

LTSP – Linux Terminal Server Project – v3.0

James McQuillan

jam@LTSP.org

Ragnar Wisløff

ragnar@wisloff.no

Copyright © 2002 av James A. McQuillan for den engelskspråklige versjonen

Copyright © 2002 av Ragnar Wisløff for den norskspråklige versjonen

Revisjonshistorie

Revisjon 1.0.4

2002–03–02

Revised by: jam

GNU/Linux er en flott plattform å bruke til tynne klienter. Hovedformålet med dette dokumentet er å vise deg hvordan man setter opp tynne klienter med LTSP. I tillegg dekker det andre emner som gjelder tynne klienter generelt.

Innholdsvertegnelse

<u>Innledning</u>	1
<u>1. Ansvarsfraskrivelse</u>	1
<u>2. Opphavsrett og lisens</u>	2
<u>Kapittel 1. Teori og prinsipper</u>	3
<u>Kapittel 2. Installasjon av LTSP på tjenermaskinen</u>	7
<u>2.1. Installasjon av RPM-pakkene</u>	7
<u>2.2. Installasjon av TGZ-pakkene</u>	7
<u>2.3. Initialisering av tjenermaskinen</u>	8
<u>2.4. Oppsett av en arbeidsstasjon</u>	8
<u>2.4.1. /etc/dhcpd.conf</u>	8
<u>2.4.2. /etc/hosts</u>	10
<u>2.4.3. /opt/lts/i386/etc/lts.conf</u>	10
<u>Kapittel 3. Oppsett av arbeidsstasjonen</u>	12
<u>3.1. Hvordan man lager en oppstartsdiskett</u>	12
<u>Kapittel 4. Start av arbeidsstasjonen</u>	14
<u>Kapittel 5. Utskrift</u>	15
<u>5.1. Oppsett på klienten</u>	15
<u>5.2. Oppsett på tjeneren</u>	15
<u>Kapittel 6. Feilsøking</u>	18
<u>6.1. Feilsøking av Etherboot diskett-oppstart</u>	18
<u>6.2. Feilsøking av DHCP</u>	18
<u>6.2.1. Sjekk alle forbindelser</u>	19
<u>6.2.2. Går tjenesten DHCP?</u>	19
<u>6.2.3. Dobbeltsjekk oppsettet av dhcpd</u>	20
<u>6.2.4. Stanses kringkastingen av iptables eller iphain?</u>	20
<u>6.2.5. Sender arbeidsstasjonen forespørselen?</u>	20
<u>6.3. Feilsøking av TFTP</u>	21
<u>6.3.1. tftpd går ikke</u>	21
<u>6.3.2. Kjernen ligger ikke der tftpd forventer å finne den</u>	21
<u>6.4. Feilsøking av NFS rot-filsystemet</u>	21
<u>6.4.1. No init found</u>	21
<u>6.4.2. Server returned error -13</u>	22
<u>6.4.3. Problemer med NFS-tjenesten (portmap, nfsd & mountd)</u>	22
<u>6.5. Feilsøking av X Window</u>	24
<u>6.6. Feilsøking av innloggings-programmet</u>	24
<u>6.6.1. Grå skjerm med en markør formet som en X</u>	24
<u>Kapittel 7. Kjerner</u>	28
<u>7.1. Standard kjerner som følger med LTSP</u>	28
<u>7.2. Bygg din egen kerne</u>	28
<u>7.2.1. Å finne kildekode for kjernen</u>	28
<u>7.2.2. Kjernepatcher</u>	29

Innholdsvertegnelse

Kapittel 7. Kjerner

7.2.3. Tilpasning av kjernen	30
7.2.4. Bygging av kjernen	32
7.2.5. Merking av kjernen for Etherboot	32

Kapittel 8. Innhold i lts.conf.....33

8.1. Eksempel på lts.conf	33
8.2. Tilgjengelige parametre i lts.conf	33
8.2.1. Generelle parametre	33
8.2.2. X-Window parametre	35
8.2.3. Parametre for berørings-skjermer	37
8.2.4. Parametre for lokale programmer	37
8.2.5. Parametre for tastatur	38
8.2.6. Parametere for bruk av skrivere	38

Kapittel 9. Lokale programmer.....40

9.1. Fordeler med å kjøre programmer lokalt	40
9.2. Ting å passe på når man setter opp lokale programmer	40
9.3. Oppsett av tjenermaskin for lokale programmer	41
9.3.1. Parametre i lts.conf	41
9.3.2. Network Information Service – NIS	41
9.4. Oppsett av programmer	41
9.5. Start av lokale programmer	43

Kapittel 10. Eksempler på oppsett.....44

10.1. Standard 2-knappers seriell mus	44
10.2. PS/2 mus med rulle	44
10.3. USB-skriver på en ThinkNic	44
10.4. Tvungen bruk av en eldre versjon av XFree86 3.3.6 Xserver	44

Kapittel 11. Andre kilder til informasjon.....45

11.1. På Internett	45
11.2. Publikasjoner på papir	45

Innledning

LTSP tilbyr en enkel måte å benytte lavkost arbeidsstasjoner, enten som grafiske eller tegnbaserte terminaler mot en GNU/Linux applikasjonstjener.

I et tradisjonelt kontormiljø benyttes Intel-baserte PC'er med relativt høy ytelse på hver pult. Hver av disse har flere gigabytes med harddisk-plass. Brukerne lagrer sine data på lokale harddisker og backup av disse gjøres sjelden eller aldri.

Er det virkelig fornuftig å ha en fullverdig datamaskin på hver pult?

Vi mener svaret er nei.

Heldigvis finnes det et alternativ. Ved å bruke LTSP kan du ta utrangerte PC'er, fjerne harddisk, diskettstasjon og CD-spiller og sette inn et nettverksort maskinen kan starte fra. Mange nettverksort har ledig plass til en brikke som gjør dette mulig.

I oppstarten får den diskløse arbeidsstasjonen sitt nettverksoppsett og en kjerne fra tjeneren og monterer rot filsystemet fra tjeneren via NFS.

Arbeidsstasjonen kan settes opp i en av tre modi:

- **Grafisk X Window**

Med X Window kan arbeidsstasjonen gis tilgang til alle programmene på tjeneren eller andre tjenere i nettverket.

- **Tegnbasert telnet-økt**

Arbeidsstasjonen kan starte flere telnet-oppkoblinger mot en tjener. Hver telnet-økt går i en virtuell terminal. Tastekombinasjonene Alt-F1 til Alt-F9 bytter mellom telnet-øktene.

- **Lokalt skall**

Arbeidsstasjonen kan settes opp til å gi deg et skall i konsollet ved oppstart. Dette er spesielt nyttig under feilsøking av problemer med X Window eller NFS.

Det virkelig fine er at du kan henge svært mange arbeidsstasjoner på en enkelt GNU/Linux tjenermaskin. Hvor mange arbeidsstasjoner? Vel, det kommer an på tjenermaskinen og på programmene som skal brukes.

Det er ikke uvanlig å ha 40 arbeidsstasjoner, alle med Netscape og Staroffice, betjent av en enkelt 2-CPU PIII-650 med 1 GB RAM. Vi vet at dette virker. Faktisk er gjennomsnittlig last sjelden over 1.0!

1. Ansvarsfraskrivelse

Hverken forfatteren eller distributøren, eller noen annen bidragsyter til dette dokumentet er på noen måte ansvarlig for fysisk, økonomisk, moralsk eller annen form for skade påført ved at forslagene i denne teksten følges.

2. Opphavsrett og lisens

Opphavsretten til dette dokumentet er fra 2001 og tilhører James McQuillan og Ragnar Wisløff. Dokumentet er offentliggjort under betingelsene i GNU Free Documentation License, som herved er inkludert ved referanse.

Kapittel 1. Teori og prinsipper

Å starte opp en diskløs arbeidsstasjon består av flere steg. Å forstå hva som skjer underveis vil gjøre det mye enklere å løse problemer som måtte oppstå.

Dette eksempelet er basert på følgende oppsett:

- Standard x86-basert arbeidsstasjon
- Linksys LNE100TX nettverkskort med Etherboot oppstartsROM.
- Intel i810-basert skjermkort
- Tjenermaskin som kjører Red Hat Linux 7.2
- DHCP
- Nettverksadresser 192.168.0.0/24

Dersom man antar at tjenermaskinen er satt opp med LTSP-pakken er dette det som skjer:

1. Når du skrur på maskinen går den gjennom sin "Power On Self Test" (POST).
2. Under egen-testen vil BIOS søke etter funksjonsbrikker. Nettverkskortet inneholder en Etherboot-brikke som er en slik utvidelsesbrikke. BIOS finner denne brikken på nettverkskortet.
3. Når POST er ferdig, overtar Etherboot-koden.
4. Etherboot-koden leter etter et nettverkskort. Hvis et slikt blir funnet, initialiseres det.
5. Etherboot-koden kringkaster så ut en DHCP-forespørsel til det lokale nettverket. Denne forespørselen inneholder MAC-adressen til nettverkskortet.
6. DHCP-daemonen på tjeneren vil fange opp denne forespørselen fra arbeidsstasjonen, og slå opp i sin liste for å se om den kjenner igjen MAC-adressen for dette nettverkskortet.
7. DHCP-daemonen bygger opp en svar-pakke som inneholder flere typer informasjon. Denne pakken sendes tilbake til arbeidsstasjonen og inneholder blant annet:
 - ◆ IP-adressen til arbeidsstasjonen
 - ◆ Nettmasken for det lokale nettverket.
 - ◆ Sti til kjernen som skal overføres til arbeidsstasjonen
 - ◆ Sti til rot-filsystemet som skal monteres
 - ◆ Andre parametre som skal gis til kjernen via kjernens kommandolinje
8. Etherboot-koden vil ta i mot svaret fra tjenermaskinen og bruke informasjonen til å sette opp nettverkskortets TCP/IP-parametre.
9. Ved hjelp av TFTP (Trivial File Transfer Protocol) vil Etherboot-koden ta kontakt med tjeneren og begynne å overføre Linux-kjernen.
10. Når kjernen er ferdig overført til arbeidsstasjonen vil Etherboot-koden legge kjernen inn på riktig sted i minnet.
11. Kontrollen overlates så til kjernen som initialiserer arbeidsstasjonen og alle komponenter den finner.

12. Her starter den virkelige moroa. I enden av kjernen er det hengt på et filsystem–bilde. Dette legges inn i minnet som en RAM–disk og blir montert midlertidig som et rot–filsystem. Kommandolinje–argumentet **root=/dev/ram0** ber kjernen om å montere bildet som rot filsystem.
13. Vanligvis vil kjernen overlate kommandoen til **init** når den er ferdig med å starte opp maskinen. Men i dette tilfellet har vi instruert kjernen om å kjøre et skallskript i stedet. For å få til dette benytter vi **init=/linuxrc** på kjernens kommandolinje.
14. Skallskriptet **/linuxrc** starter med å undersøke PCI–bussen for å se om den finner nettverkskort. For hvert PCI–kort det finner slår det opp i filen `/etc/niclist` for å se om kortet finnes der. Et treff gir som resultat drivermodulen til nettverkskortet. For ISA nettverkskort må driveren og eventuelle IRQ og IO–adresser angis eksplisitt på kommandolinjen til kjernen.
15. Når nettverkskortet er identifisert vil **/linuxrc** –skriptet legge inn kjernemodulen som støtter kortet.
16. Deretter kjøres **dhclient** for å gjøre nok en DHCP–forespørsel. Dersom vi benytter resultatet som kom fra Etherboots forespørsel, ville dette bli svelget av kjernen. I tillegg ville kjernen ignorere NFS–tjenere som måtte bli angitt i parameteren for rot filsystemet. Dette er viktig dersom du ønsker å benytte en annen maskin for NFS enn den som benyttes for TFTP.
17. Når **dhclient** får svar fra tjeneren kjører det **/etc/dhclient–script** som benytter informasjonen gitt til dhclient for å sette opp nettverkskortet eth0.
18. Opp til dette punktet har rot filsystemet vært en RAMdisk. Nå monterer **/linuxrc** et nytt rot filsystem via NFS. Katalogen som stilles til rådighet for dette på tjeneren er typisk `/opt/ltsp/i386`. Men denne kan ikke bare monteres som `/`. Først må den monteres som `/mnt`. Så utføres kommandoen **pivot_root** som bytter det nåværende filsystemet med et nytt. Når den er ferdig vil NFS–filsystemet være montert som `/` og det gamle filsystemet som `/oldroot`.
19. Når montering og bytting av rot filsystemer er jobben til skallskriptet **/linuxrc** gjort og det virkelige **init** kan kjøres.
20. Init vil lese `/etc/inittab` og begynne på oppsettet av arbeidsstasjonens miljø.
21. Init benytter prinsippet *driftsnivå*. Hvert driftsnivå setter opp et sett med tjenester for arbeidsstasjonen. Alle LTSP arbeidsstasjoner starter først i driftsnivå '2' som er satt av variabelen *initdefault* i filen `inittab`.
22. En av de første variablene i `inittab` er **rc.local**, en kommando som kjøres mens arbeidsstasjonen er i tilstanden '*sysinit*'.
23. Skriptet **rc.local** lager en RAMdisk på 1 MB som skal inneholde alt som trengs å bli skrevet til eller modifisert på noen måte.
24. RAMdisken blir montert som katalogen `/tmp`. Alle filer som skal opprettes vil befinne seg i `/tmp` og det finnes en rekke symbolske lenker som peker mot disse filene.
25. Filsystemet `/proc` monteres.
26. Hvis arbeidsstasjonen er satt opp til å benytte NFS som swap vil katalogen `/var/opt/ltsp/swapfiles` monteres som `/tmp/swapfiles` Dersom det ikke finnes en

swap-fil for arbeidsstasjonen blir en slik opprettet automatisk. Størrelsen på denne er satt i `lts.conf`.

