LTSP − Linux Terminal Server Project − v3.0

James McQuillan

jam@LTSP.org

Revision History
Revision 1.0 2001−12−02 Revised by: jam

GNU/Linux makes a great platform for deploying diskless thin clients. The primary purpose of this document is to show you how to deploy thin clients using LTSP. But, this document also covers many issues with diskless workstations in general.

Polska wersja: Wojtek Borzestowski
Wojtek na Borzestowski dot eu dot org

Spis treści

Wstęp **4**

Rozdział 1 Teoria działania **6**

Rozdział 2 Instalowanie LTSP na serwerze **10**

2.1 Instalowanie pakietu RPM 10

2.1.1 Pakiet LTSP Core - pakiet rdzenia LTSP 10
2.1.2 Inicjalizacja serwera 10
2.1.3 Specyfikacja konfiguracji stacji roboczej 11
2.1.3.1 /etc/dhcpd.conf 11
2.1.3.2 /etc/hosts 12
2.1.3.3 /opt/ltsp/i386/lts.conf 13

Rozdział 3 Ustawienia stacji roboczej **15**

3.1 Tworzenie dyskietki rozruchowej 15

Rozdział 4 Uruchamianie stacji roboczej **17**

Rozdział 5 Drukowanie **18**

5.1 Ustawienia po stronie klienta 18

5.2 Ustawienia po stronie serwera 18

Rozdział 6 Rozwiązywanie problemów **21**

6.1 Wyszukiwanie błędów na dyskietce 21

6.2 Wyszukiwanie błędów związanych z DHCP 21

6.2.1 Czy połączenia są prawidłowe? 22
6.2.2 Czy proces dhcpd działa? 22
6.2.3 Sprawdzanie plików konfiguracyjnych dhcpd 23
6.2.4 Czy ipchains lub iptables nie blokują odpowiedzi? 23
6.2.4.1 Sprawdzanie ipchains 23
6.2.4.2 Sprawdzanie iptables 23
6.2.5 Czy stacja robocza wysyła zapytanie? 24

6.3 Wyszukiwanie błędów związanych z TFTP 24

6.3.1 tftpd nie został uruchomiony 24
6.3.2 Jądro nie jest tam gdzie spodziewa się go znaleźć tftpd 25

6.4 Wyszukiwanie błędów związanych z systemem plików NFS 25

6.4.1 Nie odnaleziono init 25
6.4.2 Serwer zgłasza błąd „error –13” 25
6.4.3 Problemy z demonem NFS (portmap, nfsd i mountd) 25
6.4.3.1 Portmapper (portmap) 25
6.4.3.2 Demony NFS i MOUNT (nfs i mountd) 26

6.5 Poszukiwanie błędów związanych z X serwerem 27

6.6 Wyszukiwanie błędów związanych z managerem okien 28

6.6.1 Szary ekran z dużym kursorem w kształcie X 28

Rozdział 7 Jądra **31**

7.1 Standardowe jądra LTSP 31

7.2 Kompilacja własnego jądra 31

7.2.1 Pozyskiwanie źródła jądra ... 31
7.2.2 Łaty jądra ... 32
7.2.2.1 Łata NFS Swap ... 33
7.2.2.2 Linux Progress Patch (LPP) ... 33
7.2.3 Konfigurowanie opcji jądra ... 33
7.2.3.1 Konfiguracja jądra z initrd ... 34
7.2.3.2 Konfiguracja jądra bez initrd ... 34
7.2.4 Kompilowanie jądra ... 35
Rozdział 8 Wpisy lts.conf ... 36
8.1 Przykładowy plik lts.conf ... 36
8.2 Parametry pliku lts.conf ... 36
8.2.1 Parametry główne ... 36
8.2.2 Ustawienia X –Window ... 38
8.2.3 Parametry ekranu dotykowego ... 40
8.2.4 Parametry lokalnych aplikacji ... 41
8.2.5 Parametry klawiatury ... 41
8.2.6 Konfiguracja parametrów drukarki ... 42
Rozdział 9 Aplikacje lokalne ... 44
9.1 Korzyści wypływające z lokalnego użytkowania apps ... 44
9.2 Co należy wziąć pod uwagę stosując lokalne aplikacje? ... 44
9.3 Konfiguracja serwera dla lokalnych aplikacji ... 45
9.3.1 Wpisy w lts.conf ... 45
9.3.2 Network Information Service - NIS ... 45
9.4 Konfigurowanie aplikacji ... 46
9.5 Uruchamianie lokalnych aplikacji ... 47
Rozdział 10 Przykłady konfiguracji ... 49
10.1 Mysz szeregowa ... 49
10.2 Mysz PS/2 z kółkiem ... 49
10.3 Drukarka USB przy ThinkNic ... 49
10.4 Konfiguracja X serwera z wykorzystaniem Xfree86 3.3.6 ... 49
Rozdział 11 Inne źródła informacji ... 50
11.1 W Internecie ... 50
11.2 Publikacje drukowane ... 50

Wstęp

Projekt LTSP dostarcza prostego sposobu na wykorzystanie niedrogich stacji roboczych jako terminali serwera GNU/ Linux - działających w trybie graficznym albo „znakowym" (bazującym na tekście).

W klasycznym biurze wszystkie biurka wyposażone są w komputery osobiste, każdy z własnym twardym dyskiem o względnie dużej mocy obliczeniowej. Użytkownicy gromadzą swoje dane na lokalnych twardych dyskach a kopie zapasowe są wykonywane rzadko - jeśli w ogóle.

Czy ma sens, wyposażenie każdego biurka w tak mocny komputer?

Nasza odpowiedź brzmi: nie!

Na szczęście istnieją inne możliwości. Jeśli korzysta się z LTSP, wystarczą Pecety z dolnej półki cenowej - twardy dysk, stacje dysków i CD-ROM okażą się zbędne. Konieczna jest jedynie rozruchowa (ładująca) karta sieciowa, która wystartuje komputer. Wiele kart sieciowych posiada puste gniazda pamięci rozruchowej ROM, które tylko czekają na włożenie w nie odpowiednio zaprogramowanej pamięci.

Podczas fazy ładowania, bezdyskowa stacja robocza otrzymuje swój adres IP i jądro z serwera, potem montuje podstawowy system plików z serwera przez NFS.

Stacja robocza może być skonfigurowana w jednym z trzech trybów:

- **Graficzny X Windows**

  Podczas używania Systemu X Windows stacja robocza może korzystać ze wszystkich aplikacji na serwerze lub na innych serwerach wewnątrz sieci.

- **Bazujący na tekście / znakach: Sesje Telnet**

  Stacja robocza może utworzyć do dziewięciu sesji TELNET na serwerze. Każda sesja odbywa się na oddzielnej wirtualnej konsoli. Za pomocą ALT-F1 do ALT-F9 można przełączać się pomiędzy poszczególnymi konsolami.

- **Zachęta powłoki**

  Stacja robocza może być tak skonfigurowana, by przechodziła bezpośrednio do powłoki bash na konsoli. Jest to korzystne w przypadku problemów z wystartowaniem systemu graficznego lub NFS.

Naprawdę świetne jest to, że za pomocą jednego serwera GNU/Linux, można obsługiwać cały szereg stacji roboczych. Ich ilość zależy od rodzaju operacji na stacjach roboczych i możliwości serwera.

Nie jest niczym niezwykłym obsługiwanie 40 stacji roboczych z przeglądarką internetową Mozilla i pakietem biurowym OpenOffice.org za pomocą jednego serwera z Dual PIII-650 oraz 1GB pamięci ram. Zostało to sprawdzone i na pewno działa.

1. Zastrzeżenia

Zarówno autor jak i dystrybutorzy, jak też inni, którzy przyczynili się do powstania tej pracy nie są w żaden sposób odpowiedzialni za psychiczne, finansowe, moralne lub innego typu szkody powstałe w wyniku postępowania wg sugestii zawartych w tym tekście.

Prawo autorskie i licencja.

Do powstania tej pracy w ogromnym stopniu przyczyniła się moja sympatyczna koleżanka Sylwia Begierska. Dzięki wielkie za pomoc :-)

Do powstania tej pracy w ogromnym stopniu przyczyniła się moja sympatyczna koleżanka Sylwia Begierska. Dzięki wielkie za pomoc :-)
Grzesiek, Tobie dzięki za końcowe poprawki. Jak będę we Wrocławiu to koniecznie musimy na piwo pójść :-))))))

Aha, zapewne dokument ten może w pewnych miejscach mało przypominać oryginał, gdyż tłumaczyłem to dla siebie i w naprawdę wolnych chwilach. Później dopiero chciałem z tego zrobić coś przynajmniej podobnego do oryginału. Dlatego niniejszą polską dokumentację używasz na własną odpowiedzialność i tak jak wyżej już napisano nie odpowiadam za Twoją chorobę psychiczną po zastosowaniu rad z niniejszej pracy :-)

Rozdział 1 Teoria działania

Uruchomienie bezdyskowej stacji roboczej obejmuje wiele etapów. Zrozumienie tego co dzieje się po kolei w dużym stopniu pomoże rozwiązywać problemy w miarę ich pojawiania się.

