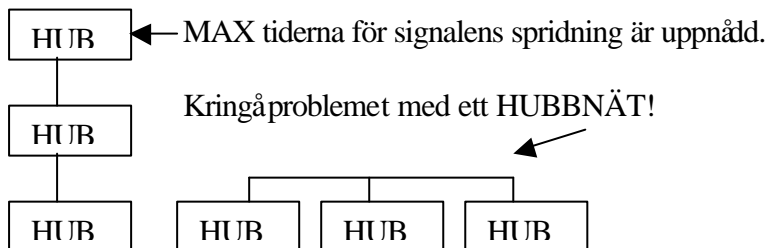
Repeater finns för LAN, WAN, MAN (Metropolitan Area Network)

Observera att Repeater förstärker också störningar. Om många Repeaters sätts in efter varann ökar störningsnivån.

Alla Repeater har en fördröjningstid.

TP Nät

TP nät: max 3 nivåer utan Switchar.

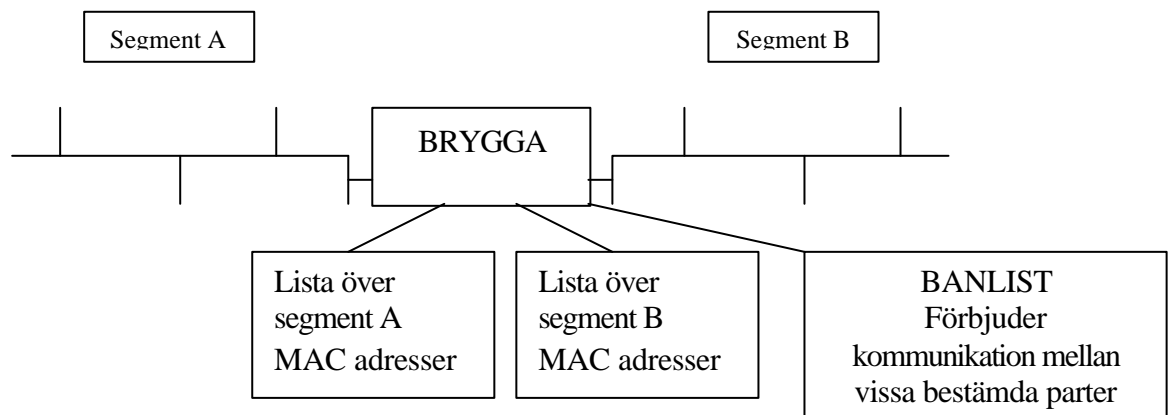


BRYGGOR

Uppgift: Att reducera nätverkstrafiken i "pratiga" nätverk. Öka prestandan i olika segment genom logisk isolering mellan dessa. Arbetar på MAC nivå

Statiska Bryggor

Du konfigurerar dessa helt manuellt. Du programmerar vilka MAC adresser som finns på varje sida om Bryggan, samt vilka av sidorna som ska ha kontakt med varann.



Dynamiska Bryggor

Till viss del manuella och självlärande.