Swapfilen vil så gjøres tilgjengelig med kommandoen **swapon**.

27. Den interne nettverksenheten **loopback** settes opp. Denne enheten har IP-adressen `127.0.0.1`.
28. Hvis arbeidsstasjonen skal benytte lokale programmer vil katalogen `/home` monteres slik at disse programmene har tilgang til brukernes hjemmeområder.
29. Flere kataloger opprettes i filsystemet under `/tmp` for å gi et hjem til endel filer som skal endres mens systemet er i drift. Kataloger som:

- a. `/tmp/compiled`
- b. `/tmp/var`
- c. `/tmp/var/run`
- d. `/tmp/var/log`
- e. `/tmp/var/lock`
- f. `/tmp/var/lock/subsys`

blir alle opprettet.

30. Nå settes X Window opp. I `lts.conf` finnes parameteren **XSERVER**. Hvis denne mangler eller er satt til "**auto**" vil en automatisk deteksjon forsøkes. Hvis skjermkortet er PCI vil leverandør og enhetsidentifikator bli funnet og det gjøres et oppslag i `/etc/vidlist`.

Hvis skjermkortet er støttet av XFree86 4.X vil **pci_scan**-rutinen gi navnet på drivermodulen. Hvis det er støttet bare av XFree86 3.3.6 vil **pci_scan** gi navnet på X-tjeneren som skal benyttes. **rc.local** vil kunne se forskjell på de to fordi de eldre 3.3.6 tjenerne starter med 'XF86_'.

31. Hvis XFree86 4.x skal benyttes vil skriptet `/etc/rc.setupx` lage en oppsettfil for X. Dersom XFree86 3.3.6 skal benyttes gjøres dette av `/etc/rc.setupx3`.

`XF86Config` lages basert på parametre i `/etc/lts.conf`.

32. Når skriptet **rc.setupx** er ferdig går kontrollen tilbake til **rc.local**. Deretter blir skriptet `/tmp/start_ws` laget, og benyttet til å starte X-tjeneren.

33. Filen `/tmp/syslog.conf` opprettes. Denne inneholder informasjon som angir for **syslogd** hvilken maskin på nettverket som skal ha loggene. En parameter i `lts.conf` styrer hvilken maskin dette er. `/etc/syslog.conf` er en symbolsk lenke som peker på `/tmp/syslog.conf`.

34. **Syslogd**-daemonen startes med oppsettfilen som ble opprettet i forrige punkt.

35. Kontrollen gis så tilbake til **init** som ser på verdien av **initdefault** for å avgjøre hvilket **driftsnivå** som skal benyttes. Fra og med versjonen `lts_core-2.08` er denne verdien **2**.

36. Driftsnivå **2** vil føre til at **init** kaller skriptet **set_runlevel** som leser `lts.conf` for å avgjøre hvilket driftsnivå arbeidsstasjonen skal ha når den er i vanlig drift.

Standard driftsnivåer for LTSP er **3**, **4** og **5**.

- ◆ **3** – Starter et kommandoskall. Nyttig for feilsøking av arbeidsstasjonen.
 - ◆ **4** – Starter en eller flere telnet-oppkoblinger i tegnbasert modus. Dette er nyttig hvis du skal erstatte gamle serielle terminaler.
 - ◆ **5** – Grafisk modus. Starter X, sender en XDMCP-forespørsel til tjeneren og viser et grafisk innloggingsbilde for å la brukeren logge seg inn på tjeneren. Du trenger en tjeneste på tjeneren som kan sende dette bildet til arbeidsstasjonen, så som **XDM**, **GDM** eller **KDM**.
-

Kapittel 2. Installasjon av LTSP på tjenermaskinen

LTSP er tilgjengelig som pakker i både *RPM* og *TGZ* formater. Velg det formatet som passer for deg og følg det tilsvarende .

Hvis du vil kjøre X på arbeidsstasjonen trenger du å overføre fire pakker til maskinen som skal brukes som tjener. Vi minner om at i dette dokumentet benytter vi som eksempel en arbeidsstasjon med et Tulip nettverkskort og et Intel i810-basert skjermkort.

1. LTSPs hovedpakke
2. Kjernepakken
3. Hovedpakken for X
4. Pakke med X skrifttyper

Pakken med X skrifttyper er ikke egentlig nødvendig, men for en førstegangsinstallasjon er det anbefalt. Når du er fortrolig med å sette opp tjenere og diskløse arbeidsstasjoner kan du lære å bruke X' skrifttype-tjener (XFS).

Etter at pakkene er installert må LTSP initialiseres. Denne prosessen innebærer å gjøre de endringer som er nødvendige i systemoppsett-filene slik at tjenermaskinen kan tilby de tjenester som trengs for at arbeidsstasjonene skal fungere.

2.1. Installasjon av RPM-pakkene

Overfør de nyeste versjonene av ltsp-pakkene og installer dem med disse kommandoene:

```
rpm -ivh ltsp_core-3.0.0-1.i386.rpm
rpm -ivh ltsp_kernel-3.0.1-1.i386.rpm
rpm -ivh ltsp_x_core-3.0.1-1.i386.rpm
rpm -ivh ltsp_x_fonts-3.0.0-0.i386.rpm
```

Kommandoene ovenfor installerer pakkene i katalogen `/opt/ltsp/i386`.

2.2. Installasjon av TGZ-pakkene

Overfør den nyeste versjonen av ltsp-pakkene og installer dem med disse kommandoene:

```
tar xzf ltsp_core-3.0.0-i386.tgz
cd ltsp_core
./install.sh
cd ..

tar xzf ltsp_kernel-3.0.1-i386.tgz
cd ltsp_kernel
./install.sh
cd ..

tar xzf ltsp_x_core-3.0.1-i386.tgz
cd ltsp_x_core
```

```
./install.sh
cd ..

tar xzf ltsp_x_fonts-3.0.0-i386.tgz
cd ltsp_x_fonts
./install.sh
cd ..
```

Kommandoene ovenfor vil installere pakkene i katalogen `/opt/ltsp/i386`.

2.3. Initialisering av tjenermaskinen

Når installasjon av pakkene er ferdig bytter du katalog til `/opt/ltsp/templates`. Her finnes flere filer som vil endre innstillingene i sentrale oppsettfiler på din tjenermaskin. Hver av disse filene er laget for å endre en enkelt systemoppsett-fil. Ta en titt på disse filene og vær sikker på at du er fortrolig med og enig i endringene de gjør. Disse endringene kan gjøre maskinen dårligere sikret mot mulige angrep. Du kan gjøre endringene manuelt i de ulike filene. Hvis du ønsker å gjøre dem automatisk kjører du kommandoen `ltsp_initialize` som superbruker:

```
cd /opt/ltsp/templates
./ltsp_initialize
```

Kommandoen over vil stille mange spørsmål om hvilke tjenester du vil endre. Den vil sette opp følgende tjenester:

- XDM – X grafisk innlogging
- GDM – Gnome grafisk innlogging
- Grafisk innlogging oppstartskript
- bootp
- NFS – filen `/etc/exports`
- tcpwrappers
- Portmapper
- syslogd
- TFTP oppstartsskript

2.4. Oppsett av en arbeidsstasjon

Nå må LTSP-tjeneren fortelles litt om arbeidsstasjonen din. Det er tre ulike filer som inneholder informasjon om arbeidsstasjonen.

1. `/etc/dhcpd.conf`
2. `/etc/hosts`
3. `/opt/ltsp/i386/etc/lts.conf`

2.4.1. `/etc/dhcpd.conf`

Arbeidsstasjonen trenger en IP-adresse og endel annen informasjon. Det får den fra DHCP-tjeneren:

- IP-adresse
- Maskinnavn
- Tjenerens IP-adresse

- Default gateway
- Sti til kjernen som skal overføres
- Maskin og sti til katalogen som skal monteres som rot filsystem

I vårt eksempel har vi valgt å benytte DHCP for tildeling av IP-adresser til arbeidsstasjoner.

Skriptet **ltsp_initialize** installerer et eksempel på en `dhcpd.conf`. Denne ligger i `/etc/dhcpd.conf.example`. Du kan kopiere denne til `/etc/dhcpd.conf` og bruke den som grunnlag for ditt endelige oppsett av DHCP. Du må manuelt endre de delene av filen som gjelder din arbeidsstasjon og tjenermaskin.

```
default-lease-time      21600;
max-lease-time          21600;

option subnet-mask      255.255.255.0;
option broadcast-address 192.168.0.255;
option routers          192.168.0.254;
option domain-name-servers 192.168.0.254;
option domain-name      "ltsp.org";
option root-path        "192.168.0.254:/opt/ltsp/i386";

shared-network WORKSTATIONS {
    subnet 192.168.0.0 netmask 255.255.255.0 {
    }
}

group {
    use-host-decl-names on;
    option log-servers 192.168.0.254;

    host ws001 {
        hardware ethernet 00:E0:18:E0:04:82;
        fixed-address 192.168.0.1;
        filename "/lts/vmlinuz.ltsp";
    }
}
```

Figur 2–1. `/etc/dhcpd.conf`

Fra LTSPs versjon 2.09pre2 av trenger du ikke lenger å spesifisere den spesifikke kjernen som skal benyttes, avhengig av nettverkskortet i arbeidsstasjonen. Standard-kjernen støtter alle nettverkskort Linux selv har støtte for. To kjerner er tatt med i LTSPs kjernepakke. En av disse har tatt inn Linux Progress Patch (LPP), den andre ikke. Navnene på disse kjernene er:

```
vmlinuz-2.4.9-ltsp-5
vmlinuz-2.4.9-ltsp-lpp-5
```

du har kanskje sett at kjernen er lagt i katalogen `/tftpboot/lts` mens spesifikasjonen i `/etc/dhcpd.conf` mangler den innledende `/tftpboot-` komponenten. Dette skyldes at Red Hat Linux versjon 7.1 og senere starter TFTP med parameteren `'-s'`. Da kjører daemonen i **sikker** modus. TFTP vil gjøre en **chroot** til `/tftpboot` når den startes og alle filer som er tilgjengelige for TFTP-daemonen er relative til katalogen `/tftpboot`.

Andre Linux-distribusjoner har kanskje ikke denne `'-s'`-opsjonen satt for TFTP, slik at du muligens må legge `/tftpboot` til spesifikasjonen av kjerne-navnet i `/etc/dhcpd.conf`.

2.4.2. /etc/hosts

Kobling av IP-adresser og maskinnavn

Generelt sett kommuniserer datamaskiner fint seg i mellom med IP-adresser. Det er oss mennesker som har behov for å sette navn på maskiner fordi vi ikke har så lett for å huske tallene i IP-adressene. Her kommer DNS og `/etc/hosts` inn i bildet. Denne koblingen mellom IP-adresser og maskinnavn er vanligvis ikke nødvendig annet enn i spesielle situasjoner, som når LTSP benyttes. Uten denne koblingen vil NFS gi feilmeldinger om manglende rettigheter når arbeidsstasjonene forsøker å montere sitt rot-filsystem.

I tillegg til NFS-problemer kan du også oppleve at de grafiske innloggingsbildene **GDM** eller **KDM** ikke virker hvis arbeidsstasjonen mangler i `/etc/hosts`.

2.4.3. /opt/ltsp/i386/etc/lts.conf

Et stort antall oppsett-parametre kan spesifiseres i `lts.conf`.

`lts.conf` har en enkel syntakse som er basert på seksjoner. **[default]** angir standard-valgene. I tillegg kan det angis avsitt som er spesielle for hver arbeidsstasjon. Navnet på disse seksjonene kan være maskinnavn, IP-adresse eller MAC-adresse.

En typisk `lts.conf` kan se ut som dette:

```
#
# Config file for the Linux Terminal Server Project (www.ltsp.org)
#

[Default]
SERVER          = 192.168.0.254
XSERVER         = auto
X_MOUSE_PROTOCOL = "PS/2"
X_MOUSE_DEVICE  = "/dev/psaux"
X_MOUSE_RESOLUTION = 400
X_MOUSE_BUTTONS = 3
USE_XFS         = N
LOCAL_APPS     = N
RUNLEVEL       = 5

[ws001]
USE_NFS_SWAP    = Y
SWAPFILE_SIZE   = 48m
RUNLEVEL       = 5

[ws002]
XSERVER         = XF86_SVGA
LOCAL_APPS     = N
USE_NFS_SWAP    = Y
SWAPFILE_SIZE   = 64m
RUNLEVEL       = 3
```

Eksempel 2–1. lts.conf

Vi tar med en kort liste over noen av mulighetene:

XSERVER

Om du har et PCI eller AGP skjermkort som er støttet av XFree86 4.1 trenger du bare `ltsp_x_core`-pakken. Denne inneholder alle driver-modulene for X4.

Et antall XFree86 3.3.6-pakker er tilgjengelige for LTSP. Bruk en av disse dersom skjermkortet ikke er støttet av XFree86 4.1.

Du kan angi parametre for hver enkelt arbeidsstasjon eller for alle samlet i `lts.conf`.