Przykład oparty jest na następującej konfiguracji:

- Standardowa x86 based workstation
- Karta sieciowa Linksys LNE100TX z bootrom'em
- Grafika na chipsecie Intel i810
- Serwer uruchomiony na dystrybucji RedHat 7.2
- DHCP
- Adresy IP hostów z puli 192.168.0.0/24

Zakładając, że serwer jest skonfigurowany pakietem LTSP, oto co się będzie działo na stacji roboczej:

1. Po włączeniu stacji roboczej, pojawi się "Power On Self Test" - POST
2. Podczas autotestu, bios będzie szukał dodatkowej pamięci stałej (expansion roms). Karta sieciowa zawiera pamięć rozruchową Etherboot, która jest rozszerzeniem ROM. Bios odnajdzie ROM na karcie sieciowej i rozpozna ją jako dodatkową pamięć stałą.
3. Kiedy POST jest zakończony, uruchomi się program zawarty w pamięci Etherboot.
4. Program Etherboot zaczyna skanowanie w celu odnalezienia karty sieciowej. Kiedy ją wykryje, nastąpi jej inicjalizacja.
5. Potem program Etherboot transmituje żądanie DHCP do lokalnej sieci. Żądanie będzie zawierać adres MAC karty sieciowej.
6. Demon DHCPD na serwerze zauważy żądanie transmisji ze stacji roboczej i sprawdzi pliki konfiguracyjne by odnaleźć wpis, który będzie pasował do adresu MAC stacji roboczej.
7. Jeśli taki wpis istnieje, demon DHCPD prześle pakiet zwrotny, zawierający dalsze informacje dla stacji roboczej. Informacja zwrotna zawiera:
   - adres IP stacji
   - ustawienia NETMASK dla lokalnej sieci
   - ścieżkę jądra do załadowania na stacji roboczej
   - ścieżkę podstawowego systemu plików do zainstalowania
   - opcjonalne parametry do przekazania dla jądra,
8. Program Etherboot otrzyma odpowiedź z serwera , i skonfiguruje odpowiednio interfejs TCP/IP karty sieciowej,
9. Używając protokołu TFTP, Etherboot zażąda od serwera załadowania jądra,
10. Kiedy całe jądro zostanie załadowane na stację, Etherboot umiejscowi je w odpowiednim miejscu w pamięci głównej.
11. Wówczas jądro przejmie kontrolę i zainicjalizuje cały system łącznie ze wszystkimi rozpoznanymi urządzeniami,

12. Uwieńczeniem powyższego jest obraz systemu plików. Zostanie on załadowany do pamięci głównej jako ramdisk i tymczasowo zamontowany jako główny system plików. Parametr **`root=/dev/ram0`** zleca jądru zamontowanie obrazu jako katalog główny.
13. Zazwyczaj, kiedy jądro zakończy ładowanie, uruchamia program **init**. Ale w tym przypadku instruujemy jądro za pomocą parametru linii poleceń **`init=/linuxrc`**, aby zamiast tego wykonało skrypt powłoki.
14. Skrypt **`/linuxrc`** rozpocznie skanowanie magistrali PCI w poszukiwaniu karty sieciowej. Dla każdego rozpoznanego urządzenia PCI nastąpi porównanie z plikiem znanych kart sieciowych **(`/etc/niclist`)**, Po odnalezieniu odpowiedniego porównania zostanie załadowany dany moduł jądra. Dla kart ISA, moduł bezsterujący musi być wyszczególniony w wierszu poleceń jądra, dodatkowo są przekazywane ewentualnie wymagane parametry adresu wejścia-wyjścia (IO) i IRQ (przerwanie).
15. Kiedy karta sieciowa zostanie zidentyfikowana, skrypt **`/linuxrc`** załaduje moduł jądra, obsługujący tę kartę.
16. Następnie zostanie rozpoczęty proces **dhclient**, aby wznowić zapytanie do serwera DHCP. To ponowne żądanie danych, tym razem w przestrzeni użytkownika, jest niezbędne z dwóch powodów: po pierwsze jeśli będziemy zależni tylko od zapytania otrzymanego z Etherboot, zostanie ono „połknięte" przez jądro, po drugie jądro zignoruje w tej sytuacji każdy serwer NFS, który mógł być wyszczególniony w ścieżce głównej (root – path) Jest to szczególnie ważne, jeśli chcemy serwer NFS był rozróżniany od serwera TFTP.
17. Kiedy **dhclient** otrzyma odpowiedź z serwera, uruchomi plik **`/etc/dhclient-script`**, i skonfiguruje interfejs **eth0**.
18. Do tego czasu główny system plików był dyskiem RAM. Teraz skrypt **`/linuxrc`** zamontuje nowy główny system plików przez NFS. Katalog eksportowany z serwera znajduje się w **`/opt/ltsp/i386`**. Nie można od razu zamontować nowego systemu plików jako **`/`** - najpierw musi być zainstalowany jako **`/mnt.`** Potem zostanie wykonany **pivot_root**. Pivot_root zamieni bieżący system plików na nowy system plików. Gdy ten proces się zakończy, system plików NFS zostanie zamontowany pod **`/`,** a stary system plików pod **`/oldroot.`**
19. Po zainstalowaniu i przesunięciu (pivoting) nowego głównego systemu plików, zostaje zakończony skrypt powłoki **`/linuxrc`** i uruchomiony właściwy program inicjujący **init**.
20. **Init** odczyta plik **`/etc/inittab`** i zacznie ustawianie środowiska stacji roboczej.
21. **Init** umożliwia tak zwany runlevel (poziom działania). Każdy runlevel definiuje różne usługi dla stacji roboczej. Stacja robocza LTSP startuje w poziomie 2. Jest to zaprogramowane przez wiersz **initdeafault** w pliku **inittab**.
22. Jednym z pierwszych wpisów w pliku **inittab** jest **rc.local** – komenda, która zostaje uruchomiona podczas gdy stacja robocza jest w stanie **„sysinit**".
23. Skrypt **rc.local** stworzy 1 mb ramdisk do zachowywania wszystkich rzeczy, które muszą być

zapisane lub zmodyfikowane w jakikolwiek sposób.

24. Ramdisk zostanie zainstalowany jako katalog **/tmp**. Wszystkie pliki, które mają być zapisane, będą rzeczywiście istniały w katalogu **/tmp**, a do tych plików istnieją symboliczne linki.
25. System plików **/proc** jest zamontowany.
26. Jeśli stacja robocza jest skonfigurowana do wymiany przez NFS, katalog **/var/opt/ltsp/swapfiles** zostanie zamontowany jako **/tmp/swapfiles**. W przypadku gdy, jeśli nie ma **swapfile** (pliku wymiany) dla stacji roboczej, zostanie on utworzony automatycznie. Rozmiar pliku wymiany jest skonfigurowany w pliku **lts.conf**. Plik wymiany uruchamiamy przy użyciu komendy **swapon.**
27. Interfejs sieciowy loopback (pętla zwrotna) zostanie skonfigurowany. Jest to interfejs sieciowy, który posiada adres IP: 127.0.0.1 .
28. Jeśli lokalne aplikacje są włączone, to montowany jest katalog **/home**, wówczas istnieje dostęp do katalogów domowych użytkowników.
29. W systemie plików **/tmp** są tworzone różne katalogi, po to aby, przechować pewne pliki, które są potrzebne podczas gdy system jest uruchomiony. Te katalogi to :
    - /tmp/compiled
    - /tmp/var
    - /tmp/var/run
    - /tmp/var/log
    - /tmp/var/lock
    - /tmp/var/lock/subsys
30. Teraz następuje konfiguracja systemu X Window. W pliku **lts.conf** znajduje się parametr nazwany XSERVER. Jeśli go brakuje lub gdy jest ustawiony na „auto", wtedy zostaje podjęta próba automatycznego wykrycia karty graficznej. Dla karty PCI, jest możliwe pobranie „PCI Vendor" i „Device ID", i porównane ich z plikiem **/etc/vidlist**.

    Jeśli karta jest wspierana przez Xfree864.X, to **pci-scan** zwróci nazwę modułu sterownika. Jeśli jest ona obsługiwana tylko przez Xfree86 3.3.6, **pci_scan** zwróci nazwę X-serwera do użycia. To wyjaśnia, który X-Serwer należy uruchomić. Skrypt **rc.local** może wykazać różnice, ponieważ starsze serwery 3.3.6 startują z nazwą zaczynającą się od 'XF86_'.

    Jeśli używamy Xfree86 4.x, wtedy przy pomocy skryptu **/etc/rc.setupx** zostanie utworzony plik XF86Config dla X4. Natomiast jeśli używamy jest XFree86 3.3.6, to skrypt /**etc/rc.setupx3** utworzy plik XF86Config. Plik Xfree86 Config zostanie utworzony na podstawie wpisów w pliku **/etc/lts.conf**.
31. Gdy skrypt **rc.setupx** jest zakończony, powracamy do **rc.local**. Wówczas zostanie utworzony skrypt **/tmp/start_ws**. Sktypt ten jest odpowiedzialny za uruchomienie X-serwera.
32. Teraz zostanie stworzony plik **/tmp/syslog.conf**. Ten plik zawiera informacje, dla demona syslogd, do której w sieci należy wysłać informacje o logowaniu. Syslog hosta należy

wyszczególnić w pliku lts.conf. Symboliczny link **`/etc/syslog.conf`** wskazuje na plik **`/tmp/syslog.conf`.**

33. Przy użyciu pliku configuracyjnego utworzonego w poprzednim etapie został uruchomiony demon **syslogd**.
34. Kontrola powraca do init. Ten decyduje na podstawie wpisu initdefault, który poziom działania ma być wykorzystany. Od czasu lts_core-2.08, wartość initdefault jest określona na 2.
35. Poziom 2 spowoduje, że init uruchomi skrypt **set_runlevel**, który odczyta plik lts.conf i określi, na którym poziomie zostanie uruchomiona stacja robocza.
36. Standardowymi poziomami dla LTSP są 3, 4 i 5.
    - Poziom **3**. Ten poziom uruchamia powłokę. Jest bardzo użyteczny do wykrywania błędów stacji roboczej.
    - Poziom **4.** Ten poziom uruchamia jedną lub więcej sesji Telnet w trybie znakowym. Jest użyteczny, gdy stare szeregowe terminale mają zostać zastąpione w prosty sposób.
    - Tryb **5** – poziom graficzny (GUI). X Window zostaje uruchomiony i przesyła zapytanie XDMCP do serwera, co spowoduje przesłanie z jego strony okienka dialogowego logowania i pozwoli na zalogowanie się na serwerze. W tym celu na serwerze musi być uruchomiony display-manager, jak np.: **XDM**,**GDM**, **KDM**.

Rozdział 2 Instalowanie LTSP na serwerze

Pakiet LTSP jest dostępny w dwóch formatach RPM i TGZ. Wybierz format, który wolisz zainstalować i postępuj wg. odpowiedniej części instrukcji.