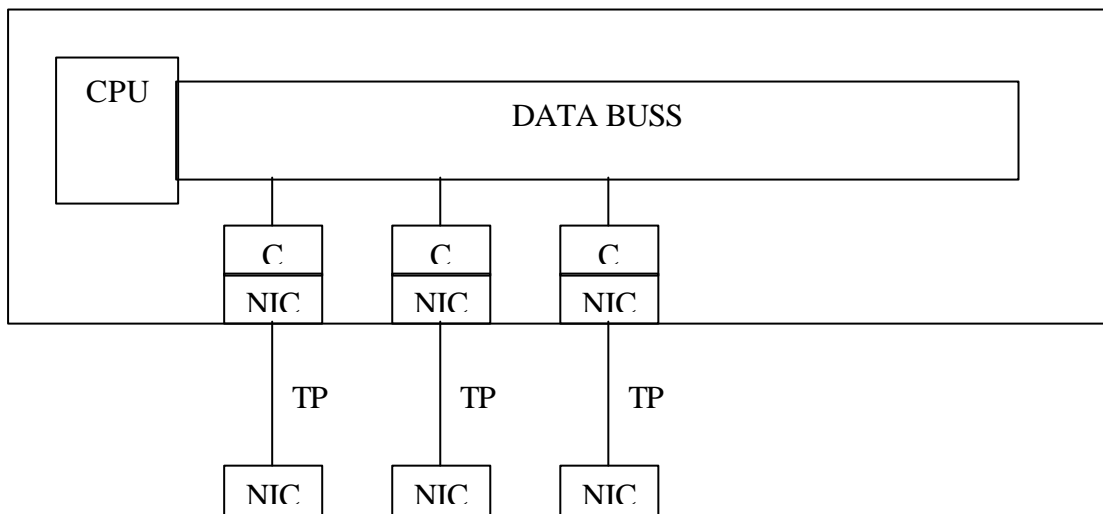
Brouter

En kombination av Dynamisk Brygga och Router.

HUBEN

Uppgift: Att göra slut på alla kollisioner ute i nätverket och att samla problem i en Bredbandsbuss inne i Hubben.

Reducera möjligheter för en enda komponent att stoppa all trafik på ett subnät. Öka bandbredden mellan olika stationer i ett nätverk.



Hubbmodeller:

- Repeterande Hub
- Passiv Hub, har bara inbyggd förstärkare men saknar CPU
- Intelligent Hub, känner vilken MAC adress nätverkskortet har som är ansluten till respektive NIC i hubben.

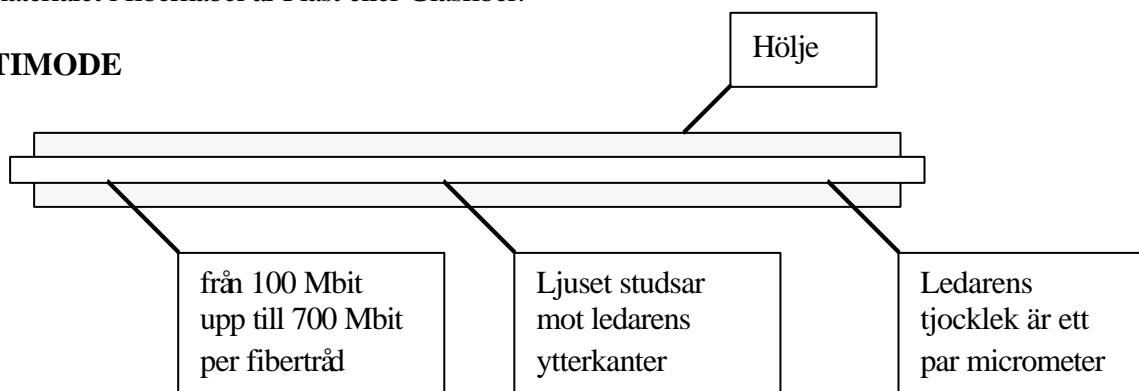
FIBERNÄT

FDDI är en vanligt förekommande standard.

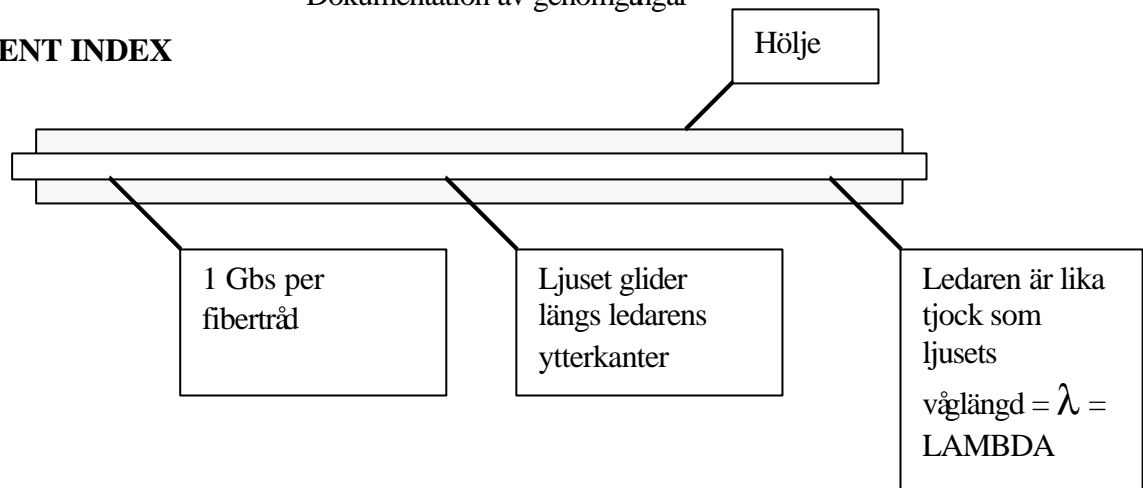
Monokomatiskt Ljus = Laser

Ledarmaterialet i fiberkabel är Plast eller Glasfiber.