Vår arbeidsstasjon har et Intel i810 brikkesett. Dette finnes automatisk slik at vi ikke trenger å oppgi noen XSERVER parameter for kortet i `lts.conf`. Om du vil, kan du oppgi modulen til kortet eller sette parameteren til 'auto' for å vise at den skal finnes automatisk.

RUNLEVEL

Vi ønsker å bruke arbeidsstasjonen i grafisk modus, så vi setter driftsnivået til '5'. Dette gjøres med parameteren `RUNLEVEL` i `lts.conf`.

Kapittel 3. Oppsett av arbeidsstasjonen

Når serveren er ferdig, er det klart for å ta en titt på oppsettet av arbeidsstasjonen.

LTSP-prosjektet handler mye om hva som skjer etter at kjernen er plassert i minnet. Det er flere måter å få plassert kjernen der, blant annet Etherboot, Netboot, PXE og diskett.

I dette dokumentet bruker vi en oppstarts-diskett med kode fra **Etherboot**-prosjektet.

3.1. Hvordan man lager en oppstartsdiskett

Etherboot er en programvarepakke for å lage et nytt program som kan lagres i en ROM. Dette programmet kan overføre nye programmer til en x86-maskin over en Ethernet-forbindelse for å bli utført. Mange nettverkskort har en plass til slike ROM-brikker, og Etherboot er verktøyet for å plassere et program i denne brikken.

—Ken Yap

Etherboot er også fri programvare, beskyttet av GNU General Public License, versjon 2 (GPL2).

Du kan laste ned en Etherboot-pakke og tilpasse den til den typen oppstarts-ROM du trenger. Så kan du bygge kildekoden for å produsere det bildet som skal skrives til EPROM-brikken eller til diskett for testing.

En mye enklere løsning er å gå til Marty Connor's nettsted www.Rom-O-Matic.net

Marty har laget en fin nettside som gjør det enkelt å tilpasse og bygge et bootrom-bilde for Etherboot. Her kan du velge hvilket nettverkskort du har og hva slags type bilde du ønsker. I tillegg kan du modifisere mange oppsett-parametre. Når du så trykker på 'Get ROM' blir programmet satt sammen og overført til din maskin som en enkelt fil.

Vår arbeidsstasjon har et Linksys LNE100TX, versjon 4.1 nettverkskort. Dette benytter et AMDTek Centaur-P brikkesett. Vi må derfor velge *centaur-p* som vår NIC/ROM.

Vi trenger ikke gjøre noen endringer i forhold til standard-oppsettet så du kan se bort fra 'Configure'-knappen.

Velg 'Floppy Bootable ROM Image' som format for filen du produserer. Det vil legge inn en 512 byte lang blokk i begynnelsen av filen, eller bildet, denne benyttes for å legge inn oppstartskode i minne og sørge for at det utføres ved start av maskinen.

Klikk på knappen merker 'Get ROM'. Oppstartsbildet produseres mens du venter. Prosessen tar bare noen sekunder og når den ferdig vil nettleseren vise et vindu som lar deg lagre filen på det stedet du ønsker på maskinen din.

Jeg lagrer vanligvis filen i /tmp-katalogen. I dette tilfelle vil det bety en filspesifikasjon som /tmp/eb-5.0.2-centaur-p.1zdisk

Etter at filen er lagret på disk må du skrive den til en diskett. Sett inn en diskett og gi følgende kommando for å overføre filen:

```
cat /tmp/eb-5.0.2-centaur-p.lzdisk >/dev/fd0
```

Kapittel 4. Start av arbeidsstasjonen

Hvis vi antar at både tjenermaskinen og arbeidsstasjonen er riktig satt opp vil det bare være nødvendig å sette inn oppstartsdisketten og skru på arbeidsstasjonen.

Etherboot-programmet leses fra diskett og legges i minnet, nettverkskortet blir funnet, DHCP-forespørselen sendes ut på nettverket og et svar returneres fra tjeneren etterfulgt av Linux-kjernen som blir overført fra tjeneren til arbeidsstasjonen. Når kjernen har startet opp komponentene i arbeidsstasjonen vil X starte opp og et innloggingsvindu dukke opp på skjermen. Dette kan se ut som i eksempelet under dersom du benytter XDM:



Figur 4–1. Login screen

Nå kan du logge inn. Det er viktig å huske at du logger deg på tjeneren. Alle kommandoer utføres på tjeneren, det er bare resultatene som vises på din lokale skjerm på arbeidsstasjonen. Nettverksorienteringen til X Window gjør dette mulig.

Du kan kjøre samtlige programmer som er lagt inn på tjeneren.

Kapittel 5. Utskrift

I tillegg til at arbeidsstasjonen er en fullt funksjonell grafisk eller tekstbasert terminal kan den også være en utskriftssentral med opp til tre skrivere koblet til parallell- eller seriellporter.

Denne funksjonen er helt usynlig for brukeren, som ikke vil merke den lille trafikken som går gjennom arbeidsstasjonen til skriveren(e).

5.1. Oppsett på klienten

LTSP bruker programmet **lp_server** på arbeidsstasjonen for å sende utskriftsjobber fra tjeneren til en skriver koblet til en av utgangene på arbeidsstasjonen.

For å gjøre utskriftsfunksjonen tilgjengelig på arbeidsstasjonen benyttes et sett med parametre i `lts.conf`.

```
[ws001]
    PRINTER_0_DEVICE = /dev/lp0
    PRINTER_0_TYPE   = P
```

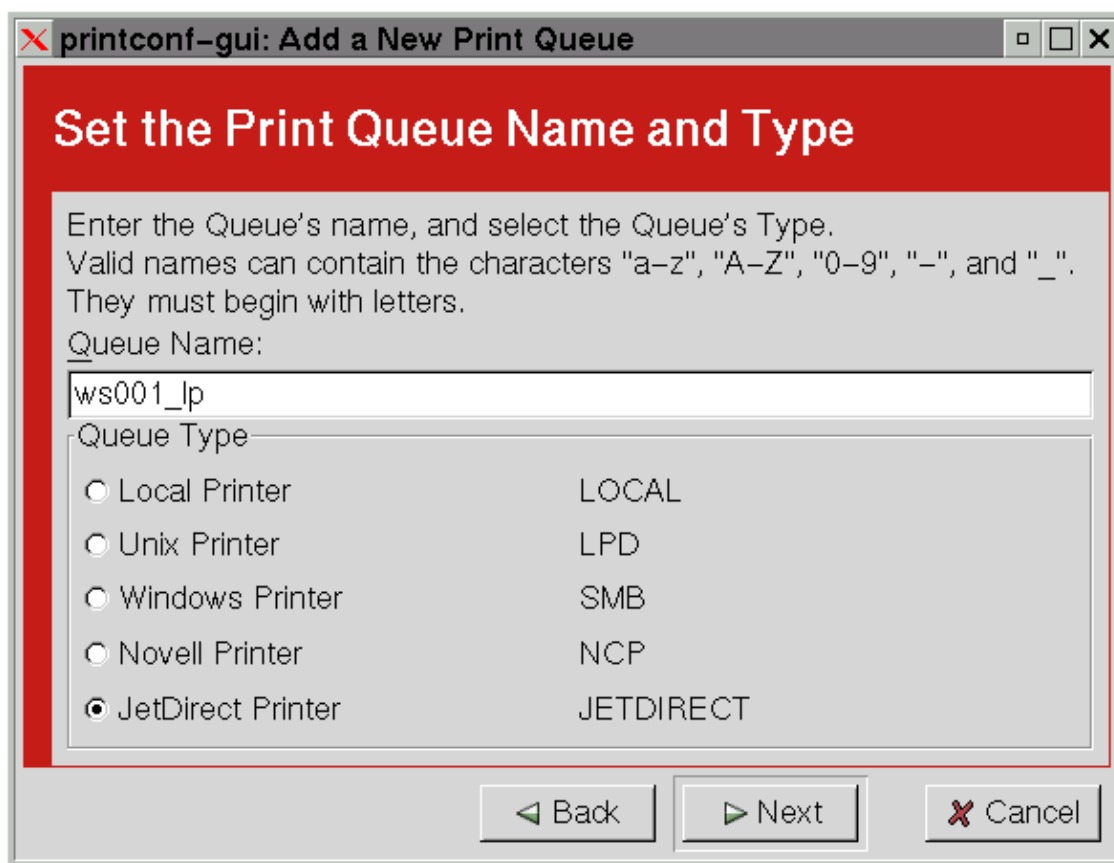
Disse linjene starter `lp_server` som en bakgrunnsprosess. Den lytter på TCP/IP-porten 9100 for å motta utskriftsjobber fra tjeneren. Utskriftsdataene som mottas blir sendt videre til parallellporten `/dev/lp0`.

Det er mange flere valg tilgjengelig. Ta en titt i avsnittet om `lts.conf` lenger ut i håndboka for mer informasjon om hvilke muligheter som finnes for bruk av skrivere og hvordan de settes opp.

5.2. Oppsett på tjeneren

Skriverstøtte på tjeneren krever at det settes opp en skriverkø. Det er ofte verktøy for dette på tjenermaskinen.

I Red Hat Linux 7.2 og senere er det både et grafisk og et tekstbasert verktøy for å klargjøre skrivere. Det grafiske er **printconf-gui**, og det tekstbaserte heter **printconf-tui**. Eldre versjoner av Red Hat Linux har programmet **printtool**, som i og for seg også finnes i versjon 7.2, men dette kaller bare **printconf-gui**. Andre distribusjoner har sine egne verktøy for å sette opp skrivere.



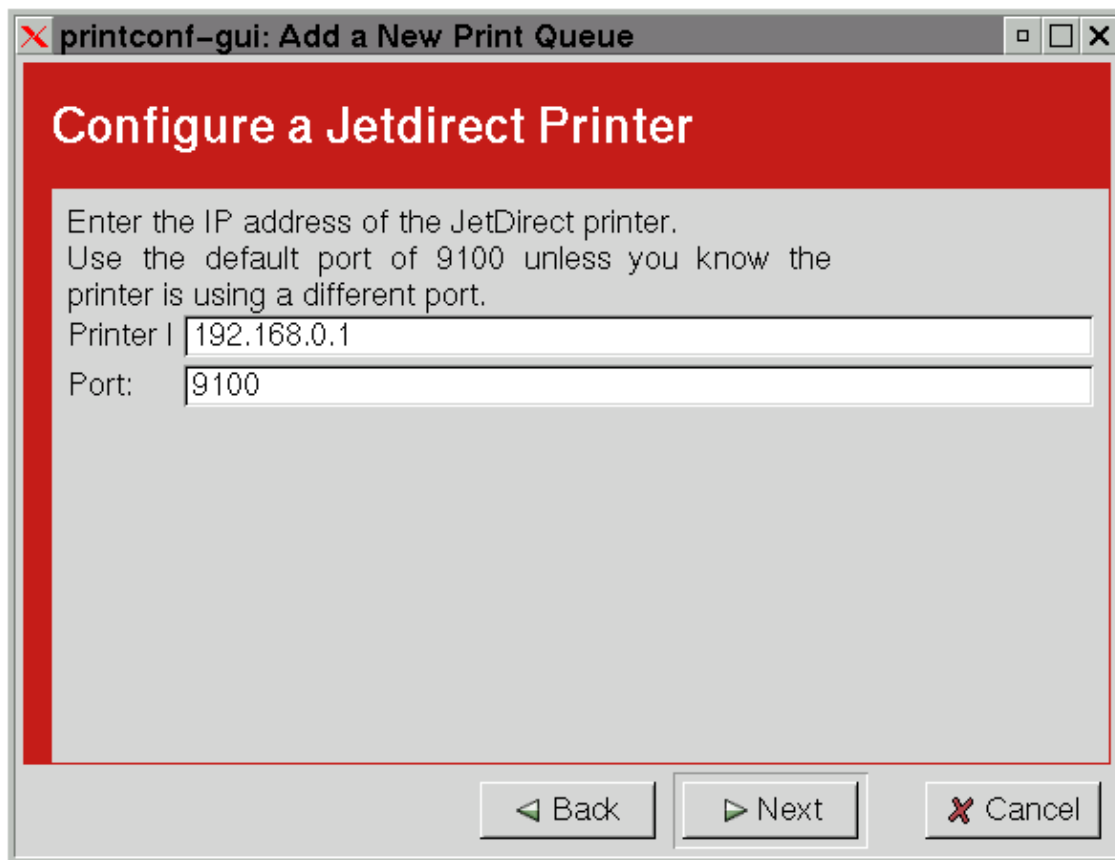
Figur 5–1. Printconf-gui – legg til ny skriver

Etter å ha startet **printconf-gui** må du legge til en ny skriver. La denne være en **JetDirect** –skriver. Programmet **lp_server** får arbeidsstasjonen til å se ut som en HP JetDirect utskriftssentral.

Du må gi skriveren et kø-navn. Dette kan være hva som helst, men la det være meningsfylt. Navnet kan inneholde bare disse tegnene:

- "a-z" små bokstaver
- "A-Z" store bokstaver
- "0-9" tall
- "-" bindestrek
- "_" understrek

Navnet i eksempelet over er **ws001_lp** og indikerer at skriveren er tilkoblet arbeidsstasjonen **ws001**.



Figur 5–2. Printconf-gui – nettverksparametre

To felter er nødvendige for å kommunisere med skriveren:

1. IP-adressen eller maskinnavnet til arbeidsstasjonen som skriveren er koblet til.
2. TCP-porten som programmet **lp_server** lytter på.