2.1 Instalowanie pakietu RPM

Jeśli chcesz uruchomić X Window na stacji roboczej, należy zainstalować cztery pakiety. Należy pamiętać, że dla celów tego przykładu wybraliśmy stację roboczą z kartą sieciową opartą na „Tulip" i kartą graficzną bazującą na chipsecie Inteli810.

- pakiet rdzenia LTSP
- pakiet jądra
- pakiet rdzenia X
- pakiet czcionek X

Pakiet czcionek X nie jest konieczny, ale zalecany przy pierwszej instalacji. Kiedy zakończysz ustawianie serwera i bezdyskowej stacji roboczej, możesz skonfigurować serwer czcionek X (XFS). Po zainstalowaniu pakietów, musi zostać zainicjowany system LTSP. Ten proces obejmuje także zastosowanie zmian w systemie plików konfiguracyjnych, co jest konieczne, aby serwer mógł udostępnić jednej lub wielu stacjom roboczym żądane usługi.

2.1.1 Pakiet LTSP Core - pakiet rdzenia LTSP

Należy pobrać najnowszą wersję pakietów **ltsp** i zainstalować je przy użyciu komendy RPM.

```
rpm -ivh lts_core-.0.0.i386.rpm
rpm – ivh lts_kernel_3.0.0.i386.rpm
rpm-ivh lts_x_core-3.0.0.i386.rpm
rpm-ivh lts_x_fonts-3.0.0.i386.rpm
```

Powyższe komendy zainstalują pakiet.

2.1.2 Inicjalizacja serwera

Po zainstalowaniu powyższych pakietów, należy przejść do katalogu **`/opt/ltsp/templates.`** Są tam pliki które zmienią konfigurację serwera. Każdy z tych plików jest odpowiedzialny za jeden plik systemowy. Przejrzyj zawartość plików i upewnij się, czy chcesz wprowadzić zmiany. Potencjalne zmiany mogą zagrażać bezpieczeństwu twojego systemu. Zmiany w plikach systemowych mogą być wprowadzone ręcznie jeśli chcesz wprowadzić je automatycznie, uruchom komendę ltsp_initalize:

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

Skrypt każdorazowo zapyta czy skonfigurować daną usługę. Dotyczy to następujących usług:

- XDM – X Display Manager
- GDM – Gnome Display Manager
- Display manager startup script
- bootp
- plik NFS `/etc/exports`
- tcpwrappers
- port mapper
- syslogd
- TFTP startup script

2.1.3 Specyfikacja konfiguracji stacji roboczej

Serwer zawiera informacje dla poszczególnych stacji roboczych w trzech plikach.

- `/etc/dhcpd.conf`
- `/etc/hosts`
- `/etc/ltsp/i386/lts.conf`

2.1.3.1 /etc/dhcpd.conf

Stacja potrzebuje adresu IP i innych informacji. Otrzyma następujące informacje z serwera DHCP:

- adres IP
- hostname
- adres IP serwera
- default gateway
- nazwę ścieżki jądra, które ma być załadowane
- ścieżkę serwera i katalogu., który ma być zainstalowany jako główny system plików

W przykładzie, zostało wybrane DHCP do przydzielania stacjom roboczym stałych adresów IP.

Skrypt **ltsp_initialize** kopiuje przykładowy plik **dhcpd.conf** o nazwie **`/etc/dhcpd.conf.example`**, można go skopiować jako plik **`/etc/dhcpd.conf`** i używać go jako podstawy do własnej konfiguracji **dhcpd**. Oczywiście należy dostosować poszczególne części tego pliku do własnego środowiska systemowego i do stacji roboczych, a zwłaszcza do adresów sprzętowych MAC ich kart sieciowych.

```
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";
    }
}
```

Od wprowadzenia wersji 2.09pre2 LTSP, nie jest już konieczne wskazywanie konkretnego jądra do załadowania. Standardowe jądro zawiera wszystkie moduły kart sieciowych wspierane i obsługiwane przez Linux. Istnieją dwa jądra: jedno jądro zawiera Linux Progres Patch (LPP), a drugie nie. Można je rozpoznać po nazwach:

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

Należy zauważyć, że jądro jest umiejscowione w katalogu **`/tftpboot/lts`**, ale w polu „nazwa pliku“, w pliku `/etc/dhcpd.conf` brakuje części ścieżki **`/tftpboot.`** Jest to spowodowane tym, że w dystrybucji Redhat 7.1 i wyższych, TFTP uruchamia się z opcją ‘-s’. Oznacza to, że demon tfdpd uruchamia się w trybie “bezpieczny”. Podczas uruchomienia wykonywany jest chroot do katalogu `/tftboot`. Dlatego, wszystkie pliki , które są dostępne dla demona tfdpd odnoszą się do katalogu `/tftboot`.

W niektórych dystrybucjach Linuksa może nie być ustawionej opcji ‘-s’ dla tfdpd, wówczas należy dodać przedrostek `/tftpboot` przed nazwą ścieżki jądra.

2.1.3.2 /etc/hosts

Adresy IP i nazwy stacji

Komputery generalnie bardzo dobrze komunikują się za pomocą adresu IP. Ludzie wolą nadawać swoim komputerom nazwy, ponieważ nie potrafią zapamiętywać liczb. Do przyporządkowania nazw liczbom służy serwer DNS lub lub plik **`/etc/hosts`**. W środowisku LTSP nie można tego pominąć, gdyż stacja robocza nie będzie mogła wówczas zainstalować systemu plików przez NFS, gdyż ten wyświetli błędy dostępu.

2.1.3.3 /opt/ltsp/i386/lts.conf

W tym pliku znajduje się cały szereg wpisów konfiguracyjnych.

Plik lts.conf ma prostą składnię i składa się z wielu sekcji. Jest wiec sekcja domyślna dla wszystkich stacji roboczych (default) oraz indywidualne sekcje stacji roboczych. Stacje robocze mogą być identyfikowane przez nazwę, adres IP lub adres sprzętowy karty sieciowej MAC.

Typowy plik lts.conf wygląda następująco:

```
# 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
```

Oto lista niektórych wariantów konfiguracji:

XSERVER

Jeśli stacja robocza posiada kartę graficzną PCI, wspieraną przez Xfree86 4.1, wystarczy tylko zainstalować pakiet lts_x_core. Zawiera on wszystkie sterowniki dla wersji X4.

Dla LTSP są dostępne liczne pakiety Xfree86 3.3.6, na wypadek, gdyby twoja karta nie była wspierana przez Xfree86 4.1.

Dla indywidualnych stacji roboczych można dokonywać wpisów w pliku lts.conf , lub też zrobić wpis domyślny, który będzie wspólny dla wszystkich stacji.

Nasza przykładowa stacja posiada Inteli810 video chipset i może zostać odnaleziona automatycznie, dlatego nie są potrzebne żadne wpisy XSERVER w pliku lts.conf. Wpis XSERVER może zostać wyszczególniony, ustawiony na funkcję ‘auto’, by pokazać że będzie można wykryć go automatycznie.

RUNLEVEL

Jeżeli stacja robocza ma pracować w trybie graficznym, należy ustawić poziom „5” (runlevel). .

Rozdział 3 Ustawienia stacji roboczej

Teraz kiedy serwer jest zainstalowany, nadszedł czas aby skupić się na ustawieniu stacji roboczej.

Projekt LTSP rozpoczyna się wówczas gdy jądro znajdzie się w pamięci operacyjnej stacji roboczej. Istnieje wiele sposobów, aby umieścić jądro w pamięci: Etherboot, Netboot, PXE lub dyskietka.

Dla celów tej pracy została użyta dyskietka rozruchowa z kodem z projektu Etherboot.

3.1 Tworzenie dyskietki rozruchowej

Etherboot jest pakietem programowym służącym do tworzenia obrazu ROM. Obraz ten zawiera program, który może przeprowadzić załadowanie oprogramowania przez sieć Ethernet.

Wiele kart sieciowych posiada podstawkę w której można umieścić układ ROM. Etherboot jest kodem, który może być umieszczony w takiej pamięci ROM.

Etherboot jest udostępniany na zasadach Open Source i chroniony przez GNU General Public License, wersja 2 (GPL2).

Pakiet Etherboot można ściągnąć i skonfigurować go do wymaganego typu pamięci rozruchowej ROM. Później obraz ten można wgrać do kości EPROM lub skopiować na dyskietkę w celu przetestowania.

Dużo prostszym rozwiązaniem jest wejście na stronę Marty'ego Connor'a – www.Rom-O-Matic.net. Marty opracował doskonały frontend oparty o www do wygenerowania obrazu bootrom z Etherboot .

Na jego stronie wybiera się typ posiadanej karty sieciowej, oraz rodzaj obrazu jaki się chce uzyskać. Istnieje jeszcze wiele dalszych możliwości konfiguracyjnych. Po wprowadzeniu wszystkich danych wystarczy kliknąć przycisk „Get ROM”, aby wygenerować obraz.

Nasza przykładowa stacja robocza posiada kartę Linksys LNE100TX, wersja 4.1. Ta karta ma chipset ADMTek Centaur-P, dlatego należy wybrać centaur-p jako nasz typ NIC/ROM.

Nie jest konieczne wprowadzanie żadnych zmian w standardowej konfiguracji, dlatego można pominąć przycisk „Configure”.

Dla wyjściowego formatu pamięci rozruchowej należy wybrać ‘Floppy Bootable ROM Image’. To spowoduje, że powstanie nagłówek zawierający 512 bajtów jako boot-loader do właściwego obrazu Etherboot. Podczas uruchamiania komputera z dyskietki boot-loader ładuje obraz do pamięci operacyjnej, gdzie nastąpi jego wykonanie.

Naciśnij przycisk „Get Rom”. Wygenerowanie obrazu potrwa tylko chwilę. Po kilku sekundach przeglądarka wyświetli okienko ”Zapisz jako”. Tam można określić, w którym miejscu zapisać plik z obrazem bootrom.