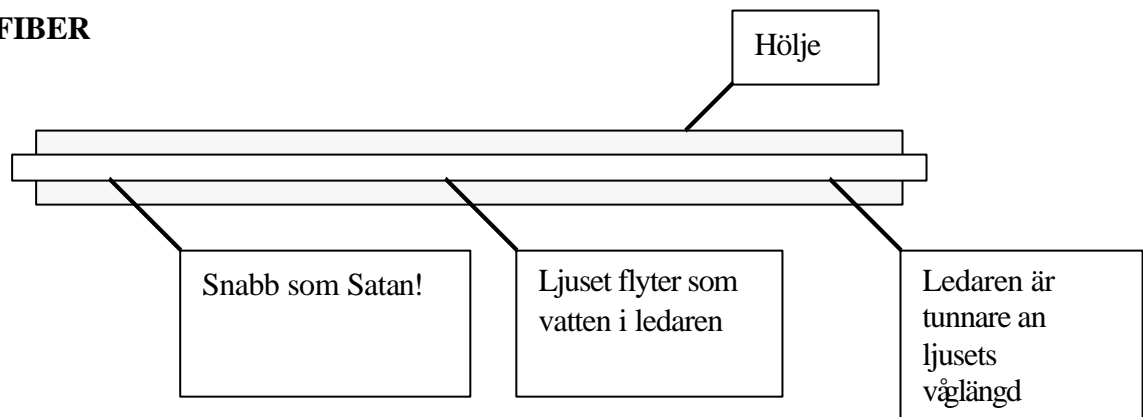
MULTIMODE



GRADIENT INDEX



MONOFIBER



SWITCHEN

Motorvägs på och avfart. Olika hastigheter från olika nätsegment sammanbinds med Switchen. Den ökar bandbredden genom att isolera de Broadcastdomäner som en Brygga.

Signalöverflöd som uppstår mellan 100 och 10 Mbit buffras i minnen.

Switchens portar kan ha IP nummer och MAC adresser.

Switchen kan känna av en hel domäns MAC adresser på en port.

SNMP

Styrnings och övervaknings verktyg för hela nätverk.