Den første skriveren som kobles til en arbeidsstasjon vil bruke TCP/IP-porten **9100**. Den andre skriveren finnes på **9101**, og den tredje på **9102**.

Kapittel 6. Feilsøking

Hvis arbeidsstasjonen ikke kommer opp, til tross for at du har fulgt instruksjonene til punkt og prikke, må du til med å feilsøke installasjonen.

Det første du må gjøre er å finne ut hvor langt maskinen kommer i oppstartsprosessen før noe går galt.

6.1. Feilsøking av Etherboot diskett–oppstart

Når du starter fra diskett bør du se noe som ligner på dette:

```
loaded ROM segment 0x0800 length 0x4000 reloc 0x9400
Etherboot 5.0.1 (GPL) Tagged ELF for [LANCE/PCI]
Found AMD Lance/PCI at 0x1000, ROM address 0x0000
Probing...[LANCE/PCI] PCnet/PCI-II 79C970A base 0x1000, addr 00:50:56:81:00:01
Searching for server (DHCP)...
<sleep>
```

Eksempelet over viser hva du kan forvente å se på skjermen på arbeidsstasjonen når du starter den fra diskett. Linjene indikerer at Etherboot er startet. Hvis de ikke er der kan det være du har en ødelagt floppy eller at programmet som ble skrevet til den enten hadde en feil eller ikke ble skrevet riktig til disketten.

Hvis du ser noe som ligner på teksten under, tyder det på at programmet som ligger på disketten ikke er det riktige for ditt nettverkskort.

```
ROM segment 0x0800 length 0x8000 reloc 0x9400
Etherboot 5.0.2 (GPL) Tagged ELF for [Tulip]
Probing...[Tulip]No adapter found
<sleep>
<abort>
```

Hvis oppstarten kommer til punktet der nettverkskortet finnes og den riktige MAC–adressen vises kan man anta at disketten er god.

6.2. Feilsøking av DHCP

Når nettverkskortet er initialisert vil det kringkaste en DHCP–forespørsel på lokalnettverket for å finne en DHCP–tjener.

Hvis arbeidsstasjonen får et gyldig svar fra en DHCP–tjener vil informasjonen i svaret brukes til å sette opp parametrene for nettverkskortet. Hvis denne operasjonen gikk bra vil IP–adressen som gis til nettverkskortet vises på skjermen. Denne meldingen man se slik ut:

```
ROM segment 0x0800 length 0x4000 reloc 0x9400
Etherboot 5.0.1 (GPL) Tagged ELF for [LANCE/PCI]
Found AMD Lance/PCI at 0x1000, ROM address 0x0000
Probing...[LANCE/PCI] PCnet/PCI-II 79C970A base 0x1000, addr 00:50:56:81:00:01
Searching for server (DHCP)...
<sleep>
Me: 192.168.0.1, Server: 192.168.0.254, Gateway 192.168.0.254
```

Hvis du ser en linje som starter med 'Me:' etterfulgt av en IP-adresse kan du gå ut fra at DHCP virker. Du kan da gå videre til å finne ut om TFTP fungerer.

Hvis du i stedet ser teksten under, etterfulgt av mange <sleep>-meldinger er noe galt. Noen få slike meldinger er normalt før DHCP-tjeneren svarer.

```
Searching for server (DHCP)...
```

Å finne ut hva som er galt er ofte vanskelig, men her er noen punkter å sjekke:

6.2.1. Sjekk alle forbindelser

Er arbeidsstasjonen fysisk koblet til samme nettverk som tjenermaskinen?

Vær sikker på at det er lys i alle dioder som viser forbindelse mellom nettverksutstyret.

Hvis du kobler arbeidsstasjonen og tjenermaskinen direkte (uten switch eller hub) må du være sikker på at du bruker en krysset kabel. Hvis du kobler via en switch eller hub, må du være sikker på at kablene ikke er krysset – både mellom arbeidsstasjonen og switch/hub, og mellom switch/hub og tjener-maskinen.

6.2.2. Går tjenesten DHCP?

Det er helt nødvendig at prosessen **dhcpcd** kjører på tjeneren. Det er et par måter å finne ut dette på.

dhcpcd går vanligvis i bakgrunnen og lytter på UDP port 67. Prøv med **netstat**-kommandoen for å se om noe virkelig lytter på denne porten:

```
netstat -an | grep ":67 "
```

Du bør se et resultat omtrent som:

```
udp      0      0  0.0.0.0:67          0.0.0.0:*
```

Den fjerde kolonnen viser IP-adressen og porten, atskilt av et kolon (':'). En adresse med bare nuller ('0.0.0.0') indikerer at den lytter på alle nettverks-enheter. Hvis du har to nettverkskort, **eth0** og **eth1**, vil **dhcpcd** lytte på begge.

Selv om **netstat** viser at noe lytter på port 67, er det ikke sikkert at det er **dhcpcd**. Det kan være **bootpd**, selv om det er usannsynlig siden bootp ikke lenger er del av de mest vanlige Linux-distribusjonene.

For å være sikker på at **dhcpcd** kjører, kan du bruke kommandoen **ps**.

```
ps aux | grep dhcpcd
```

Du burde få et resultat omtrent som dette:

```
root 23814 0.0 0.3 1676 820 ?        S 15:13 0:00 /usr/sbin/dhcpcd
root 23834 0.0 0.2 1552 600 pts/0    S 15:52 0:00 grep dhcpcd
```

Den første linjen viser at **dhcpcd** kjører. Den andre linjen er bare vår **grep**-kommando som ikke er avsluttet.

Hvis du ikke ser noen linje med **dhcpcd** må du sjekke at tjeneren er satt opp for å starte i driftsnivå 5 og at **dhcpcd** automatisk vil starte på dette driftsnivået. På Red Hat Linux-baserte maskiner kan du bruke

kommandoen **ntsysv** og bla nedover til du finner dhcpd og verifisere at tjenesten er satt til å starte. Du kan prøve å starte den med denne kommandoen: **service dhcpd start**. Følg med på hva som blir skrevet ut som respons, den kan inneholde feilmeldinger.

6.2.3. Dobbeltsjekk oppsettet av dhcpd

Er det et avsnitt for arbeidsstasjonen i `/etc/dhcpd.conf`?

Du bør sjekke at 'hardware ethernet' stemmer fullstendig overens med nettverkskortets MAC-adresse, og at 'fixed-address' er riktig.

6.2.4. Stanses kringkastingen av iptables eller ipchains?

6.2.4.1. Sjekking av ipchains

Kjør denne kommandoen og se hva resultatet er:

```
ipchains -L -v
```

Hvis du ser noe som ligner på dette:

```
Chain input (policy ACCEPT: 229714 packets, 115477216 bytes):
Chain forward (policy ACCEPT: 10 packets, 1794 bytes):
Chain output (policy ACCEPT: 188978 packets, 66087385 bytes):
```

kan du være rimelig sikker på ikke ipchains ikke står i veien for pakkene.

6.2.4.2. Sjekking av iptables

Kjør denne kommandoen og se hva resultatet er:

```
iptables -L -v
```

Hvis du ser noe som ligner på dette:

```
Chain INPUT (policy ACCEPT 18148 packets, 2623K bytes)
  pkts bytes target      prot opt in      out     source           destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target      prot opt in      out     source           destination
Chain OUTPUT (policy ACCEPT 17721 packets, 2732K bytes)
  pkts bytes target      prot opt in      out     source           destination
```

stanser ikke iptables dhcp-forespørselen.

6.2.5. Sender arbeidsstasjonen forespørselen?

Prøv å følge meldingene i systemloggen `/var/log/messages` på tjenermaskinen mens arbeidsstasjonen starter opp. Du kan få en kontinuerlig oppdatering av loggen med kommandoen:

```
tail -f /var/log/messages
```

Dette legger inn nye linjer i loggen etter hvert som de ulike programmene skriver til den.

```
server dhcpd: DHCPDISCOVER from 00:50:56:81:00:01 via eth0
server dhcpd: no free leases on subnet WORKSTATIONS
server dhcpd: DHCPDISCOVER from 00:50:56:81:00:01 via eth0
server dhcpd: no free leases on subnet WORKSTATIONS
```

Hvis du ser meldinger som dem over, med teksten 'no free leases', indikerer det at **dhcpd** går på tjeneren, men ikke kjenner igjen arbeidsstasjonen som ber om en IP-adresse. F.eks. kan det være at MAC-adressen ikke stemmer.

6.3. Feilsøking av TFTP

Etherboot bruker TFTP for å overføre en Linux-kjerne fra tjener-maskinen. Dette er en relativt enkel protokoll, men noen ganger kan det være vanskelig å få den til å virke.

Hvis du ser en melding som ligner på denne:

```
Loading 192.168.0.254:/lts/vmlinuz-2.4.9-ltsp-5 |
```

der det siste tegnet '|' veksler mellom '|', '\', '-' and '/', slik at det ser ut som en roterende stolpe, er det et tegn på at kjernen blir overført. Da virker TFTP som den skal.

Hvis du ikke ser denne roterende stolpen kan du ha et av følgende problem:

6.3.1. tftpd går ikke

Red Hat Linux 7.1 startes **tftpd** av **xinetd**. Informasjon om hvordan **tftpd** skal fungere finnes i filen `/etc/xinetd.d/tftp`.

6.3.2. Kjernen ligger ikke der tftpd forventer å finne den

Kjernen må være lagt på et sted der **tftpd** kan få tak i den. Hvis '-s' er spesifisert når **tftpd** kjøres må alle filspesifikasjoner som sendes til **tftpd** være relative til `/tftpboot`. Så hvis linjen **filename** i `/etc/dhcpd.conf` angir `/lts/vmlinuz-2.4.9-ltsp-5`, så må kjernen befinne seg i `/tftpboot/lts/vmlinuz-2.4.9-ltsp-5`

6.4. Feilsøking av NFS rot-filsystemet

Det er flere ting som kan hindre rot-filsystemet i å bli montert. Blant disse er:

6.4.1. No init found

Hvis du får denne feilmeldingen:

```
Kernel panic: No init found. Try passing init= option to kernel.
```

er det sannsynlig at du monterer feil katalog som rot-filsystem, eller at `/opt/ltsp/i386`-katalogen er tom.

6.4.2. Server returned error –13

Hvis du får denne feilmeldingen:

```
Root-NFS: Server returned error -13 while mounting /opt/ltsp/i386
```

indikerer den at katalogen `/opt/ltsp/i386` ikke er tatt inn i NFS oppsett-filen `/etc/exports`.

Kikk i systemloggen `/var/log/messages` på tjenermaskinen for å se om det er noen spor. En linje som dette:

```
Jul 20 00:28:39 jamlap rpc.mountd: refused mount request from ws004
for /opt/ltsp/i386 (/): no export entry
```

bekrefter vår mistanke om at det er en feil i `/etc/exports`.

6.4.3. Problemer med NFS-tjenesten (portmap, nfsd & mountd)

NFS er en tjeneste det kan være en kompleks og vanskelig oppgave å feilsøke. Derfor er det viktig å vite hvordan de involverte tjenestene skal settes opp og hvilke verktøy som er tilgjengelige for å stille en diagnose, slik at denne jobben blir så enkel som mulig.

Tre tjenester kjører i bakgrunnen (daemoner) for å få NFS til å fungere riktig. Disse er **portmap**, **nfsd** and **mountd**.

6.4.3.1. portmap

Hvis du får disse feilmeldingene:

```
Looking up port of RPC 100003/2 on 192.168.0.254
portmap: server 192.168.0.254 not responding, timed out
Root-NFS: Unable to get nfsd port number from server, using default
Looking up port of RPC 100005/2 on 192.168.0.254
portmap: server 192.168.0.254 not responding, timed out
Root-NFS: Unable to get mountd port number from server, using default
mount: server 192.168.0.254 not responding, timed out
Root-NFS: Server returned error -5 while mounting /opt/ltsp/i386
VFS: unable to mount root fs via NFS, trying floppy.
VFS: Cannot open root device "nfs" or 02:00
Please append a correct "root=" boot option
Kernel panic: VFS: Unable to mount root fs on 02:00
```

er det mest sannsynlig at **portmap** ikke kjører. Du kan undersøke dette ved å bruke kommandoen **ps**:

```
ps -e | grep portmap
```

Hvis **portmap** går vil du se et resultat som dette:

```
30455 ?          00:00:00 portmap
```

netstat kan også benyttes for å sjekke **portmap**, som bruker port 111 for både UDP og TCP. Prøv denne kommandoen:

```
netstat -an | grep ":111 "
```

som skal gi resultatet:

```
tcp    0    0 0.0.0.0:111      0.0.0.0:*        LISTEN
udp    0    0 0.0.0.0:111      0.0.0.0:*
```

Hvis du ikke ser noe som ligner på dette kan du anta at tjenesten **portmap** ikke kjører. Den kan startes med denne kommandoen:

```
/etc/rc.d/init.d/portmap start
```

For å være sikker på at **portmap** starter når maskinen startes kan du bruke **ntsysv**. Finn **portmap** og sjekk at det er krysset av for at den skal starte.

6.4.3.2. Bakgrunnsprogrammene NFS og MOUNT (nfsd & mountd)

NFS selv har to bakgrunnsprogrammer (daemoner) som må være i drift for å sikre at NFS fungerer: **nfsd** og **mountd**. Begge startes fra skriptet `/etc/rc.d/init.d/nfs`.

Du kan bruke kommandoen **ps** for å sjekke om de kjører.

```
ps -e | grep nfs
ps -e | grep mountd
```

Hvis resultatet viser at en eller begge programmene ikke er i drift, må du starte dem.