W tym przykładzie obraz ma nazwę”eb-5.0.2-centaur-p.lzdsk” i można go zapisać do katalogu

tymczasowego */tmp/eb-5.0.2-centaur-p.lzdsk.*

Po zapisaniu obrazu na twardym dysku, należy go zapisać na dyskietkę. W tym celu trzeba umieścić dyskietkę w stacji i uruchomić następującą komendę:

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

Rozdział 4 Uruchamianie stacji roboczej

Zakładając, że serwer i stacja robocza zostały skonfigurowane prawidłowo, pozostawałaby już tylko kwestia włożenia dyskietki do napędu i włączenia stacji roboczej.

Program Etherboot zacznie wczytywać informacje z dyskietki do pamięci, karta sieciowa zostanie zidentyfikowana i zainicjowana, do sieci zostanie wysłane żądanie dhcp, serwer DHCP wyśle odpowiedź, a jądro zacznie się ładować na stację roboczą.

Kiedy jądro zainicjuje sprzęt stacji, uruchomi się X Window i na monitorze pojawi się okienko logowania, podobne do tego poniżej:

W tym momencie powinno być możliwe normalne zalogowanie się. Warto pamiętać, że logowanie odbywa się na serwerze. Wszystkie aplikacje uruchamiają się tak naprawdę na serwerze, a tylko wyświetlają się na stacji roboczej. To właśnie jest atut X Window.

Wszystkie programy znajdujące się na serwerze, do których masz uprawnienia są do Twojej dyspozycji.

Rozdział 5 Drukowanie

Poza tym, że stacja robocza jest w pełni funkcjonalnym terminalem graficznym lub znakowym, może także pełnić rolę serwera drukującego (print server), pozwalającego na przyłączenie 3 drukarek do portu równoległego lub portów szeregowych.
Wszystko to jest nieodczuwalne dla użytkownika stacji. Nawet nie zauważy dodatkowego obciążenia sieci.

5.1 Ustawienia po stronie klienta

LTSP używa programu lp_server, do przekierowywania zadań drukowania, z serwera do drukarki przyłączonej do jednego z portów na stacji roboczej.
W pliku lts.conf jest zestaw wpisów konfiguracyjnych do udostępnienia drukarki na stacji roboczej.

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

Powyższy wpis spowoduje, że program lp_server uruchomi się jako demon, oczekujący na porcie 9100 na zadania drukowania z serwera. Zadania drukowania zostaną przekierowane do drukarki podłączonej do portu równoległego **/dev/lp0.**
Jest wiele innych dostępnych opcji. Można je znaleźć w dalszej części tej pracy – w opisie lts.conf.

5.2 Ustawienia po stronie serwera

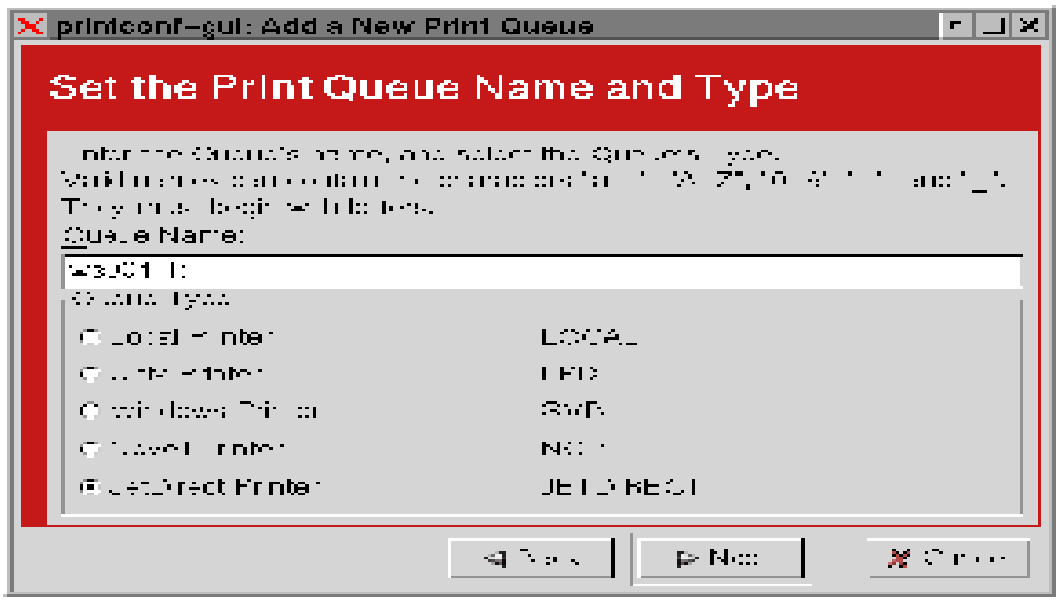
Aby zainstalować drukarkę na serwerze należy użyć odpowiednich narzędzi konfiguracyjnych.
Wersja Redhat 7.2 zawiera narzędzia konfigurowania drukarki, uruchamiane zarówno w trybie graficznym i znakowym. Printconf-gui jest programem graficznym, a narzędzie pracujące w trybie znakowym nosi nazwę printconf-tui. Starsze wersje Redhat zawierają program nazywany printtool. Printtool istnieje także w wersji Redhat 7.2, ale w naszym przykładzie używamy printconf-gui. Inne dystrybucje Linuksa mają swoje własne narzędzia programowe do konfigurowania drukarek.

Po uruchomieniu narzędzia, dodajemy nową drukarkę. Program lp_server pozwoli stacji roboczej emulować serwer drukowania HP JetDirect. W tym celu na serwerze musi zostać utworzona drukarka JetDirect.
Nowo utworzona drukarka potrzebuje własnej nazwy. Nazwa ta może być co prawda wybierana dowolnie, ale powinna być sensowna i składać się tylko z następujących znaków.

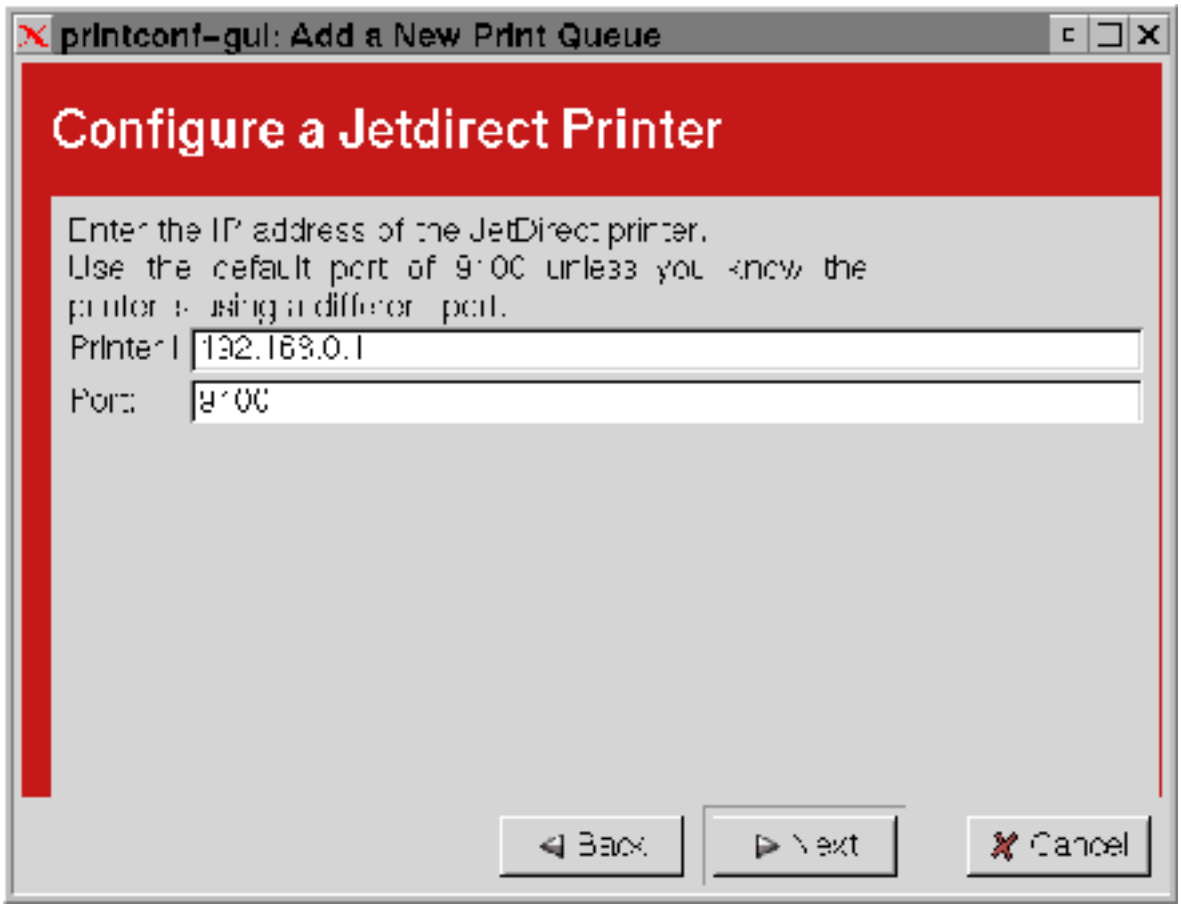
- „a-z” – małe litery
- “A-Z” – duże litery
- “0-9” – cyfry

- “-“ – myślniki
- “_“ – podkreślenia

W niniejszej pracy użyta jest nazwa **ws001_lp**. Nazwa ta pozwala skojarzyć, że drukarka jest podłączona do stacji roboczej **ws001.**



W okienku dialogowym należy uzupełnić dwa pola:



- 1. adres IP lub nazwę stacji roboczej, do której podłączona jest drukarka

- 2. port TCP – na którym nasłuchuje demon lp-server

Pierwsza drukarka, będzie na porcie **9100,** druga drukarka na porcie **9101**, a trzecia na **9102.**

Rozdział 6 Rozwiązywanie problemów

Jeśli, mimo postępowania wg poprzednich rozdziałów, twoja stacja nie uruchamia się, wtedy musisz rozpocząć proces wyszukiwania błędów, które mogły powstać podczas instalacji.
Pierwszą rzeczą jest uzmysłowienie sobie jak dalece zaawansowany jest proces rozruchu stacji roboczej.