Routrar

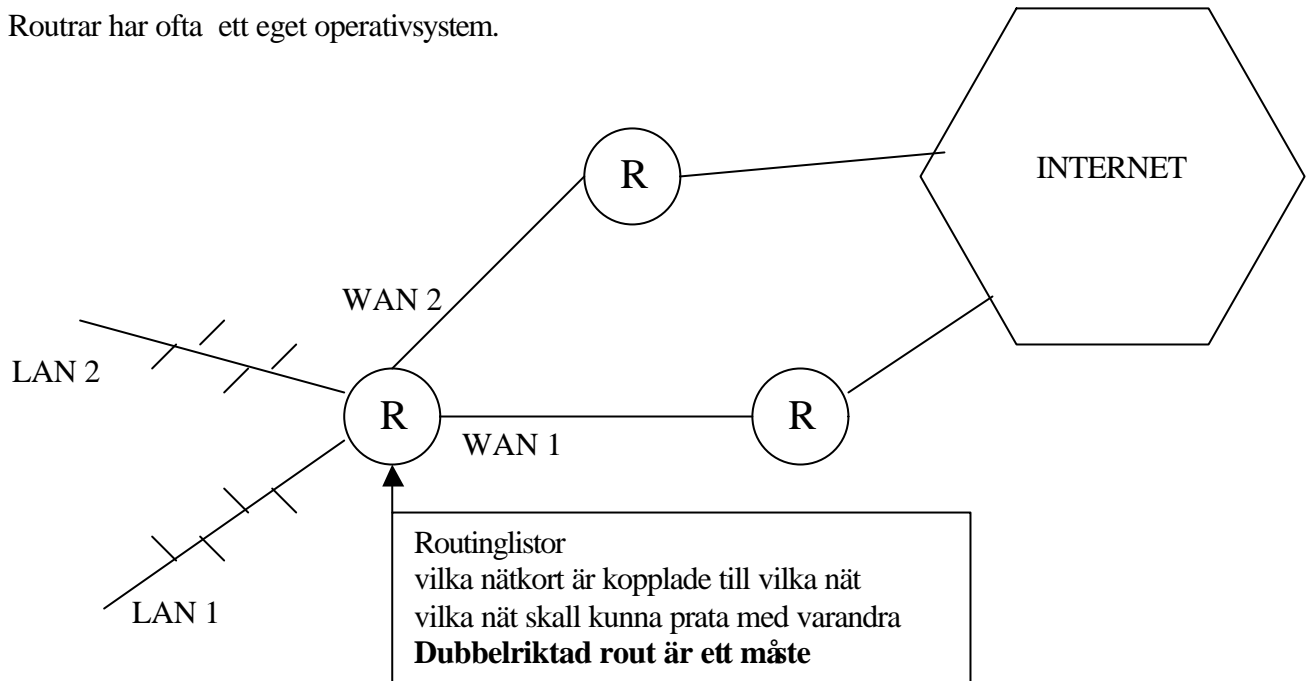
TCP / IP Routrar

Uppgift: Dirigera datapaketet mellan domäner / nätverk.

Routern är ofta hårdvarudestination för datatrafiken utanför mina LAN. Mjukvaruadressen (IP adressen) är den egentliga destinationen någon annanstans.

Routern analyserar IP adressen som är destination och sänder vidare datapaketet i den riktningen som destinationen finns.

Routrar har ofta ett eget operativsystem.



Dynamiska Routrar:

delar och lär sig av varandra i nätet, dock finns grund routing, statisk lagrad och skapad av dig eller en annan människa.

Protokoll: RIP v1, RIP v2, OSPF (Open Shortest Path First)

GATEWAY

Protokoll och mediaomvandlare.

Vem kan ha en Gateway?

- Min egen Dator
- En Router
- En Server

PSTN = Public Switched Telephone Network

Kommandon:

arp = visar vilka MAC och relaterade IP nummer datorn varit i kontakt med.

netstat = visar datorns nätstatistik, in och utgående trafik

route = routing listor och modifiering av dessa.

route add <dest> <source> <nic> <ev. nätmask>

route delete <dest>

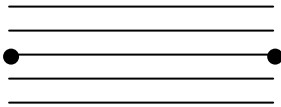
traceroute <dest> = visar väg och namnupplösning
dig <dest> = visar väg och namnupplösning
ping <dest> = kollar om kontakt erhålls
tcpdump = all trafik som passerar mitt NIC

fredag den 24 september 1999

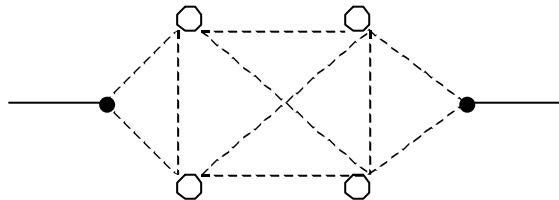
Nätlösningar

●————● Point to Point

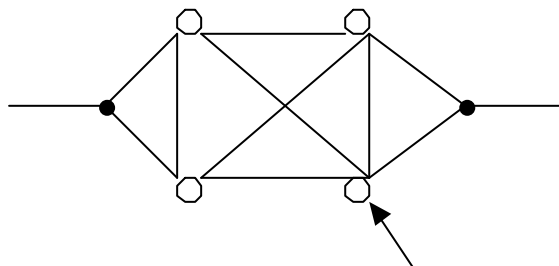
- PPP
- SLIP
- DIP



Last tankas med flera linjer (bundle) mellan punkterna. Styrs av trafiken mellan punkterna och kostnaden. Detta kan ske hårdvarumässigt eller mjukvarumässigt.



Switchat nätverk, uppkopplingsker steg för steg. PSTN
Uppkopplingar varar under hela kommunikationstiden.



Paket switchade nätverk. Routers i punkterna!

tisdag den 28 september 1999

PAKETHANTERING I SOLARIS

- # pkginfo = listar vilka paket som är installerade
- # pkginfo -d <sökväg> = Listar vilka paket som finns under den specificerade sökvägen.
- # pkginfo -l <paket> = Listar utförlig information om paketet
- # pkginfo | wc -l = Visar antalet installerade paket.
- # pkgrm <paket> = Tar bort paketet
- # pkgadd -d <sökväg> = Installerar paket
- # pkgchk -p <sökväg> = Används för att se om filer har ändrat på sig efter installationen.