Opptarts-skriptet er ment å skulle starte begge de nødvendige programmene hvis kommandolinje-argumentet benyttes slik: **service nfs restart**. Av en eller annen grunn (en feil?) virker dette ikke for dette skriptet. Det er bare **mountd** som startes, mens **nfsd** ikke startes med denne kommandoen. Derfor bør du gjøre en slik omstart av NFS i to steg:

```
/etc/rc.d/init.d/nfs stop
/etc/rc.d/init.d/nfs start
```

Du kan få feilmeldinger med den første (**stop**), men det er greit. Den andre linjen skal vise **OK** som resultat.

Hvis programmene virkelig er i gang, men NFS fremdeles ikke virker, kan du verifisere at NFS-programmene er registrert av **portmap** med kommandoen **rpcinfo**.

```
rpcinfo -p localhost
```

Her skal du se et resultat som ligner på

```
program vers proto  port
100000    2    tcp    111  portmapper
100000    2    udp    111  portmapper
100011    1    udp    856  rquotad
100011    2    udp    856  rquotad
100005    1    udp    1104 mountd
100005    1    tcp    2531 mountd
100005    2    udp    1104 mountd
100005    2    tcp    2531 mountd
100003    2    udp    2049 nfs
```

Dette indikerer at både **nfs** (nfsd) og **mountd** er i gang og har meldt seg til tjeneste hos **portmap**.

6.5. Feilsøking av X Window

Ojoj! Dette er sannsynligvis den vanskeligste delen av å sette opp en LTSP arbeidsstasjon: å få X Window til å virke skikkelig. Hvis du har et relativt nytt skjermkort i arbeidsstasjonen og dette er støttet av XFree86 trenger det ikke være problematisk. Dersom også skjermen kan håndtere et relativt stort antall frekvenser og oppløsninger bør oppsettet gå smertefritt. Vanligvis er problemet at valg av driver eller XFree86 tjenerversjon ikke stemmer overens med kortet.

At en X-tjener ikke virker med skjermkortet er vanligvis ganske åpenbart. Enten starter ikke X i det hele tatt, eller så er skjermen helt forvrengt.

Når arbeidsstasjonen er klar til å starte X Window vil den kjøre skallskriptet `/tmp/start_ws`, som starter X lokalt på arbeidsstasjonen. Oppstarten angir at X skal forsøke å koble opp mot maskinen angitt av `-query` hvor et innloggingsprogram, f.eks. **XDM**, **GDM** eller **KDM** lytter på forespørsler fra maskiner på nettverket.

Gangen i å starte X på en LTSP arbeidsstasjon er at skriptet `start_ws` blir kjørt av **init**, og i sin tur starter X. Hvis `start_ws` ikke greier å starte X, vil **init** gjøre et nytt forsøk. Ti ganger vil **init** forsøke å få i gang X på denne måten, deretter gir det opp fordi det tror at det blir restartet for fort. Den feilmeldingen som angir grunnen til at X ikke vil starte kan nå ses på skjermen.

Å vente på at X skal gjøre ti feilslåtte forsøk på å starte kan være ganske irriterende. En enkel måte å unngå dette på er å starte arbeidsstasjonen i driftsnivå 3, slik at X ikke startes automatisk. I stedet får du opp en **bash** kommandolinje. Fra kommandolinjen kan du starte X manuelt med kommandoen

```
sh /tmp/start_ws
```

X gjør et forsøk på å starte, og når forsøket feiler vil du få tilbake kommandolinjen. Feilmeldingene vil være skrevet ut på skjermen slik at det er mulig å avgjøre hva årsaken til feilen er.

6.6. Feilsøking av innloggings-programmet

Programmet som viser innloggingsbildet på arbeidsstasjonens skjerm er en bakgrunnsprosess på tjeneren. Denne lytter etter forespørsler fra X på andre maskiner som prøver å sette opp en forbindelse. Hvis denne forbindelsen er etablert vil et innloggingsbilde bli vist på skjermen, slik at brukeren kan autentisere seg og få tilgang til maskinen.

De tre mest vanlige innloggings-programmene er:

- XDM – som har vært i bruk så lenge noen kan huske. Den følger med som standard i X – pakken.
- GDM – 'Gnome Display Manager'. Del av arbeidsflaten Gnome.
- KDM – 'KDE Display Manager'. Del av skrivebordsflaten KDE.

De fleste GNU/Linux-distribusjonene har med alle disse.

6.6.1. Grå skjerm med en markør formet som en X

Dette er en indikasjon på at X er oppe og kjører, men at det ikke er opprettet kontakt med et innloggings-program. Noen vanlige årsaker til dette kan være:

1. Innloggings-programmet er ikke i gang.

På nylige versjoner av Red Hat Linux (7.0 og senere) startes innloggingsbildet fra **init**. I fila `/etc/inittab` er det en linje som ser slik ut:

```
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Skriptet **prefdm** avgjør hvilket program som vil håndtere innloggingen.

Hvilket innloggingsbilde som benyttes er avhengig av hvilke pakker som er installert. Dersom Gnome er lagt inn vil GDM være standard. Dersom Gnome ikke er installert vil skriptet **prefdm** sjekke om KDE er installert. Hvis det er, vil KDM settes som standard innloggingsbilde. Hvis ikke KDE heller er installert, vil XDM settes som innloggingsbilde.

Ved å bruke kommandoen **netstat** kan du se om programmet som styrer innloggingsbildet er i gang. Bruk denne kommandoen på tjenermaskinen:

```
netstat -ap | grep xdmcp
```

Du burde få respons som viser at det går en prosess som lytter på port 177, xdmcp-protokollens port.

```
udp        0      0 *:*xdmcp          *:*                1493/gdm
```

Her ser vi at **gdm** kjører med prosess-nr. 1493 og lytter på xdmcp-porten.

Hvis du får et resultat som det over, og kan være rimelig sikker på at det riktige programmet kjører på tjenermaskinen, må du flytte fokus til arbeidsstasjonen. Du må være sikker på at denne sender ut xdmcp-forespørsler til riktig tjener.

I `lts.conf` kan man angi hvilken IP-adresse som benyttes av maskinen som sender innloggingsbildet til arbeidsstasjonen. Dersom man velger å angi denne IP-adressen eksplisitt (det er valgfritt), er formatet på den aktuelle linjen:

```
XDM_SERVER = 192.168.0.254
```

IP-adressen i dette eksempelet er ikke nødvendigvis den samme som i ditt nettverk og for dine maskiner.

Hvis 'XDM_SERVER' ikke er eksplisitt oppgitt, vil ip-adressen antas å være den samme som er angitt for 'SERVER'. Hvis denne heller ikke er oppgitt benyttes **192.168.0.254**.

Uansett hvordan den er angitt, må du bare sørge for at den IP-adressen som benyttes av arbeidsstasjonen når den sender ut forespørselen om innloggingsbilde er den samme som den som gjelder for tjeneren som leverer innloggingsbildet.

2. Programmet som styrer innloggingsbildet kan være satt opp til å overse forespørsler fra andre maskiner i nettverket.

Hvis du er sikker på at programmet kjører og at arbeidsstasjonen benytter riktig IP-adresse for forespørslerne, kan det være at programmet ikke reagerer på XDMCP-forespørsler fra arbeidsstasjonen. Du må sjekke oppsettet for den aktuelle pakken (GDM, KDM eller XDM) for å finne ut om den faktisk håndterer forespørsler fra andre maskiner.

◆ XDM

Standard oppsett for Red Hat Linux er å hindre tilgang til innloggingsbildet fra andre maskiner. Når du kjører tilpasningsprogrammet **ltsp_initialize** vil de nødvendige endringene bli gjort for å gi slik tilgang. Hvis systemet allikevel ikke fungerer bør du sjekke innstillingene i `/etc/X11/xdm/xdm-config`. Let etter en linje som ligner på denne:

```
DisplayManager.requestPort: 0
```

Denne linjen MÅ være kommentert ut for at XDM skal lytte på port 177 for innkommende forespørsler fra andre maskiner.

En annen oppsettfil er også viktig for at XDM skal kunne levere innloggingsbildet. Dette er `/etc/X11/xdm/Xaccess` som MÅ inneholde en linje som starter med en stjerne '*'. Red Hat Linux kommer med denne linjen kommentert ut, og **ltsp_initialize** skal fjerne kommentar-tegnet. Hvis fjerninnloggingen allikevel ikke fungerer bør du kontrollere denne oppsettfilen. Linjen bør se omtrent slik ut:

```
* #any host can get a login window
```

◆ KDM

Nyere versjoner av KDM har egen oppsettfil, **kdmrc**. De ulike Linux-distribusjonene plasserer denne filen på forskjellige steder. Red Hat Linux 7.2 har den i `/etc/kde/kdm/kdmrc`. For å finne den i andre distribusjoner bør du bruke kommandoen **locate kdmrc**.

I `kdmrc` er det avsnittet som starter med **[Xdmcp]** som styrer om et innloggingsbilde sendes til andre maskiner. Sørg for at **Enable** er satt til **true** i dette avsnittet.

Eldre versjoner av KDM bruker XDMs oppsettfiler. Disse finner du i `/etc/X11/xdm`.

◆ GDM

GDM bruker et eget sett med oppsettfiler. Disse er lagt i katalogen `/etc/X11/gdm`.

Den viktigste filen er `gdm.conf`. Finn avsnittet **[xdmcp]**. Her finner du parameteren 'Enable', denne må være satt til '1' eller 'True', avhengig av hvilken versjon av GDM som er i bruk. Her er et eksempel:

```
[xdmcp]
Enable=true
HonorIndirect=0
MaxPending=4
MaxPendingIndirect=4
MaxSessions=16
MaxWait=30
MaxWaitIndirect=30
Port=177
```

Merk linjen med 'Enable=true'. Eldre versjoner av GDM bruker '0' og '1' som verdier for henholdsvis 'false' og 'true'. Nyere versjoner benytter 'false' og 'true'.

3. Hvis programmet som styrer innloggingsbildet kjører og lytter, på riktig port, etter forespørsler fra andre maskiner kan det være et nettverksproblem som hindrer at bildet kommer opp på

arbeidsstasjonens skjerm. Programmet må kunne gjøre om IP-adressen til et maskinnavn, det gjøres som regel med filen `/etc/hosts` der det må finnes en linje med IP-adressen og maskinnavnet til den aktuelle arbeidsstasjonen. Alternativt må arbeidsstasjonen være satt opp i DNS-tabellene.

Kapittel 7. Kjerner

Man må ta noen avgjørelser når det gjelder kjernene som skal benyttes på arbeidsstasjonene. Først og fremst gjelder dette om du vil bruke en av de ferdiglagde kjernene som følger med LTSP, eller om du vil bygge egne. Du må også bestemme deg for om du vil vise grafikk fremfor tekst under oppstarten av arbeidsstasjonen – med bruk av **Linux Progress Patch (LPP)**.

7.1. Standard kjerner som følger med LTSP

LTSP tilbyr to ferdige kjerner – en med LPP, en uten.

Begge kjernene har patchen for NFS Swap innlagt.

7.2. Bygg din egen kjerne

Det er to måter å lage kjerner for LTSP. Standardmetoden er å benytte noe som kalles en 'initiell ramdisk', eller forkortet **initrd**. Denne ramdisken er et lite filsystem som er hengt på kjernen. Initrd'en legges inn i minnet og så snart kjernen er startet vil den montere opp dette filsystemet. Det er noen viktige fordeler med initrd. For det første kan man bygge nettverksdriverne som moduler og legge inn riktig modul ved oppstart. En slik løsning gjør det mulig å ha en enkelt kjerne for så godt som alle typer nettverkskort. For det andre kan man kjøre DHCP-klienten som et bruker-program i stedet for et kjernebasert. I bruker-modus har man flere muligheter for å be om og ta i mot parametre fra DHCP-tjeneren. Det gjør også kjernen litt mindre. Den andre muligheten er å lage en kjerne uten en initrd. Driveren for det aktuelle nettverkskortet må da bygges inn i selve kjernen sammen med IP-autoconfig og 'rot-filsystem over NFS'. Denne framgangsmåten gjør kjernen litt mindre og den vil starte opp litt raskere. Når arbeidsstasjonen er oppe og går er det så godt som ingen forskjell i hvordan arbeidsstasjonen virker.

Standard LTSP-kjerne har en initiell ramdisk som tar seg av gjenkjenning av nettverkskortet og DHCP-forespørsler. Et viktig kriterium for initrd'en var å gjøre den så liten som mulig. Derfor valgte vi uClinux' libc-erstatning og busybox som verktøyene vi trenger under oppstart.

Hvis du vil lage dine egne kjerne, trenger du å overføre pakken `ltspp_initrd_kit` til din lokale maskin. Pakken inneholder det nødvendige hierarkiet til rot-filsystemet og et skript for å bygge initrd'en.

7.2.1. Å finne kildekoden for kjernen

Det er vanligvis en god ide å starte med fersk kildekode fra **ftp.kernel.org** når man skal bygge kjerne. En av grunnene til dette er at de fleste distribusjonene, som f.eks. Red Hat, legger inn en rekke patcher i 'sine' kjerne. Disse er et stykke fra den offisielle kjernens kildekode.

Overfør pakken med kildekoden for kjernen du vil bruke og lagre den i katalogen `/usr/src`. Kjernen finnes under `/pub/linux/kernel` på ftp-tjeneren til `ftp.kernel.org`. Bruk en relativt moderne 2.4.x-kjerne fordi du trenger funksjonaliteten i **devfs**.

Dersom du også vil ha mulighet til å benytte NFS til swap eller Linux Progress Patch (LPP) må du finne patchene til disse funksjonene. Patchene må stemme overens med kjernen du skal bygge. I skrivende stund er 2.4.9 siste kjerne med støtte for disse funksjonene.

Vi vil i dette eksempelet benytte kjerneversjon 2.4.9. URL'en til denne er
`ftp://ftp.kernel.org/pub/linux/kernel/v2.4/linux-2.4.9.tar.bz2`

Pakk ut kildekoden i katalogen `/usr/src`. Her må du passe litt på, for når du pakker ut vil det lages en ny katalog med navnet `linux`. Hvis du allerede har denne katalogen med kildekode til kjernen, vil den bli overskrevet. Det er nok ikke det du vil... Gi nytt navn til en eventuell `/usr/src/linux`-katalog før du pakker ut kildekoden til kjernen.

Kildekode-pakken vi hentet er komprimert med verktøyet **bzip2**. Derfor må vi dekomprimere den før den sendes til utpakking med **tar**. Du kan bruke denne kommandoen for å gjøre begge operasjoner:

```
bunzip2 <linux-2.4.9.tar.bz2 | tar xf -
```

Etter at utpakkingen er ferdig vil du ha en ny katalog, `/usr/src/linux`. Denne inneholder hele kildekoden til Linux-kjernen, det såkalte kjernetreet. På dette stadiet pleier jeg å gi denne katalogen et nytt, mer meningsfylt navn.