6.1 Wyszukiwanie błędów na dyskietce

Jeśli stacja robocza jest uruchamiana z dyskietki można zobaczyć na ekranie coś w tym stylu:

```
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>
```

Jak pokazuje powyższy przykład, proces uruchamiania stacji można obserwować na monitorze. Jeśli nie pojawiły się te informacje, może oznaczać to, że dyskietka jest uszkodzona, lub też obraz został nieprawidłowo zapisany.
Jeśli pojawiły się informacje takie jak poniżej, można przypuszczać, iż na dyskietce został zapisany nieprawidłowy obraz.

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

Jeśli karta sieciowa została rozpoznana i wyświetlił się prawidłowy adres sprzętowy MAC, oznacza to, że dyskietka jest dobra, a błąd leży gdzie indziej.

6.2 Wyszukiwanie błędów związanych z DHCP

Kiedy karta sieciowa jest zainicjowana, prześle transmisje DHCP do lokalnej sieci, szukając serwera DHCP.
Jeśli stacja robocza otrzyma z serwera DHCP, to skonfiguruje kartę sieciową, a na monitorze powinien wyświetlić się adres IP. Oto przykład tego co powinien pokazywać ekran:

```
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
```

Jeżeli ostatnia z linijek rozpoczyna się od ‘Me:’ wiadomo, że DHCP działa poprawnie. Należy więc sprawdzić czy działa TFTP.

Jeśli jednak ekran wyświetli poniższą wiadomość, a po niej wiele zapisów <sleep>, oznacza to, że coś idzie nie tak. Może się też zdarzyć, że serwer odpowie dopiero po jednej lub dwóch informacjach <sleep>.

```
Searchin for server (DHCP) ...
```

Wykrycie błędu może być czasem trudne, oto kilka wskazówek, co należałoby sprawdzić w pierwszej kolejności.

6.2.1 Czy połączenia są prawidłowe?

Czy stacja robocza jest fizycznie prawidłowo podłączona do tej samej sieci co serwer?

Czy po włączeniu stacji roboczej, zapaliły się diody (link-LEDs) na przełączniku (switch).

Jeśli stacja i serwer są połączone bezpośrednio, należy sprawdzić, że użyty kabel jest typu cross-over. W przypadku gdy zastosowano przełącznik (switch), należy użyć zwykłego kabla ‘straight-thru’, zarówno pomiędzy stacją i przełącznikiem, jak i pomiędzy przełącznikiem a serwerem.

6.2.2 Czy proces dhcpd działa?

Istnieje kilka sposobów na sprawdzenie czy dhcpd uruchomił się na serwerze.

Zwykle dhcpd działa w tle, nasłuchując na porcie 67 udp. Aby to sprawdzić, należy uruchomić komendę netstat:

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

Ta komenda powinna spowodować wyświetlenie poniższej linii:

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

Czwarta kolumna zawiera adres IP i port, oddzielone dwukropkiem (‘:’). Adres zbudowany z samych zer wskazuje na to, że nasłuchiwanie trwa na wszystkich interfejsach sieciowych, zatem jeśli skonfigurowano dwa np.: eth0 i eth1, to dhcpd nasłuchuje na obydwóch interfejsach.

Sam fakt, że **netstat** pokazuje, iż cokolwiek nasłuchuje na porcie udp 67, nie oznacza wcale, że jest to dhcpd - może to być także bootpd. Jest to jednak mało prawdopodobne, ponieważ bootp nie jest już włączany do większości wersji Linuksa.

Aby upewnić się, że to jednak dhcpd, należy uruchomić komendę ps.

```
ps aux | grep dhcpd
```

Powinno to wyglądać mniej więcej tak:

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

Pierwsza linijka wskazuje, że **dhcpd** został uruchomiony. Druga linijka jest po prostu naszą komendą **grep.**

Jeśli nie wyświetliła się żadna linijka lub wyświetliła się tylko druga, należy sprawdzić, czy w skryptach startowych istnieje wpis uruchamiający serwer dhcpd.

Jeśli wpisy uruchamiające serwer dhcpd istnieją, a dhcpd nie działa można spróbować uruchomić serwer ręcznie. W tym celu w systemie Redhat konieczne jest wykonanie komendy:

```
/etc/init.d/dhcpd start
```

Należy zwrócić uwagę na ewentualne informacje o błędach.

6.2.3 Sprawdzanie plików konfiguracyjnych dhcpd

Czy w pliku /etc/dhcpd.conf istnieje wpis dla stacji roboczej?

Należy sprawdzić zwłaszcza ustawienie ‘fixed-adress’ w pliku konfiguracyjnym dhcpd, aby upewnić się, że dokładnie dopasowuje kartę do stacji roboczej.

6.2.4 Czy ipchains lub iptables nie blokują odpowiedzi?

6.2.4.1 Sprawdzanie ipchains

Należy uruchomić poniższą komendę i obserwować, co nastąpi:

```
ipchains –L -v
```

Jeśli wyświetli się coś takiego:

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

Oznacza to, że za błędy nie można winić ipchains.

6.2.4.2 Sprawdzanie iptables

Należy uruchomić poniższą komendę i obserwować, co nastąpi:

```
iptables -L -v
```

Jeśli wyświetli się coś takiego:

```
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
```

To znaczy, że za błędy nie można winić iptables.

6.2.5 Czy stacja robocza wysyła zapytanie?

Należy zaobserwować plik /**var/log/messages** podczas ładowania stacji roboczej. Służy do tego następująca komenda:

```
tail-f /var / log / massages
```

Dzięki powyższej komendzie będzie można obserwować na bieżąco jakie zmiany zachodzą w pliku.

```
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
```

Powyższe informacje, wskazują, że dhcpd działa, ale nie wie nic o pytającej maszynie. Należy wówczas skontrolować wpisy dla stacji roboczej w pliku /etc/dhcpd.conf.

6.3 Wyszukiwanie błędów związanych z TFTP

Etherboot używa TFTP do załadowania jądra Linuksa z serwera. Jest to prosty protokół, ale czasami TFTP sprawia problemy.

Jeśli na stacji roboczej podczas procesu ładowania zobaczysz wiadomość podobną do tej:

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

a przy tym ostatni znak w linijce (‘I’), szybko przeskakuje pomiędzy znakami ‘I’, ‘\’, ‘-‘ I ‘/’ (co troszkę przypomina kręcący się „pręt”), oznacza to, że jądro jest ładowane. Co zazwyczaj wskazuje na to, ze TFTP działa prawidłowo.

Jeśli nie widać tej „karuzeli”, niestety oznacza to problemy. Możliwe usterki mogą być następujące:

6.3.1 tftpd nie został uruchomiony

W wersji Redhat 7.1, tftp jest uruchamiany przez xinetd. Istnieje specjalny skrypt /etc/xinetd.d/tftp, uruchamiający tftpd.

6.3.2 Jądro nie jest tam gdzie spodziewa się go znaleźć tftpd

Jądro musi być ulokowane w takim miejscu, aby było dostępne dla demona tftpd. Jeśli do

wystartowania tftpd użyto opcji ‘-s’, wówczas wszystko, o co pytają stacje musi odnosić się do /tftpboot. Tak więc, jeśli zapis filename w pliku /etc/dhcpd.conf jest ustawiony na /lts/vmlinuz.tulip, wtedy jądro powinno być w /tftpboot/lts//vmlinuz.tulip

6.4 Wyszukiwanie błędów związanych z systemem plików NFS

Jest wiele rzeczy, które mogą przeszkodzić w zamontowaniu podstawowego systemu plików:

6.4.1 Nie odnaleziono init.

Jeśli pojawiła się następująca informacja o błędach:

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

najbardziej prawdopodobne jest, że została podjęta próba zainstalowania nieodpowiedniego katalogu, jako podstawowego systemu plików, lub katalog ltsroot jest pusty.

6.4.2 Serwer zgłasza błąd „error –13”

Jeśli pojawiła się następująca informacja o błędach:

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

Oznacza to, że w pliku **/etc/exports** nie ma wpisu o katalogu **/tftpboot/lts/ltsroot.**

W tej sytuacji należy sprawdzić plik logów **/var/log/massages,** który może zawierać pewne wskazówki. Wpis jak poniżej:

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

potwierdza, że zapis w /etc/exports nie jest poprawny.

6.4.3 Problemy z demonem NFS (portmap, nfsd i mountd)

Odnalezienie błędów NFS może być skomplikowane. Zrozumienie tego, co ma zostać skonfigurowane i jakie narzędzia do diagnozowania problemów są dostępne, z pewnością to ułatwi. Aby NFS działał poprawnie na serwerze muszą być uruchomione trzy demony: portmap, nfsd i mountd.

6.4.3.1 Portmapper (portmap)

Jeśli pojawiła się następująca wiadomość:

```
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
```

można przypuszczać, że demon portmap nie został uruchomiony. Można to sprawdzić przy użyciu komendy ps:

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

Jeśli portmapper działa wyświetli się wydruk taki jak:

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

Inną możliwością jest użycie polecenia netstat. Jako że portmapper używa portu 111 TCP i UDP, należy uruchomić co następuje:

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

Na wyjściu powinien pojawić się następujący wydruk:

```
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:*
```

Jeśli nic się nie wyświetliło, oznacza to, że portmapper nie działa. Aby go uruchomić ręcznie wykonaj:

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

Jeśli ten sposób doprowadził do uruchomienia portmappera, to należy zmodyfikować skrypty startowe tak, aby uruchamiał się on podczas ładowania systemu. W dystrybucji Redhat uruchom w tym celu ntsysv.

6.4.3.2 Demony NFS i MOUNT (nfs i mountd)

NFS ma dwa demony, które muszą funkcjonować, są to nsfd i mountd. Oba są uruchamiane przez skrypt /etc/rc.d/init.d/nfs

Poleceniem ps można sprawdzić czy są uruchomione.

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

Jeśli jeden z nich albo obydwa nie działają, należy je uruchomić.

Właściwie można tu skorzystać z argumentu restart, dostępnego dla skryptu **/etc/rc.d/init.d/nfs.**

Jeśli uruchomił się tylko mountd należy wprowadzić następującą sekwencję komend.