Filen */var/sadm/install/contents* innehåller en lista över vilka filer som har installerats genom paket.

PATCHER

SOLARIS

Information om alla patcher som är installerade hittar du i */var/sadm/patch*

- patchadd = installera en patch
- patchrm = avinstallera en patch
- patchadd -p = visar vilka patcher som är installerade
- shorev -p = nästan samma som ovan

onsdag den 29 september 1999

RUNLEVELS I LINUX

- 1 Single user mode
- 2 Multiuser, without NFS (The same as 3, if you do not have networking)
- 3 Full multiuser mode
- 4 unused
- 5 X11
- 6 reboot (Do NOT set initdefault to this)

RUNLEVELS I SOLARIS

0	PROM monitor level
1	Administrative state (single user mode with filesystem mounted)
2	Multi user level (with no resources shared)
3	Multi user level (with resources shared)
4	Not currently used
5	Halt
6	Reboot to default run level 3
S eller s	Single user mode

init 0 är samma sak som kommandot "halt"

init 6 är samma sak som kommandot "reboot"

shutdown har samma funktion som *init* men kan informera användare att systemet kommer att gå ner.

Dessa boot kommandon kan användas från PROM:en på en Sparc:

boot -a	Steg för steg start.
boot -s	Single user mode (bra för att fixa till problem med startscript).
boot -r	För att läsa om filen /etc/system

[måndag den 18 oktober 1999](#)

Klassiska programmeringsspråk i Unix

Högnivåspråk:

- Tolkande / Interpreterande språk
- Kompilerande språk

C , C++, Ada, Fortran 77

Pascal, Modula

Kriterier för val av språk / verktyg:

Shellscript: Filhantering, typ organiserande uppdrag.

- Installationsprogram
- Enklare rutinuppdrag
- In och utloggningshantering
- Uppstartsförlopp

Ideer: Shellscript som tenderar att bli större än 20 kb så bör ett C program eller liknande skrivas i stället.

Grafik och ljud lämpar sig för högnivåspråk men TCL kan anges som skal mot shellscript.

- Shellscript finns i olika dialekter, en för varje shellmiljö; sh, bash osv

- GNU AWK
- Python
- Javascript (Webbhistorier)
- Perl
- zed

Hur bryta ner ett problem?

- Försök att genomsåda problemet genom att bryta ner det i så små delar som möjligt.
- Skriv ner delproblemen i sk. MEMOKOD

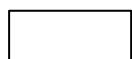
exempel: Koka Kaffe

a) Deklarera kaffebryggaren (en gång)

pulverbehållare
vattenbehållare
vattenkokare
värmeplatta
filter
dosering
mätanordning

1. Styrka?
2. Antal koppar?
3. Beräkna mängd pulver och vatten
4. Fylla på pulver och vatten
5. Vattenkokare skall startas
Koka tills vattnet är slut
6. Stoppa vattenkokaren
7. Starta värmeplattan
8. ge signal

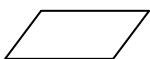
Flödesdiagram



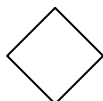
Händelseruta



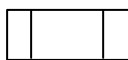
Start och Stopp av program, programhuvud



Händelseruta

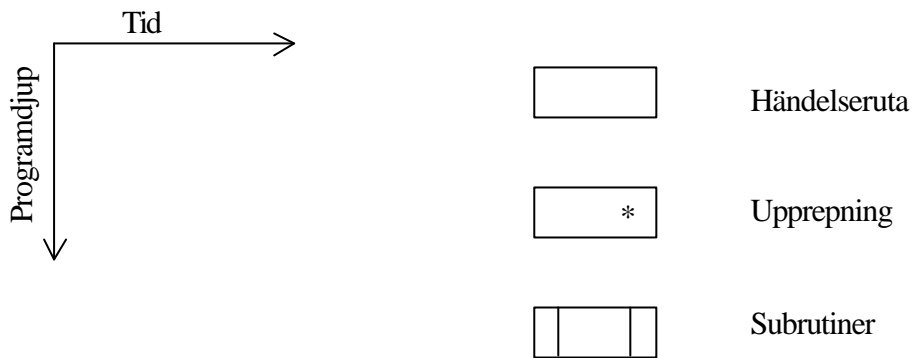


villkorsruta



Subrutinsruta

JSP – Diagram (Jacksonville Structured Programming Language)



UML

Objektrelaterad programmering

Problem löses upp i projekt. Problemen blir "globala". Generella strukturer.