```
mv linux linux-2.4.9
```

Etter at navneendringen er gjort flytter du deg til den nye katalogen:

```
cd /usr/src/linux-2.4.9
```

Vanligvis endrer jeg `Makefile` før jeg starter med tilpasningen av den nye kjernen. Nær toppen av fila er det en variabel som heter **EXTRAVERSION**. Jeg gir denne verdien **ltsp-1** slik at kjerneversjonen blir **2.4.9-ltsp-1**. Slik versjonsangivelse gjør det enklere å holde orden på de ulike kjernene senere. Toppen av `Makefile` skal se slik ut når endringene er gjort:

```
VERSION = 2
PATCHLEVEL = 4
SUBLEVEL = 9
EXTRAVERSION = -ltsp-1

KERNELRELEASE=$(VERSION) . $(PATCHLEVEL) . $(SUBLEVEL) $(EXTRAVERSION)
```

7.2.2. Kjernepatcher

Etter at kjernetreet er pakket ut vil du kanskje legge inn patcher, f.eks. NFS Swap patch og Linux Progress Patch. Disse patchene må legges inn FØR du gjør tilpasningen av kjernen. En patch er kildekode som inneholder ny eller tilleggs-funksjonalitet.

7.2.2.1. NFS Swap patch

Denne patchen legger til muligheten for en arbeidsstasjon å benytte en swapfil som fysisk befinner seg på en annen maskin med en aktiv NFS-tjener. Det er best å ha så mye minne i en arbeidsstasjon at den ikke trenger å benytte seg av swap-mekanismen. Men dette er ikke alltid mulig, særlig i gamle maskiner. Da kan muligheten for å benytte swap over NFS gjøre det mulig å ta i bruk en ellers ubrukelig maskin.

Hvis den aktuelle katalogen er `/usr/src/linux-2.4.9` og patchen ligger i `/usr/src` kan du gjøre følgende for å sjekke patchen:

```
patch -p1 --dry-run <../linux-2.4.9-nfs-swap.diff
```


Dette vil sjekke om det ville gått greit å legge patchen inn i kjernetreet. Hvis kommandoen ikke ga noen feilmeldinger kan du kjøre den uten parameteren **--dry-run** option.

```
patch -p1 <../linux-2.4.9-nfs-swap.diff
```

7.2.2.2. Linux Progress Patch (LPP)

Linux Progress Patch lar deg vise et grafisk skjermbilde under oppstart-prosessen. De tekstlinjene som vanligvis vises på skjermen under oppstart blir sendt til en annen virtuell skjerm, og den grafiske skjermen vises i stedet. Spesielle instruksjoner gjør at man kan vise fram en framdriftsstolpe for å angi hvor langt oppstartsprosessen er kommet.

Du kan sjekke LPP på samme måte som med NFS Swap med denne kommandoen:

```
patch -p1 --dry-run <../lpp-2.4.9
```

Hvis du ikke får noen feilmelding kan du legge inn patchen med:

```
patch -p1 <../lpp-2.4.9
```

7.2.3. Tilpasning av kjernen

Du kan nå bruke et av de spesielle programmene beregnet på tilpasning av Linux-kjernen. Disse gir ulik form for interaktivitet og kan startes med en av de følgende kommandoer:

- make xconfig

Dette vil starte X Window-versjonen av tilpasningsprogrammet for Linux-kjernen

- make menuconfig

Dette starter en tekstbasert versjon av tilpasningsprogrammet for Linux-kjernen.

- make config

Dette starter en enkel linje-for-linje-versjon av tilpasningsprogrammet for Linux-kjernen.

7.2.3.1. Kjernetilpasninger ved bruk av initrd

Tilpasning av kjernen for bruk av initrd forutsetter at følgende parametre er satt:

- File systems → /dev filesystem support

Filsystemet /dev må være tilgjengelig. Dette styres i avsnittet 'File systems'. IKKE kryss av for 'Automatically mount at boot'. Monteringen foretas av skriptet **/linuxrc**

- Block devices → RAM disk support

LTSP arbeidsstasjoner forutsetter at kjernen har støtte for en RAM-disk. Dette styres i avsnittet 'Block devices'.

- Block devices → Initial RAM disk (initrd) support

Dette valget må også krysses av.

- Processor type and features → Processor family

Du må passe på at kjernen du lager vil kunne brukes på prosessoren i arbeidsstasjonen. Dette styrer du i avsnittet 'Processor type and features'. Du bør også skru av støtte for SMP (symmetrisk multiprosessering) med mindre du faktisk har flere prosessorer i arbeidsstasjonen.

- File systems → Network file systems → NFS Client support

Arbeidsstasjonen vil montere sitt rot filsystem via NFS, så det er nødvendig med støtte for NFS i arbeidsstasjonens kjerne.

Det skulle være alle de nødvendige valgene. I tillegg kan du skru av mange andre funksjoner for å redusere størrelsen på kjernen.

7.2.3.2. Tilpasning av kjerne som ikke skal bruke initrd

Tilpasning av en kjerne for bruk uten initrd er litt annerledes.

- Block devices → RAM disk support

LTSP arbeidsstasjoner trenger støtte for RAM-disk.

- Block devices → Initial RAM disk (initrd) support

Denne må skrues av.

- Networking options → IP:kernel level autoconfiguration

Denne må være tilgjengelig, så kryss av her. Kjernen vil automatisk sette opp Ethernet nettverks-enheten eth0, basert på parametre gitt på kjernens kommandolinje.

Det er ikke nødvendig å oppgi parametre for DHCP, BOOTP eller RARP siden Etherboot allerede har gjort en DHCP- eller BOOTP-forespørsel og har den nødvendige informasjonen. IP-parametrene legges inn som nødvendig på kjernens kommandolinje og sparer kjernen for jobben med å spørre selv.

- Network device support → Ethernet (10 or 100Mbit)

Når ikke initrd benyttes må det velges en driver som passer til nettverkskortet. Driveren MÅ være bygget inn i kjernen siden nettverkskortet må virke før rot filsystemet kan monteres og kjernemoduler er tilgjengelig. Dette er en viktig del av hvordan en kjerne med og uten en initrd virker.

- File systems → /dev filesystem support

Fra og med LTSP versjon 2.09pre2 trengs støtte for **devfs** uansett om initrd benyttes eller ikke.

- File systems → Automatically mount at boot

Dersom `initrd` IKKE benyttes må `/dev`-filsystemet monteres av kjernen som del av oppstartsprosessen. Så kryss av for 'Y' her.

- File systems → Network file systems → NFS Client support

Siden arbeidsstasjonen skal montere rot filsystemet sitt ved hjelp av NFS er det nødvendig med støtte for NFS-klienter.

7.2.4. Bygging av kjernen

For å gjøre ting enklere er en kopi av `.config` inkludert i `ltsp_initrd_kit`-pakken. Denne kopien kan legges inn i katalogen `/usr/src/linux-2.4.9` og brukes som utgangspunkt for ditt eget kjerne-oppsett.

Når du er ferdig med å velge (og velge vekk) kjerne-opsjoner må kjernen bygges. Dette utføres med disse kommandoene:

```
make dep
make clean
make bzImage
make modules
make modules_install
```

Du kan henge dem etter hverandre slik:

```
make dep && make clean && make bzImage && make modules && make modules_install
```

Den doble ampersand'en (&) angir at kommandoen foran må fullføres uten feil for at den neste skal kunne utføres. Hvis den andre utføres uten feil også, vil den tredje startes, osv.

Når kjernen er ferdig bygget vil den kunne finnes som filen
`/usr/src/linux-2.4.9/arch/i386/boot/bzImage`.

7.2.5. Merking av kjernen for Etherboot

For at Etherboot skal kunne håndtere en Linux-kjerne må denne behandles spesielt. Denne prosessen kalles å "merke" kjernen, og vil legge inn ekstra kode som utføres før kjernen overtar kontrollen over maskinen ved oppstart. Verktøyet som benyttes for denne merkingen kalles '**mknbi-linux**'.

`ltsp_initrd_kit` inneholder skall-scriptet **buildk**, og dette har alle de nødvendige besvergelesene som trengs for å forberede kjernen til oppstart over nettverket.

Kapittel 8. Innhold i lts.conf

Da LTSP ble laget var ett av kriteriene at det skulle være mulig å håndtere varierende typer maskinvare for arbeidsstasjonene. Den kombinasjonen av prosessor, nettverks- og skjermkort som finnes i de arbeidsstasjonene du installerer i dag vil ikke være tilgjengelig om tre måneder når neste bolk med tynne klienter skal installeres.

Derfor har vi sørget for at det skal være mulig å angi et oppsett for hver enkelt arbeidsstasjon. Oppsett-filen er `lts.conf` og den finnes i katalogen `/opt/ltsp/i386/etc`.

Formatet i `lts.conf` tillater en blokk med standard parametre og andre blokker med individuelle oppsett for hver arbeidsstasjon. Hvis alle maskinene du benytter som tynne klienter er like kan du greie deg med en enkelt '[Default]' blokk som angir alle nødvendige parametre.

8.1. Eksempel på lts.conf

Dette er et eksempel på hvordan `lts.conf` kan se ut:

```
[Default]
SERVER           = 192.168.0.254
X_MOUSE_PROTOCOL = "PS/2"
X_MOUSE_DEVICE   = "/dev/psaux"
X_MOUSE_RESOLUTION = 400
X_MOUSE_BUTTONS = 3
USE_XFS          = N
RUNLEVEL         = 5

[ws001]
XSERVER          = auto
X_MOUSE_PROTOCOL = "Microsoft"
X_MOUSE_DEVICE   = "/dev/ttyS1"
X_MOUSE_RESOLUTION = 50
X_MOUSE_BUTTONS = 3
X_MOUSE_BAUD     = 1200

[ws002]
XSERVER          = XF86_Mach64

[ws003]
RUNLEVEL         = 3
```

8.2. Tilgjengelige parametre i lts.conf

8.2.1. Generelle parametre

Kommentarer

Kommentarer starter med en skigard '#' og fortsetter til slutten av linjen.

LTSP_BASEDIR

Angir hvor LTSP sitt rot filsystem kan finnes. Standard verdi er `/opt/ltsp`

SERVER

Angir hvilken maskin som benyttes i parametrene XDM_SERVER, TELNET_HOST, XFS_SERVER og SYSLOG_HOST, hvis disse ikke angis eksplisitt. Hvis installasjonen har en maskin som benyttes for alle funksjonene kan du legge inn denne her og hoppe over de andre server- parametrene. Standard verdi er **192.168.0.254**.

SYSLOG_HOST

Hvis du ønsker å sende logger til en annen maskin enn standard-maskinen kan denne andre maskinen angis her. Hvis denne parameteren ikke er angitt vil verdien i 'SERVER' benyttes.

NFS_SERVER

Angir IP-adressen til NFS-tjeneren som benyttes når katalogen /home skal monteres. Standard verdi er innholdet i **SERVER**.

USE_NFS_SWAP

Angi **Y** hvis det ønskelig å benytte swap over NFS. Standard verdi er **N**.

SWAPFILE_SIZE

Med denne parameteren kontrolleres størrelsen på swap-filen. Standard verdi er **64MB**.

SWAP_SERVER

Swap-filen kan ligge på hvilken som helst maskin på nettverket som kan håndtere den. Angi IP-adressen til denne maskinen. Standard verdi er innholdet av **NFS_SERVER**.

NFS_SWAPDIR

Katalogen på tjeneren som eksporteres med NFS. Standard verdi er /var/opt/ltsp/swapfiles. Sørg for at denne katalogen er satt opp i /etc/exports.

TELNET_HOST

Hvis arbeidsstasjonen skal benytte et tegnbasert grensesnitt vil innholdet av denne parameteren benyttes for å finne maskinen som skal ta i mot TELNET-påloggingen. Standard-verdi er innholdet i **SERVER**.

DNS_SERVER

Brukes for å bygge /etc/resolv.conf.

SEARCH_DOMAIN

Brukes for å bygge /etc/resolv.conf.

MODULE_01 til og med MODULE_10

Opp til ti kjernemoduler kan legges inn ved å benytte disse parametrene. Angi som verdi hele kommandolinjen du ville brukt sammen med **insmod**. Eksempel:

```
MODULE_01 = uart401.o
MODULE_02 = sb.o io=0x220 irq=5 dma=1
MODULE_03 = opl3.o
```

Hvis verdien av denne parameteren er et absolutt filnavn vil kommandoen **insmod** bli brukt til å installere modulen. Ellers vil **modprobe** bli benyttet.

RAMDISK_SIZE

Når arbeidsstasjonen starter opp lager den en ramdisk og monterer den i katalogen /tmp. Du kan

kontrollere størrelsen på dette filsystemet med denne parameteren. Angi verdien i Kbytes (1024 bytes). For en ramdisk på 1 megabyte bruk **RAMDISK_SIZE = 1024**

Hvis du endrer størrelsen på ramdisken her må du også endre størrelsen på ramdisken i kjernen. Dette kan bygges inn, eller, hvis du bruker Etherboot eller Netboot, angis når du merker kjernen med mknbi–linux.

Standard verdi er 1024 (1 MB).

RCFILE_01 thru RCFILE_10

Ytterligere RC–script kan bli utført av scriptet rc.local. Legg det scriptet du ønsker utført i katalogen /etc/rc.d og angi navnet i en av disse parametrene her.

SOUND

Hvis lydpakken til LTSP er installert må du angi **Y** her. Da vil scriptet **rc.sound** bli utført og lydkortet og annen programvare nødvendig for å få lyd til å virke bli satt opp. Standard verdi er **N**.

8.2.2. X–Window parametre

XDM_SERVER

Hvis du ønsker å benytte en annen maskin enn standard–tjeneren til å vise fram XDM innloggings–bildet kan du angi denne her. Standard verdi er innholdet i 'SERVER'.

XSERVER

Angir hvilken X–tjener arbeidsstasjonen skal benytte. For PCI og AGP skjermkort er denne parameteren ikke nødvendig, siden scriptet rc.local søker etter slike kort og tilordner den riktige driveren. For å vise at denne funksjonen skal benyttes kan det være greit å sette verdien til **auto**.

For ISA skjermkort, eller for å tvinge inn en spesiell driver, kan du angi en verdi her. Bruk navnet på driveren.

Hvis verdien begynner med **XF86_** vil XFree86 3.3.6 benyttes. I motsatt fall brukes XFree86 4.1.x. Standard verdi er **auto**.

X_MODE_0 til X_MODE_2

Opp til tre Modelines eller oppløsninger kan settes opp for arbeidsstasjonen. Denne parameteren kan ta to ulike typer verdier. Enten angis en oppløsning eller en komplett Modeline.

```
X_MODE_0 = 800x600
    eller
X_MODE_0 = 800x600 60.75 800 864 928 1088 600 616 621 657 -HSync -VSync
```

Hvis ingen verdi er angitt for X_MODE_? vil de innebygde Modelines bli benyttet sammen med oppløsningene 1024x768, 800x600 og 640x480.

Hvis det er angitt verdier for en eller flere av X_MODE_? vil disse verdiene overstyre de innebygde standard–verdiene

X_MOUSE_PROTOCOL

Verdier som stemmer med XFree86 Pointer Protocol kan benyttes her. Typiske verdier er "PS/2" og "Microsoft". Standard verdi er "**PS/2**".

X_MOUSE_DEVICE

Dette er porten musen er koblet til. Hvis det er en seriell mus må dette reflektere en seriellport , f. eks. **/dev/ttyS0** eller **/dev/ttyS1**. Hvis det er en PS/2 type mus vil verdien typisk være **/dev/psaux**. Standard verdi er **/dev/psaux**.

X_MOUSE_RESOLUTION

Her skal det angis en verdi som benyttes for 'Resolution' for musen i **XF86Config**-filen. En typisk verdi kan være **50** for en seriell-mus og **400** for en PS/2-mus. Standard verdi er **400**.

X_BUTTONS

Verdien her forteller systemet hvor mange knapper musen har. Vanligvis er dette **2** eller **3**. Standard verdi er **3**.

X_MOUSE_EMULATE3BTN

Benytt denne parameteren for å angi om X-serveren skal behandle en mus med to knapper som om den hadde tre. Dette gjør den ved å se et samtidig klikk på begge knappene som et klikk på den midtre knappen. Standard verdi er **N**.

X_MOUSE_BAUD

Dette definerer baud-raten for serielle mus. Standard verdi er **1200**.

X_COLOR_DEPTH

Antallet bits som benyttes for fargedybden. Verdier her skal være **8**, **15**, **16**, **24** eller **32**. 8 bits gir 256 farger, 16 gir 65 536 farger, 24 gir 16 millioner og 32 bits gir 4,2 milliarder farger. Standard verdi er **16**.

USE_XFS

Du kan velge å benytte en X Font Server (XFS) eller lese fontene fra filsystemet montert via NFS. XFS er en enkel måte å ha fontene samlet på ett sted, men det har vært problemer med å håndtere mer enn 40 arbeidsstasjoner. Bruk enten **Y** eller **N** som verdier. Standard verdi er **N**. Hvis du ønsker å kjøre en font-tjener kan du angi hvilken maskin denne kjører på ved å benytte parameteren **XFS_SERVER**.

XFS_SERVER

Hvis du har en XFS-tjener som leverer fonter kan du angi hvilken maskin dette er ved å legge inn IP-adresse eller maskinnavn i denne parameteren. Hvis det ikke angis noen verdi her vil verdien angitt for **SERVER** benyttes.

X_HORZSYNC

Angir **HorizSync**-parameteren som benyttes av XFree86. Standard verdi er "**31-62**".

X_VERTREFRESH

Angir **VertSync**-parameteren som benyttes av XFree86. Standard verdi er "**55-90**".

XF86CONFIG_FILE

Hvis du ønsker å bygge opp en komplett XF86Config-fil selv, kan du legge denne i katalogen **/opt/ltsp/i386/etc**. Navnet på filen oppgis som verdi her. F.eks.

```
XF86CONFIG_FILE = XF86Config.ws004
```

8.2.3. Parametre for berørings-skjermer

USE_TOUCH

Hvis du kobler en berørings-skjerm til arbeidsstasjonen kan du benytte verdien **Y**. I så fall vil de øvrige variablene under kunne ta hånd om spesifikke sider ved oppsettet av slike enheter. Standard verdi er **N**.

X_TOUCH_DEVICE

En berørings-skjerm virker som en mus og snakker vanligvis med arbeidsstasjonen gjennom en seriell-port. Du kan angi hvilken seriellport dette er med denne verdien, f.eks. **/dev/ttyS0**. Det er ingen standard verdi her.

X_TOUCH_MINX

Kalibrering av en EloTouch-skjerm. Standard verdi er **433**.

X_TOUCH_MAXX

Kalibrering av en EloTouch-skjerm. Standard verdi er **3588**.

X_TOUCH_MINY

Kalibrering av en EloTouch-skjerm. Standard verdi er **569**.

X_TOUCH_MAXY

Kalibrering av en EloTouch-skjerm. Standard verdi er **3526**.

X_TOUCH_UNDELAY

Kalibrering av en EloTouch-skjerm. Standard verdi er **10**.

X_TOUCH_RPTDELAY

Kalibrering av en EloTouch-skjerm. Standard verdi er **10**.

8.2.4. Parametre for lokale programmer

LOCAL_APPS

Hvis du vil kunne kjøre programmer lokalt på arbeidsstasjonen setter du denne variabelen til **Y**. Det er flere andre ting som må gjøres på sentralmaskinen for å få lokale programmer til å fungere. Se i avnsittet om 'Lokale programmer' i LTSP-manualen for å finne mer informasjon om hvordan dette gjøres. Standard verdi er **N**.

NIS_DOMAIN

Hvis du velger å bruke lokale programmer må du ha en NIS-tjener på nettverket. Variabelen **NIS_DOMAIN** brukes til å spesifisere NIS domene-navnet. Dette må stemme overens med det domene-navnet som er brukt på NIS-tjeneren. Dette er **IKKE** det samme som et Internett-domene. Standard verdi er **ltsp**.

NIS_SERVER

Angi IP-adressen til NIS-tjeneren hvis du ikke vil benytte kringkasting for å finne den.

8.2.5. Parametre for tastatur

Alle støttefilene for tastatur er nå tatt med i /opt/ltsp/i386–hierarkiet. Å sette opp internasjonale tastatur er redusert til å sette opp XFree86. Flere parametre er tilgjengelige for å gjøre dette mulig.

Verdiene for parameterene er de samme som finnes i dokumentasjonen for XFree86. Gyldige verdier for XFree86 er også gyldige for LTSP.

Vi vil gjerne ta med i dokumentasjonen hvilke verdier som er gyldige for ulike typer internasjonale tastatur. Dersom du har informasjon om dette ber vi deg gi denne videre til utviklergruppen for LTSP. Alle bidrag tas i mot med takk.

XkbTypes

Standard verdi er ordet '**default**'.

XkbCompat

Standard verdi er ordet '**default**'.

XkbSymbols

Standard verdi er '**us(pc101)**'.

XkbModel

Standard verdi er '**pc101**'.

XkbLayout

Standard verdi er '**us**'. Bruk '**no**' for norske tastatur.

8.2.6. Parametere for bruk av skrivere

Opp til tre skrivere kan kobles til hver arbeidsstasjon. En kombinasjon av skrivere med parallell– og seriell tilkobling kan settes opp ved å benytte de følgende parameterene i oppsettfilen **lts.conf**.

PRINTER_0_DEVICE

Enhetsnavnet for den første skriveren. Verdier som **/dev/lp0**, **/dev/ttyS0** eller **/dev/ttyS1** kan benyttes.

PRINTER_0_TYPE

Typen skriver. Tillatte verdier er '**P**' for parallell og '**S**' for seriell.

PRINTER_0_PORT

TCP/IP–porten som skal benyttes på arbeidsstasjonen. Standard verdi er '**9100**'.

PRINTER_0_SPEED

Hvis skriveren har seriell–tilkobling kan baud–raten med denne verdien. Standard verdi er '**9600**'.

PRINTER_0_FLOWCTRL

Flyt–kontrollen for serielle skrivere kan settes her. Benytt '**S**' for kontroll via programvare (XON/XOFF) eller '**H**' for kontroll via maskin–vare. Hvis ingenting er spesifisert vil '**S**' bli benyttet.

PRINTER_0_PARITY

Pariteten kan defineres for serielle skrivere med denne variabelen. Valgene er 'E' (like), 'O' (odde) eller 'N' (ingen). Dersom ingenting er spesifisert vil 'N' bli benyttet.

PRINTER_0_DATABITS

Antallet databits kan angis. Valgene er '5', '6', '7' og '8'. '8' blir brukt som verdi om ingenting er spesifisert.

PRINTER_1_DEVICE

Enhetsnavn for skriver nr. 2.

PRINTER_1_TYPE

Enhetstype for skriver nr. 2.

PRINTER_1_PORT

TCP/IP-port for skriver nr. 2.

PRINTER_1_SPEED

Baudrate for for skriver nr. 2. (seriell)

PRINTER_1_FLOWCTRL

Flyt-kontroll for skriver nr. 2 (seriell)

PRINTER_1_PARITY

Paritet for skriver nr. 2 (seriell)

PRINTER_1_DATABITS

Data-bits for skriver nr. 2 (seriell)

PRINTER_2_DEVICE

Enhetsnavn for skriver nr. 3

PRINTER_2_TYPE

Enhetstype for skriver nr. 3.

PRINTER_2_PORT

TCP/IP-port for skriver nr. 3.

PRINTER_2_SPEED

Baudrate for for skriver nr. 3. (seriell)

PRINTER_2_FLOWCTRL

Flyt-kontroll for skriver nr. 3 (seriell)

PRINTER_2_PARITY

Paritet for skriver nr. 3 (seriell)

PRINTER_2_DATABITS

Data-bits for skriver nr. 3 (seriell)

Kapittel 9. Lokale programmer

LTSP gir deg muligheten til å velge mellom å kjøre programmer lokalt på arbeidsstasjonen eller sentralt på tjeneren.

Den klart enkleste løsningen er å kjøre LTSP med alle programmer på tjeneren. Alle programmene benytter tjenerens prosessor og minne, men viser skjermbildene sine på arbeidsstasjonens skjerm og benytter dens tastatur og mus.

Dette er en grunnleggende funksjon i X Window. Arbeidsstasjonen fungerer akkurat som en standard X Window terminal.

For at en bruker skal kunne kjøre et program på arbeidsstasjonen trenger denne å vite forskjellige ting om brukeren, f. eks.:

- Brukerens id
- Hovedgruppen brukeren hører til
- Brukerens hjemmekatalog

LTSP benytter seg av NIS – Network Information Service for å gjøre bruker- og gruppeinformasjon tilgjengelig for arbeids- stasjonene.

9.1. Fordeler med å kjøre programmer lokalt

Det er fordeler med å kjøre programmer lokalt på arbeidsstasjonen.

- Reduserer belastningen på tjeneren. I store nettverk med minne-intensive programmer som Netscape, kan det gi bedre ytelse å kjøre slike programmer lokalt på arbeidsstasjonen. Det forutsetter at arbeidsstasjonen er kraftig nok til å håndtere det.
- Løpske programmer vil ikke påvirke andre brukere.
- Støtte for lyd er mye enklere å sette opp når programmet som spiller lyden kjører på arbeidsstasjonen.

9.2. Ting å passe på når man setter opp lokale programmer

Det er mer utfordrende å sette opp programmer som skal kjøres lokalt.

- Større krav til arbeidsstasjonen. Den trenger mer minne og en kraftigere prosessor. 64 MB RAM er et godt utgangspunkt.
 - NIS – for å kunne kjøre programmer på arbeidsstasjonen må man kunne identifisere seg, m.a.o. må arbeidsstasjonen vite hvem brukeren er. Dette krever en eller annen form for passordautentisering. NIS er valgt som metode for denne autentiseringen over nettverket.
 - Nye kataloger må eksporteres fra arbeidsstasjonen for å kunne monteres via NFS.
 - Tregere oppstart av programmer fordi de må hentes via NFS og forårsaker økt nettverksaktivitet. Siden hver enkelt kopi av programmet kjører på en egen prosessor vil de ikke kunne trekke på Linux' innebygde evne til å la de ulike kopiene dele programkode mellom seg. Slik deling fører til reduksjon i forbruk av minne og gjør at det går raskere å starte opp nye instanser av programmet.
-

9.3. Oppsett av tjenermaskin for lokale programmer

9.3.1. Parametre i `lts.conf`

Noen parametre må settes i filen `lts.conf`:

LOCAL_APPS

Denne må settes til **Y**. Det fører til at det følgende skjer under oppstart av arbeidsstasjonen:

1. Katalogen `/home` på tjeneren vil monteres via NFS.
2. `/var/yp/nicknames` opprettes på arbeidsstasjonen.
3. **portmapper** startes på arbeidsstasjonen.
4. **xinetd** startes på arbeidsstasjonen.
5. `/etc/yp.conf` opprettes på arbeidsstasjonen.
6. Kommandoen **domainname** kjøres med verdien av **NIS_DOMAIN** fra `lts.conf` som parameter.
7. Kommandoen **ypbind** kjøres på arbeidsstasjonen.

NIS_DOMAIN

For at NIS skal fungere må alle nodene i nettverket som skal være assosiert med en spesifikk NIS-tjener være medlem av samme NIS-domene. Slike domener må ikke forveksles med DNS-domener. Du bruker **NIS_DOMAIN** for angi NIS-domenet arbeidsstasjonen hører til.

NIS_SERVER

NIS vil enten prøve å koble seg til en spesifikk NIS tjener eller kringkaste en forespørsel på nettverket for å finne en slik tjener. Hvis du vil angi en spesifikk tjener oppgir du ip-adressen som verdi i **NIS_SERVER**.

9.3.2. Network Information Service – NIS

NIS er en klient-tjener type tjeneste. På tjeneren går det en daemon som aksepterer forespørsler fra klienter (arbeidsstasjoner). Denne daemonen kalles **ypserv**.

På arbeidsstasjonen går det en prosess som kalles **ypbind**. Når arbeidsstasjonen trenger informasjon om en bruker vil den bruke **ypbind** for å etablere en forbindelse til **ypserv** på tjeneren.

Hvis NIS allerede er i drift i det nettverket du skal innlemme arbeidsstasjonene i, trenger du ikke å sette opp **ypserv** også på LTSP-tjeneren. Det holder å sette inn verdier i **NIS_DOMAIN** og **NIS_SERVER** som stemmer med det som allerede finnes.

Hvis NIS ikke er i bruk i nettverket må du sette opp tjeneren til å kjøre tjenesten **ypserv**.

For å lære om hvordan man setter opp en NIS-tjener kan du lese *The Linux NIS(YP)/NYS/NIS+ HOWTO*. Denne finner du på <http://www.linuxdoc.org/>. Se listen over referanser i slutten av denne håndboka.

9.4. Oppsett av programmer

For å få et program til å kjøre på arbeidsstasjonen må du legge alle delene på et sted de kan finnes. av arbeidsstasjonen.

Med eldre versjoner av LTSP (2.08 og tidligere) ble mange kataloger eksportert fra tjeneren og montert av arbeidsstasjonen. Kataloger som `/bin`, `/usr/bin`, `/lib` og `/usr` kunne ses av arbeidsstasjonen.

Problemet med denne løsningen er at arbeidsstasjon og tjener må ha samme arkitektur for å kunne virke. Faktisk kan så små forskjeller som mellom Pentium 2 (i686) og klassisk Pentium (i586) skape problemer. Dette skyldes at det vil være forskjellige biblioteker i bruk på de to arkitekturene.

Derfor er den reneste måten å håndtere dette å ha et komplett sett med kataloger med alle binærer og biblioteker arbeidsstasjonen trenger. Disse er da helt uavhengige av tjenerens binærer og biblioteker.

Oppsett av et program for å kjøre lokalt består i å sørge for at alle de riktige bitene legges på riktig sted i katalogstrukturen. En av pakkene tilgjengelig på LTSPs nettsted er lokal Netscape og denne installerer en rekke filer i katalogen `/opt/ltsp/i386/usr/local/netscape`. Ting som java-klasser, hjelpefiler, eksekverbare binærfiler og skript havner der.

Netscape trenger ingen ytterligere system-biblioteker, slik at det ikke er nødvendig å plassere slike i katalogen `/opt/ltsp/i386/lib`. Men mange andre programmer trenger slike tilleggsbiblioteker.

Hvordan avgjør du hvilke biblioteker som er nødvendig? Her kommer kommandoen **ldd** til nytte.

La oss anta at du ønsker å få et gitt program til å kjøre lokalt. Vi bruker **gaim** som eksempel. **gaim** er en klient for forskjellige chat-protokoller som brukes til å kommunisere med andre på nettsteder beregnet for dette.

Det første du må gjøre er å finne ut hvor binærfilen **gaim** er plassert. På Red Hat Linux 7.2 er dette katalogen `/usr/bin`.

Når binæren **gaim** er funnet kan du kjøre **ldd** mot den:

```
[jam@server ~]$ ldd /usr/bin/gaim
        libaudiofile.so.0 => /usr/lib/libaudiofile.so.0 (0x40033000)
        libm.so.6         => /lib/i686/libm.so.6 (0x40051000)
        libnsl.so.1       => /lib/libnsl.so.1 (0x40074000)
        libgnomeui.so.32  => /usr/lib/libgnomeui.so.32 (0x4008a000)
        libart_lgpl.so.2  => /usr/lib/libart_lgpl.so.2 (0x4015d000)
        libgdk_imlib.so.1 => /usr/lib/libgdk_imlib.so.1 (0x4016c000)
        libSM.so.6        => /usr/X11R6/lib/libSM.so.6 (0x40191000)
        libICE.so.6       => /usr/X11R6/lib/libICE.so.6 (0x4019a000)
        libgtk-1.2.so.0   => /usr/lib/libgtk-1.2.so.0 (0x401b1000)
        libdl.so.2        => /lib/libdl.so.2 (0x402df000)
        libgdk-1.2.so.0   => /usr/lib/libgdk-1.2.so.0 (0x402e3000)
        libgmodule-1.2.so.0 => /usr/lib/libgmodule-1.2.so.0 (0x40319000)
        libXi.so.6        => /usr/X11R6/lib/libXi.so.6 (0x4031d000)
        libXext.so.6      => /usr/X11R6/lib/libXext.so.6 (0x40325000)
        libX11.so.6       => /usr/X11R6/lib/libX11.so.6 (0x40333000)
        libgnome.so.32    => /usr/lib/libgnome.so.32 (0x40411000)
        libgnomesupport.so.0 => /usr/lib/libgnomesupport.so.0 (0x40429000)
        libesd.so.0       => /usr/lib/libesd.so.0 (0x4042e000)
        libdb.so.2        => /usr/lib/libdb.so.2 (0x40436000)
        libglib-1.2.so.0  => /usr/lib/libglib-1.2.so.0 (0x40444000)
        libcrypt.so.1     => /lib/libcrypt.so.1 (0x40468000)
        libc.so.6         => /lib/i686/libc.so.6 (0x40495000)
        libz.so.1         => /usr/lib/libz.so.1 (0x405d1000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Eksempelet over viser alle bibliotekene programmet **gaim** er dynamisk linket mot.

De fleste programmer som bruker delte biblioteker er avhengige av den dynamiske bibliotek-lasteren **ld-linux** for å finne og laste de enkelte delte bibliotekene. Noen programmer vil laste disse bibliotekene manuelt med funksjonskallet **dlopen()**. For slike programmer vil **ldd** ikke vise hvilke biblioteker som trengs. I slike tilfeller kan kommandoen **strace** benyttes til å kjøre programmet. I utskriften fra kjøringen vil du kunne finne referanser til kall av **dlopen()** der det aktuelle biblioteket er angitt som parameter til kallet.

Når listen over nødvendige biblioteker er klar må disse plasseres på riktig sted i katalogsamlingen til LTSP, `/opt/ltsp/i386`.

9.5. Start av lokale programmer

I X Window vil programmer typisk kjøre relativt til hvor vindus-håndtereren kjører. Med andre ord, hvis vindushåndtereren går på tjeneren og viser resultatene på arbeidsstasjonen – vil programmer som startes også kjøre på tjeneren og vise sine skjermbilder på arbeidsstasjonen.

I slike tilfeller må man be arbeidsstasjonen selv å kjøre programmet hvis det skal gå lokalt. Dette kan løses ved å benytte kommandoen **rsh**.

Dette er et eksempel på hvordan man kan få programmet **gaim** til å kjøres lokalt på arbeidsstasjonen:

```
HOST=`echo $DISPLAY | awk -F: '{ print $1 }'`
rsh ${HOST} /usr/bin/gaim -display ${DISPLAY}
```

Eksempelet over kan skrives inn i et **xterm** terminalvindu eller legges inn i et skallskript og utføres ved hjelp av et ikon på arbeidsflaten.

Oppstart av Netscape lokalt kan gjøres på samme måten, men i tillegg må en miljøvariabel settes før programmet startes.

```
HOST=`echo $DISPLAY | awk -F: '{ print $1 }'`
rsh ${HOST} MOZILLA_HOME=/usr/local/netscape \
  /usr/local/netscape/netscape -display ${DISPLAY}
```

Kapittel 10. Eksempler på oppsett

Nesten alle driftsparametre for arbeidsstasjonen kan settes med variable i filen `lts.conf`, som vanligvis er å finne i katalogen `/opt/ltsp/i386/etc`.

10.1. Standard 2–knappers seriell mus

Dette er et eksempel på hvordan man angir parametre for en seriell mus.

```
X_MOUSE_PROTOCOL = "Microsoft"
X_MOUSE_RESOLUTION = 400
X_MOUSE_DEVICE = "/dev/ttyS0"
X_MOUSE_BUTTONS = 2
X_MOUSE_EMULATE3BTN = Y
```

10.2. PS/2 mus med rulle

Dette er et eksempel på et avsnitt i `lts.conf` for en Intellimouse:

```
X_MOUSE_PROTOCOL = "IMPS/2"
X_MOUSE_DEVICE = "/dev/psaux"
X_MOUSE_RESOLUTION = 400
X_MOUSE_BUTTONS = 5
X_ZAxisMapping = "4 5"
```

10.3. USB–skriver på en ThinkNic

ThinkNic arbeidsstasjoner har en USB–port som kan benyttes til å koble til en lokal skriver. Dette er et eksempel på verdier og parametre i `lts.conf` for å få dette til å virke:

```
MODULE_01 = usb-ohci
MODULE_02 = printer
PRINTER_0_DEVICE = /dev/usb/lp0
PRINTER_0_TYPE = S
```

10.4. Tvungen bruk av en eldre versjon av XFree86 3.3.6 Xserver

Hvis ikke annet angis blir XFree86 4.1.0 benyttet for å drive arbeidsstasjonen. Dersom man ønsker å benytte en eldre XFree86–versjon(3.3.6) må man først installere riktig pakke. Så må man eksplisitt angi i `lts.conf` at denne skal benyttes for den aktuelle arbeidsstasjonen. Dette er et eksempel der SVGA–Xserveren skal brukes:

```
XSERVER = XF86_SVGA
```

Kapittel 11. Andre kilder til informasjon

11.1. På Internett

1. LTSPs nettsted

www.LTSP.org

2. Diskless–Nodes HOWTO for Linux

www.linuxdoc.org/HOWTO/Diskless–HOWTO.html

3. Nettstedet til Etherboot–prosjektet

etherboot.sourceforge.net

4. The Rom–O–Matic site

www.Rom–O–Matic.net

5. XFree86–Video–Timings–HOWTO

www.linuxdoc.org/HOWTO/XFree86–Video–Timings–HOWTO.html

6. XFree86 Mouse Support

www.xfree86.org/current/mouse.html

7. The Linux NIS(YP)/NYS/NIS+ HOWTO

www.linuxdoc.org/HOWTO/NIS–HOWTO.html

11.2. Publikasjoner på papir

- 1.

Managing NFS and NIS
Hal Stern
O'Reilly & Associates, Inc.
1991
ISBN 0–937175–75–7

- 2.

TCP/IP Illustrated, Volume 1
W. Richard Stevens
Addison–Wesley
1994
ISBN 0–201–63346–9

X Window System Administrator's Guide

Linda Mui and Eric Pearce

O'Reilly & Associates, Inc.

1993

ISBN 0-937175-83-8

(Volume 8 of the The Definitive Guides to the X Window System)