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

Mogą zaistnieć błędy przy komendzie stop, ale nie stanowi to problemu. Mimo to przy wykonaniu skryptu z argumentem start status powinien być OK.

Jeśli demony działają, a NFS nadal nie, posługując się poleceniem rpcinfo, można stwierdzić czy są zarejestrowane w portmapperze.

```
rpcinfo -p localhost
```

Powinny się wyświetlić następujące informacje:

```
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
```

Wskazują one, że nfs (nfsd) i mountd zarejestrowały się w portmapperze.

6.5 Poszukiwanie błędów związanych z X serwerem.

Najtrudniejszym zadaniem jest prawidłowe skonfigurowanie X serwera. Nie stanowi to problemu w przypadku zupełnie nowej karty graficznej wspieranej przez Xfree86 i monitora, który może poradzić sobie z różnym zakresem rozdzielczości i odświeżania ekranu. Jeśli w takim przypadku coś nie działa, to najprawdopodobniej X server jest nieodpowiedni dla twojej karty.

Można to łatwo sprawdzić: albo uruchamianie X servera zakończy się wyświetleniem informacji o błędach, albo wyświetlanie nie będzie prawidłowe.

X server uruchamia się na stacji roboczej poprzez /tpm/start_ws. Obywa się to automatycznie na poziomie 5, jako ostatni etap procesu ładowania, ale może być też przeprowadzone ręcznie na poziomie 3. Skrypt uruchomi X server za pomocą opcji –**query,** wskazującą na serwer, na którym działa manager wyświetlana, taki jak XDM, GDM lub KDM.

X serwer został uruchomiony przez skrypt start_ws, który sam jest uruchamiany przez program init. Jeśli skrypt zawiedzie, init jeszcze dziesięciokrotnie spróbuje ponownie uruchomić X server. Jeśli do tej pory stacja robocza nie została wyłączona, na monitorze pojawi się informacja X servera o błędzie.

Dziesięciokrotne oglądanie niepowodzeń X serwera może być irytujące. W tym wypadku najprostszym sposobem, jest uruchomienie stacji na poziomie 3, wówczas X server **nie jest uruchamiany automatycznie.** Po wystartowaniu na poziomie 3 pojawia się znak zachęty powłoki bash. – Można rozpocząć ręczne uruchamianie X servera przy pomocy następującej komendy:

```
sh  /tmp/start_ws
```

Zamiast po dziesięciu próbach, X server podda się już po pierwszej i poda informacje o błędach.

6.6 Wyszukiwanie błędów związanych z managerem okien

Display manager jest demonem, który uruchamia się na serwerze, oczekując na X server, aby nawiązać z nim kontakt. Kiedy kontakt zostanie nawiązany, wyświetli się okienko logowania na ekranie, oferując użytkownikowi możliwość zalogowania się na serwerze.

Trzema najpopularniejszymi menedżerami okien są:

- XDM – Jest częścią standardowego systemu X window.
- GDM – Gnome Display Manager. Jest częścią pakietu Gnome.
- KDM – KDE Display Manager. Jest to część systemu KDE.

Większość najnowszych dystrybucji GNU/Linux zawiera wszystkie trzy.

6.6.1 Szary ekran z dużym kursorem w kształcie X.

Oznacza to, że x server działa, ale nie może nawiązać kontaktu z menadżerem wyświetlania. Możliwymi przyczynami tego zjawiska mogą być:

- Menedżer wyświetlania nie działa. W dystrybucji Redhat 7.1, manager wyświetlania uruchamiany jest przez init. W pliku /etc/inittab, jest linijka, która wygląda jak ta poniżej:

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

Skrypt prefdm określi, który manager wyświetlania ma się uruchomić.

Domyśliny manager wyświetlania zależy od tego jaki pakiet został zainstalowany. Jeśli zainstalowano Gnome, to domyślnym DM jest GDM. Kiedy Gnom nie jest zainstalowany wtedy skrypt prefdm sprawdzi czy nie zainstalowano KDE. W tym wypadku domyślnym menadżerem jest KDM. Jeśli także KDM nie jest zainstalowany, to wtedy domyślnym menadżerem wyświetlania zostanie XDM.

Przy użyciu komendy netstat, można sprawdzić, który menadżer wyświetlania został uruchomiony. W tym celu zastosuj następujące polecenie:

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

Przebieg nasłuchiwania xdmcp na porcie 177 powinien być widoczny na ekranie.

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

Widać tu, że gdm działa z procesem numer 1493 i trwa nasłuchiwanie na porcie xdmcp.

Jeśli polecenie netstat zwróci linię podobną do powyższej, wiadomo, że trwa nasłuchiwanie menedżera wyświetlania, wtedy należy się upewnić, że stacja wysyła zapytanie XDMCP do właściwego serwera.

W pliku lts.conf, można wprowadzić wpis, który określa adres IP serwera, który ma uruchamiać menedżera wyświetlania. Wpis ten jest opcjonalny, więc jeżeli istnieje, powinien przypominać następujący:

```
XDM_SERVER  =  192.168.0.254
```

Oczywiście, adres IP dla danej sieci może się różnić od przykładu powyżej.

Jeśli wpis 'XDM_SERVER' jest nieobecny, można użyć wpisu 'SERVER'. Jeśli ten również jest nieobecny, używa się samego adresu IP (192.168.0.254)

Tak czy inaczej, należy się upewnić, że podany adres IP jest aktualnie adresem serwera uruchamiającego menedżera wyświetlania.

- Manager wyświetlania może być skonfigurowany tak, aby ignorował żądania ze zdalnych hostów.

Jeśli udało się określić, że menedżer wyświetlania działa, istnieje możliwość, że został on skonfigurowany tak, aby ignorował żądania XDMCP ze zdalnych maszyn. Należy więc sprawdzić, czy menadżer wyświetlania faktycznie został skonfigurowany na ignorowanie tych żądań.

- **XDM**

  W domyślnej konfiguracji, RedHat uniemożliwia logowanie się przez XDM. Powinien zadbać o to skrypt ltsp_initialize, ale jeśli tego nie zrobił, należy sprawdzić plik / etc/X11/xdm/xdm.config i odnaleźć w nim wpis podobny do tego poniżej:

```
DisplayManager.requestPort:     0
```

  Ten zapis musi być koniecznie wykomentowany (na początku linii musi być znak #), ażeby zapytania innych maszyn mogły być przyjmowane.

  Aby XDM mógł obsługiwać zapytania innych maszyn, ważna jest jeszcze zawartość pliku / etc/X11/xdm/Xaccess, który musi zawierać wiersz rozpoczynający się od gwiazdki '*'. Wpis ten zazwyczaj jest częścią tego pliku, jednak w RedHacie jest on wykomentowany. I tak jak wcześniej skrypt ltsp_initialize powinien zadbać o odkomentowanie tego wiersza. Jeśli XDM nadal nie będzie działał, należy sprawdzić ten plik ręcznie. Ważny wiersz powinien wyglądać tak:

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

- **KDM**

Nowsze wersje KDM mają plik o nazwie **kdmrs**. Niestety poszczególne dystrybucje Linuksa przechowują ten plik w różnych lokalizacjach. Dla Redhat 7.2 jest to **/etc/kde/kdm/kdmrc**. W innych, w celu zlokalizowania go, należy uruchomić polecenie **locate**.

Wpis, który kontroluje, czy odległe stacje mogą się zalogować, jest w sekcji **Xdmcp**. Należy upewnić się, że zapis Enable jest ustawiony na true.

Starsze wersje KDM używają plików konfiguracyjnych XDM, umieszczonych w **/etc/X11/xdm**

- **GDM**

GDM używa różnych plików konfiguracyjnych, znajdujących się w katalogu /etc/X11/gdm.

W głównym pliku gdm.conf należy sprawdzić sekcję Xdmcp, powinien się tam znajdować wpis 'Enable'. W zależności od wersji GDM musi on być ustawiony na '1' lub 'true'. Przykładowo:

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

Należy zauważyć, że w linii 'Enable=true', do włączania lub wyłączania zdalnych XDMPC, starsze wersje GDM używają '0' i '1', a wersje nowsze słów 'false' i 'true'.

Jeżeli stwierdzono, że menadżer wyświetlania działa i jest skonfigurowany tak, aby akceptował zapytania zdalnych maszyn, wyszukiwanie przyczyny błędów może jeszcze przebiegać następująco: Należy sprawdzić odwzorowanie adresów IP z nazwami hostów. Nazwy stacji roboczych muszą być zawarte w pliku /etc/hosts albo posiadać aktualny wpis w DNS.

Rozdział 7 Jądra

Dla stacji roboczych, można używać jądra standardowego, które jest dostępne do ściągnięcia, można też skonfigurować i skompilować własne jądro. Można wybrać, czy podczas ładowania systemu mają być wyświetlane informacje tekstowe, czy graficzny pasek postępu uruchamiania. Wyświetlanie graficznego paska jest możliwe poprzez LPP (Linux Progress Patch), ale tylko w przypadku kart graficznych z modułem VESA.

7.1 Standardowe jądra LTSP

Pakiet jądra z LTSP zawiera dwa jądra, jedno z **Linux Progress Patch** i drugie bez.
Oba jądra mają już zastosowaną łatkę, która umożliwia Swap przez NFS.

7.2 Kompilacja własnego jądra

Istnieją dwie możliwości, aby skonfigurować jądro dla LTSP. Najczęściej stosowaną metodą jest zastosowanie „Initial Ram Disk", w skrócie initrd. Obraz **initrd** jest małym systemem plików dołączonym do jądra. Obraz systemu plików initrd jest ładowany do pamięci i kiedy jądro jest już załadowane, montuje ten ramdisk jako główny system plików. Jest kilka korzyści z używania obrazu initrd. Po pierwsze, można skompilować sterowniki sieci jako moduły i załadować odpowiedni moduł podczas uruchamiania. To pozwala aby jedno jądro, wspierało w zasadzie wszystkie wszystkie karty sieciowe.