Programmstrukturen

- Programhuvud:
Var, När, Hur, för vem, av vem
- Inkluderingsdel
- Deklarationsdel ←
- ofta variabler, konstanter, funktioner och variabler
- Programkoden, Huvudprogrammet
- Subrutiner

tisdag den 19 oktober 1999

Argument till script

```
# script.sh kalle nisse other
```

Inne i scriptet blir dessa argument omvandlade till variabler.

```
$1 $2 $3 osv ...
```

Test om det finns argument

```
#!/bin/sh  
if [ $# -gt 0 ] ; then  
  echo $#  
  echo "Argument finns"  
else  
  echo "Inga argument finns"  
fi
```

Hantering av argument och shellmiljö runt shellsript

`$#` antal argument till shellsript (heltal)
 `$?` sista raden som utförts i scriptet eller innan scriptet
 `$0` shellsriptets namn
 `$@` argument till shellsriptet i form av en sträng
 `$*` som ovan

Interaktioner / Uppprepningar

”for loopen”;

syntax:

```
for <värde> in <databas>
```

```
do
```

```
.
```

```
.
```

```
done
```

`<databas>` = mellanslag separerad sträng eller vagnretur separerad sträng
OBS!! Mellanslag går före Vagnretur

`<värde>` = tilldelas sekvensionellt till de olika ”elementen” i databasen som är mellanslags eller vagnretursseparerad

exempel:

```
for ARGX in "$@"
```

```
do
```

```
echo $ARGX
```

```
done
```

OBS!! Kontrollera alltid att argument finns innan du startar upp for loopar mm.

Strängjämförelser

= lika med
!= icke lika med
-n stränglängd längre än 0 tecken
-z stränglängd 0 tecken

Numerisk kontroll

tssh

-gt	större än	>
-eq	lika med	=
-le	mindre än eller lika med	<=
-ne	skilt från	<>
-ge	lika med eller större än	>=
-lt	mindre än	<

Boolsk kontroll

! not
-o or
-a and

Fil och katalog kontroll

-d katalog
-f fil
-x körbar
-r läsbar
-w skrivbar
-e finns sökvägen?
-o owner (testar om ägaren stämmer)
-z (-s) nollfil (touch-ad)

while [uttryck]

```
do  
.  
.  
done
```

↑ vilket som helst

Uttryck testas innan komandon utförs. dvs. falskt → komandon utförs ej, även första gången. Slingan upprepas så länge uttrycket / villkoret är sant.

OBS!!

För att komma ur en slinga av denna typ måste villkoret bli falskt av de komandon som omfattas av slingan eller via "flaggsystem"

until [uttryck]

```
do  
.  
.  
done
```

Samma som while men until utförs alltid en gång efter villkoret tas bort. Inmatningar från användare är bra att kolla med until.

shift <nummer>

flyttar argument till ett shellsript ett eller ett <nummer> "steg" åt vänster.

repeat <antal> <nummer>

upprepar komandon antal gånger

onsdag den 20 oktober 1999

Hur gör man ett menysystem?

```
if [ "$VAL1" = "1" ] ; then
    bla bla
elif [ "$VAL1" = "2" ] ; then
    bla bla
```

case-satsen (switch)

```
case $VAL in <databas>
    0|1) uttryck1 ;;
esac
```

VERKTYG

break	ta sig ur loopar eller andra villkorsfunktioner
exit	går ur nuvarande shell (hoppa ur shellsript)
exec	anropar nytt shell och kör i detta shell, återvänder ej

FUNKTIONER

```
funktionen_nisse () {
    uttryck
    .
    .
}
```

VARIABLER OCH SHELLSCRIPT

systemvariabler

miljövariabler

"egna" variabler

lokala variabler: existerar enbart i den miljö de skapas

globala variabler: syns överallt i hela systemet

tisdag den 2 november 1999

C – Programmering

- Funktionsrelaterad (allting är funktioner)
- Deklarationsdelen
 1. Includes (Headerfiler med standardinstruktioner)
 2. Typer (Olika egen definierade typer)
 3. Variabler av global typ
 4. Funktioner (egna funktioner)(Lokala variabler)
- Kompilatordirektiv