Inną korzyścią jest to, że możemy uruchomić klienta DHCP jako program „user-land" zamiast kernel-space". Uruchomienie klienta w user-land dostarcza lepszej kontroli nad opcjami otrzymywania i wysyłania inf. z serwera. To także czyni jądro nieznacznie mniejszym.

Jest jeszcze inna możliwość skonfigurowania jądra bez używania initrd. Budowanie jądra bez initrd wymaga, aby sterowniki kart sieciowych były statystycznie wkompilowane w jądro, a ponadto konieczne jest dodanie do konfiguracji jądra opcji IP-Autoconfig i „Root filesystem on NFS". Jądro bez initrd jest nieznacznie mniejsze i będzie się uruchamiało nieco szybciej. Zastosowanie jednej lub drugiej metody nie ma zasadniczego znaczenia dla funkcjonowania stacji roboczej.

Standardowe jądro LTSP zawiera Initial Ramdisk (initrd), który przejmuje wykrywanie karty sieciowej, i wysyła zapytanie DHCP w przestrzeni użytkownika. Istotne jest, aby obraz initrd uczynić tak małym jak to tylko możliwe. Dlatego zastosowano uClinux zamiast biblioteki libc ,a busybox dla narzędzi, potrzebnych podczas uruchamiania.

Aby kompilować własne jądra, powinieneś ściągnąć pakiet ltsp_initrd_kit. Zawiera on hierarchię głównego systemu plików, oraz skrypt do budowania obrazu.

7.2.1 Pozyskiwanie źródła jądra

Aby zbudować własne jądro, najlepiej pobrać najnowsze źródła jądra z ftp.kernel.org. Podyktowane

jest to tym, że poszczególne dystrybucje, jak np. Redhat, dodają do jąder wiele łat, tak że ich źródła jąder, różnią się od jądra oficjalnego.

Należy ściągnąć pożądany kod źródłowy jądra i zapisać go w katalogu /usr/src. Na serwerze ftp.kernel.org.ftp jądra znajdują się w katalogu /pub/linux/kernel. Należy wybrać jądro z serii 2.4.x, aby zapewnić sobie wsparcie dla devsf.

Dla obsługi NFS swapping lub Linux Progress Patch, należy utwierdzić się, że łaty te pasują do wersji jądra. Aktualnie (2004) najnowszą wersją jądra jest 2.4.22, które wspiera te dodatki.

Dla potrzeb niniejszej pracy użyto jądra 2.4.9.

```
ftp://ftp.kernel.org/pub/linux/kernel/v2.4/linux-2.4.9.tar.bz2
```

Po ściągnięciu źródeł należy je rozpakować do katalogu /usr/src. Uwaga: po rozpakowaniu, źródła jądra znajdą się w katalogu linux. Może się zdarzyć, że istnieje już katalog o takiej nazwie z innymi źródłami. Aby nie nadpisać danych w tym katalogu, należy zmienić jego nazwę przed rozpakowaniem źródeł.

Ściągnięta paczka jest skompresowana przy pomocy programu kompresującego bzip2. Dlatego należy go zdekompresować przed rozpakowaniem tarem. Przykładowo:

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

Rozpakowany katalog linux będzie zawierał całe drzewko źródła. W tym momencie, warto zmienić nazwę katalogu linux na nazwę kojarzącą się z wersją jądra, np. linux-2.4.9. Na przykład:

```
mv linux linux-2.4.9
```

Po wykonaniu powyższej czynności, należy przejść do katalogu linux-2.4.9:

```
cd linux-2.4.9
```

Przed rozpoczęciem konfiguracji nowego jądra dobrze jest zmodyfikować Makefile. Zaraz na początku pliku znajduje się zmienna nazywana EXTRAVERSION. Po ustawieniu jej na ‘ltsp-1’, aktualna wersja jądra będzie ‘2.4.9-ltsp-1’, co ułatwi później identyfikowanie jądra. Początek pliku Makefile powinien wyglądać następująco:

```
VERSION = 2
PATCHLEVEL = 4
SUBLEVEL = 9
EXTRAVERSION = -ltsp-1
KERNELRELEASE=$(VERSION).$(PATCHLEVEL).$(SUBLEVEL)$(EXTRAVERSION)
```

7.2.2 Łaty jądra

Po rozpakowaniu jądra, do dyspozycji są różne łaty, np. łatka NFS Swap lub Linux Progress Patch.

Te patche MUSZĄ być zaaplikowane przed rozpoczęciem konfiguracji jądra.

7.2.2.1 Łata NFS Swap

Łatka NFS Swap pozwala jądru stacji na używanie pliku wymiany, zlokalizowanego na serwerze NFS. Zazwyczaj zaleca się aby stacja robocza posiadała taką ilość pamięci RAM, aby nie musiała korzystać ze swap. Ale czasem wyposażenie stacji w większą ilość pamięci może być trudne, szczególnie w przypadku starych komputerów. W takich przypadkach NFS-swap przywróci funkcjonalność nieużytecznej maszynie.
Jeśli bieżącym katalogiem jest /usr/src/linux-2.4.9, a łatka znajduje się w /usr/src, można przetestować patch za pomocą poniższej komendy:

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

W ten sposób można sprawdzić, czy patch pasuje do jądra. Jeśli proces ten zakończył się bezbłędnie, można pominąć opcję **--dry-run** i zaaplikować źródłu łatę.

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

7.2.2.2 Linux Progress Patch (LPP)

Linux Progress Patch pozwoli na skonfigurowanie graficznego logo, wyświetlanego podczas procesu ładowania systemu. Pozostałe informacje o ładowaniu są przekierowywane na inną konsolę a specjalne instrukcje w skryptach ładujących zadbają o to, aby został wyświetlony graficzny pasek postępu uruchamiania systemu.
Do sprawdzenia łaty LPP, podobnie jak w przypadku NFS-swap można użyć następującego polecenia:

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

Jeśli test zakończy się sukcesem, wtedy można zaaplikować łatę w następujący sposób:

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

7.2.3 Konfigurowanie opcji jądra

Aby skonfigurować jądro można użyć jednego z następujących poleceń:

- **make xconfig**

  Jest to graficzna wersja narzędzia do konfiguracji jądra.

- **make menuconfig**

  To narzędzie wyświetla wygodne menu w konsoli tekstowej.

- **make config**

  Jest to interakcyjne narzędzie tekstowe do konfiguracji jądra z linii poleceń.

7.2.3.1 Konfiguracja jądra z initrd

Zastosowanie initrd wymaga następujących opcji:

- **File systems -> /dev filesystem support**

Należy włączyć opcję „/dev file system support", mieści się ona w sekcji "File systems". Nie należy włączać opcji "Automaticlay mount a boot" ponieważ montowanie odbywać się będzie za pomocą skryptu /linuxrc.

- **Block devices -> RAM disk support**

Stacje robocze LTSP wymagają obsługi RAM disk. Opcja ta znajduje się w sekcji 'Block devices'.

- **Block devices -> RAM disk support**

Również ta opcja musi być włączona

- **Processor type and features -> Processor family**

Należy się upewnić, że konfigurowane jądro będzie pasowało do procesora stacji roboczej. Można to ustawić w sekcji „Procesor type and features". Dla pojedynczego procesora należy wyłączyć obsługę SMP.

- **File systems -> Network file systems -> NFS Client support**

Stacja montuje swój system plików przez NFS, tak więc wymagana jest obsługa klienta NFS.
To były wymagane opcje. Można wyłączyć jeszcze wiele innych opcji, aby zredukować rozmiar jądra.

7.2.3.2 Konfiguracja jądra bez initrd

Konfigurowanie jądra bez initrd różni się w kilku punktach od konfigurowania jądra z initrd:

- **Block devices -> RAM disk support**

Stacje LTSP wymagają obsługi RAM disk

- **Block devices -> Initial RAM disk (initrd) support**

Ta opcja musi być wyłączona.

- **Networking options -> IP:kernel level autoconfiguration**

Ta opcja musi być włączona. Dzięki temu podczas ładowania jądro automatycznie skonfiguruje interfejs sieciowy eth0 na podstawie parametrów ładowania jądra.
Aktywowanie opcji DHCP, BOOTP lub RARP nie jest konieczne, ponieważ program Etherboot zawiera zawiera zapytania DHCP lub BOOTP i przekaże parametry IP do jądra. Dzięki temu jądro nie musi już generować własnych zapytań.

- **Network device support -> Ethernet (10 or 100Mbit)**

Rezygnując z initrd, należy wybrać odpowiedni sterownik posiadanej karty sieciowej. Musi to być statycznie powiązane z jądrem, ponieważ interfejs ethernet musi być podniesiony przed zamontowaniem systemu plików. To właśnie jest różnica w stosunku do jądra z initrd.

- **File systems -> /dev filesystem support**

Od wersji LTSP 2.09pre2, wymagana jest obsługa devfs. Niezależnie, czy używa się initrd czy nie, musi być włączona.

- **File systems -> Automatically mount at boot**

Jeśli nie używa się initrd, system plików jest montowany przez jądro, podczas jego ładowania. W tym wypadku należy aktywować tę opcję (‘Y’).

- **File systems -> Network file systems -> NFS Client support**

Stacja montuje swój system plików przez NFS, tak więc wymagana jest obsługa klienta NFS.

7.2.4 Kompilowanie jądra.

Dla ułatwienia, w pakiecie ltsp_initrd_kit załączony jest plik .config. Należy go skopiować do katalogu /usr/src/linux-2.4.9.

Po zakończeniu konfiguracji jądra, należy je skompilować. Do kompilacji służą następujące polecenia:

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

Można je wprowadzić w jednej linii, jak poniżej:

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

Podwójny znak „&” oznacza, że drugie polecenie będzie wykonane wtedy, gdy pierwsze zostanie zakończone poprawnie, a trzecie wtedy, po poprawnym zakończeniu drugiego, itd.

Po zakończonej kompilacji, nowe jądro znajduje się w:
/usr/src/linux-2.4.7/arch/i386/boot/bzImage.