- Huvudprogramet / Huvudfunktionen

```
/* Var, när, hur och varför */  
. .  
#include <stdio.h>  
  
void test ();  
void test ()  
{  
printf ("Hello world \n");  
};  
main ()  
{  
test ();  
}
```

Standard
Input Output

Deklarera test, alla variabler,
konstanter, procedurer och
funktioner måste ha ett
minnesutrymme att leva på

Plats för ev argument

Kompilering av program

CC C compiler

I Linux

GCC GNU compiler

C compilatorn är konstruerad för att kunna generera körbara binärer till respektive plattform / CPU.

Själva C compilatorn måste dock vara körbar på aktuell plattform.

"Alla" headers, libbar, includes och typer är generella.

PORTABELT

Flytta program mellan olika plattformar med hjälp av
omkompilering, innehåller ny kodning.

```
# cc hello.c ( # cc -o hello.exe hello.c)
```

```
# a.out (#hello.exe) ← vår binär
```

Variabler och datatyper

int	heltal	± 32 768
char	tecken	ascii 7/8 bitar
float	flyttal / decimaltal	32-bitars tal
double	flyttal / decimaltal	64-bitars tal

Deklarera variabler

```
int talA;  
char tecken;  
int talB, talC, talD;
```

Står de i "huvudet" av programet blir de automatiskt globala.
Står de inne i funktionen blir de lokala.

Kan även vara deklarerade inne i main.

```
main (  
{  
int A, B, C, D;  
.  
koden  
.  
}
```

unsigned int	0 – 65 535
long int	32-bitars tal
short int	8-bitars tal

Typdeklaration

```
type def long heltal;
```

```
.  
main (  
{  
heltal Q;  
heltal M;  
}
```

```
#define NULL '\0'  
#define CR '\n'  
#define PII '686'  
define MAX_VALUE 90
```

Konstanter och min / max konstanter STORA BOKSTÄVER

ut och inmatning

- filer, nät
- interaktiva, konsolen

getchar ();

bokstav = getchar (); (ett tecken)

char bokstav;

.

.

printf ("Tryck påen tangent \n");

bokstav = getchar ();

printf ("Du tryckte på%c \n", bokstav);

putchar (bokstav);

%c	char	tecken	
%d		decimala heltalsdelen	
%e		exponenten	*10 ³
%o		oktal del	bas 8
%s		strängar	
%f		flyttal	(6.2f = 6 heltal och 2 decimaler)
%x		hexadecimal	

OBS!

Vederböriga typer är viktiga vid utskrift så att utskriften skrivs ut på ett korrekt sätt och att ej körningsfel uppstår.

Markörstyrning

\b	backspace, bakåsteg
\r	vagnretur, radbyte
\f	sidmatning, ny sida
\n	ny rad
\0	null tomtecken
\t	tabb

Villkorsoperander

>
<
<>
>=
<=
==

Jämförelser mellan tal

if (talA > talB)

= + - * /
upp- / nedräkning
++
--
a = talA ++

Villkor && ~ and
Villkor || ~or

if (talA > talB && talC > talD)

Bitoperationer

&	bitvis
	eller
<<	vänstershift
>>	högershift
~	ett komplement
!	not
^	exclusive eller

Villkor

if
else if
else if
else

for
do ~ while
goto ~ "Usch!"
switch

```
if ( talA > 12)
{
programkod
};
```

Om villkor tenderar till att växa tilldela en variable till ett villkor

```
for (( initiering ) ; (villkor) ; [stegning])
    <instruktioner att upprepa>
```

Exempel:

```
for ((tal=0) ; (tal < 10) ; tal ++)
printf ("Tal: %d", tal);
```

<i>initiering</i>	<i>tal = 0</i>
<i>villkor</i>	<i>tal < 10</i>
<i>stegning</i>	<i>tal ++</i>

```
while (villkor)                                KOLLAR FÖRST, KÖR SEN
{
<utför>
};
```

```
do                                             KÖR FÖRST OCH KOLLAR SEN
.
uttryck
.
while (villkor)
```

```
break;
avbryt pågående uppdrag
```

```
continue;
gå tillbaka ett steg därifrån det bröts och fortsätter
```