7.2.5 Tagging jądra dla Etherboot.

Aby Etherboot poradził sobie z jądrem Linuks’a, musi być ono do tego odpowiednio przygotowane. Określa się to mianem „kernel tagging”. Tagging dodaje do jądra dodatkowy program, który jest wykonywany, zanim kontrolę przejmie jądro. Program do taggingu jądra nazywa się ‘mkbni-linux’. Pakiet ltsp_initrd_kit zawiera skrypt powłoki nazywany buildk, zawierający wszystkie polecenia, potrzebne, aby przygotować jądro do ładowania sieci.

Rozdział 8 Wpisy lts.conf

Podczas projektowania LTSP, było jasne, że jedną z trudniejszych kwestii będzie różnorodność sprzętu w stacjach roboczych. Niezależnie od tego jaka kombinacja procesora, karty sieciowej i graficznej zostałaby zastosowana, było oczywiste, że po trzech miesiącach, przy podłączaniu kolejnych stacji, będzie ona nieosiągalna.
Dlatego ważne jest, aby w pliku lts.conf istniały wpisy identyfikujące poszczególne stacje robocze, plik ten znajduje się w katalogu /opt/ltsp/i386/etc.
Format lts.conf pozwala na zastosowanie ustawień globalnych dla wszystkich stacji, a ponadto umożliwia zastosowanie indywidualnych ustawień poszczególnych maszyn. Jeśli wszystkie stacje są identyczne, wystarczy wprowadzić wszystkie ustawienia konfiguracyjne w sekcji ’Default’.

8.1 Przykładowy plik lts.conf

```
[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 Parametry pliku lts.conf

8.2.1 Parametry główne

Comments

Komentarze rozpoczynają się od znaku ‘#’

LTSP_BASEDIR

W opcji tej ustawia się, gdzie zlokalizowany jest podstawowy system plików LTSP. Wartość domyślna to /opt/ltsp

SERVER

Opcja ta wskazuje adres serwera dla XDM_SERVER, TELNET_HOST, XFS_SERVER i SYSLOG_HOST, chyba że dla określonej usługi zostanie wskazany inny serwer. Jeśli jest tylko jedna maszyna, która jest serwerem dla wszystkiego, można tu tylko określić adres i pominąć pozostałe parametry serwera. Jeśli ta wartość nie jest określona, będzie używana wartość domyślna: 192.168.0.254.

SYSLOG_HOST

Jeśli logi mają być przesyłane na inną maszynę niż domyślny serwer, można to określić w tym miejscu. Jeśli ten parametr nie jest określony, wtedy zostaną użyte parametry opcji ‘SERVER’ opisane powyżej.

NFS_SERVER

Parametr ten określa adres IP servera NFS, który ma być używany do montowania systemu plików /home. Jeśli ta wartość nie zostanie podana, wtedy zostaną użyte parametry opcji ‘SERVER’ opisane powyżej.

USE_NFS_SWAP

Ustawienie tej opcji na „Y” uruchamia swap/NFS. Wartością domyślną jest „N”.

SWAPFILE_SIZE

W tej opcji można określić rozmiar swapfile. Wartością domyślną jest 64M.

SWAP_SERVER

Plik wymiany może istnieć na dowolnym serwerze w sieci, pod warunkiem, że ma on wystarczającą pojemność dysku. W opcji tej można podać adres takiego serwera. Wartością domyślną jest ustawienie NFS_SERVER.

NFS_SWAPDIR

Katalog na serwerze, który jest eksportowany przez NFS. Wartością domyślną jest /var/opt/ltsp/swapfiles. Należy się upewnić, że katalog ten jest wpisany do pliku /etc/exports.

TELNET_HOST

Jeśli stacja ma być używana jako terminal znakowy, można tu podać, który komputer jest serwerem

Telnet dla tej stacji roboczej. Jeśli ta wartość nie jest ustawiona, wtedy zostaną użyte parametry opcji ‘SERVER’.

DNS_SERVER

Na podstawie tej opcji jest generowany plik resolv.conf stacji roboczej

SEARCH_DOMAIN

Również ta opcja generuje plik resolv.conf stacji roboczej.

MODULE_01 do MODULE_10

Dzięki tym wpisom można załadować do 10 modułów jądra. Wpisy te można podawać w taki sposób, jak po poleceniu insmond, przykładowo:

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

Jeśli wartością tego parametru jest pełna ścieżka, wtedy do załadowania modułu zostanie użyty insmod, w przeciwnym razie zostanie użyty modprobe.

RAMDISK_SIZE

Kiedy uruchamia się stacja robocza, tworzy ona ramdysk i montuje go w katalogu /tmp. Przy użyciu tego parametru można kontrolować rozmiar tego ramdysku. Wielkość tą należy określić w kilobajtach (1024 bajtów). Przykładowo: aby stworzyć ramdisk o pojemności 1MB, należy określić: **RAMDISK_SIZE=1024.**

Zmieniając tu rozmiar ramdysku, należy go także zmienić w jądrze. Jeśli wykonuje się tagging jądra za pomocą mknbi-linux, wielkość ta musi być albo statycznie wkompilowana, albo podana przy użyciu Etherboot lub Netboot.

Wartością domyślną jest 1024KB (1MB)

RCFILE_01 do RCFILE_10

Dodatkowe skrypty rc mogą być wykonywane prze skrypt rc.local. Wystarczy je umieść w katalogu /etc/rc.d i określić tutaj ich nazwę.

SOUND

Jeśli pakiet LTSP Sound jest zainstalowany, wartość musi być ustawiona na „Y”, aby uruchomić skrypt rc.sound, który skonfiguruje kartę dźwiękową i demona dźwięku. Wartością domyślną jest N.

8.2.2 Ustawienia X –Window

XDM_SEVER

Jeśli zapytanie XDM ma być wysłane do innego serwera niż domyślny, można to ustawić w tej opcji. Jeśli ten parametr nie zostanie określony, zostanie użyty parametr opcji ‘SERVER’.

XSERVER

Definiuje X SERVER, który uruchamiany jest na stacji roboczej. Dla kart AGP i PCI, parametr ten nie jest wymagany. Skrypt rc.local powinien automatycznie wykryć takie karty. Aby było to czytelne można ustawić tę wartość na „auto”.

Dla kart graficznych ISA, lub aby wybrać określony X server, należy w tym wpisie podać nazwę modułu sterownika albo X serwera.

Jeśli wpis rozpoczyna się od XF86_, wtedy zostanie użyty Xfree86 3.3.6. W przeciwnym razie zostanie użyty Xfree86 4.1.x. Wartością domyślną jest auto.

X_MODE_0 do X_MODE_2

Można tu podać do trzech modelines lub rozdzielczości ekranu. Wpis ten zapis przybierać dwie wartości. Może to być rozdzielczość ekranu lub kompletny modeline.

```
X_MODE_0 = 800x600
or
X_MODE_0 = 800x600 60.75 800 864 928 1088 600 616 621 657 –HSync –Vsync
```

Jeśli żaden X_MODE nie został określony, zostaną użyte domyślne modelines i rozdzielczości 1024x768, 800x600 i 640x480.

Jeśli określi się jeden lub więcej wpisów X_MODE, domyślne modelines nie zostaną użyte.

X_MOUSE_PROTOCOL

Może być tu wprowadzona każda wartość, współpracująca z Xfree86 Pointer Protocol. Typowe wartości to „Microsoft” i „PS/2”. Wartością domyślną jest „PS/2”.

X_MOUSE_DEVICE

Określa się tu urządzenie do którego podłączona jest mysz. Jeśli jest to mysz szeregowa, będzie to port szeregowy /dev/ttyS0 lub /dev/ttyS1. Jeśli jest mysz PS/2, urządzeniem będzie /dev/psaux. Domyślną wartością jest /dev/psaux.

X_MOUSE_RESOLUTION

Definiuje się tu rozdzielczość myszy. Typowa wartością dla myszy szeregowej jest 50, a dla PS/2 - 400. Wartością domyślną jest 400.

X_BUTTONS

W tym miejscu definiuje się ile przycisków ma mysz. Zwykle są to 2 lub 3. Wartością domyślną jest 3.

X_MOUSE_EMULATE3BTN

Jeśli opcja ta zostanie ustawiona na „Y” (YES), umożliwi emulowanie trzeciego klawisza myszy na myszy dwuklawiszowej. Wtedy trzeci klawisz będzie osiągalny poprzez szybkie kliknięcie obu klawiszy jednocześnie. Wartością domyślną jest N.

X_MOUSE_BAUD

Dla szeregowych myszy można tu zdefiniować baud rate. Wartością domyślną jest 1200.

X_COLOR_DEPTH

Można tu określić głębię kolorów. Dostępnymi wartościami są: 8, 15, 16, 24 i 32: 8 bitów = 256 kolorów, 16 = 65536 kolorów, 24 = 16 milionów kolorów a 32 = 4,2 bilionów kolorów!!! Nie każdy X server może obsługiwać wszystkie wartości. Wartością domyślną jest 16.

USE_XFS

Dla czcionek można uruchomić X Font Server (XFS) lub czytać je poprzez NFS. Serwer czcionek powinien ułatwić zarządzanie czcionkami w jednym centralnym miejscu, ale jeśli liczba stacji przekraczała 40 pojawiały się problemy. Dla tej opcji są dwie wartości „Y” i „N”. Wartością domyślną jest „N”. Jeśli ma być używany serwer czcionek, można użyć zapisu XFS_SERVER do określenia, który host w danej chwili będzie odgrywał rolę serwera czcionek.

XFS_SERVER

Jeśli używa się serwera czcionek, w opcji tej można podać jego adres IP. Jeśli opcja ta nie zostanie określona, zostanie użyta wartość opcji **SERVER**.

X_HORZSYNC

Określa się tu parametry HorizSync (odświeżanie poziome). Wartością domyślną jest „31-62”.

X_VERTREFRESH

Określa się tu parametry VertRefresh (odświeżanie pionowe). Wartością domyślną jest „55-90”.

XF86CONFIG_FILE

Posiadając własny kompletny plik XF86Config, można umieścić go w katalogu / tftboot/lts/ltsroot/etc. Wtedy jako wartość tej opcji należy podać jego nazwę.

```
XF86CONFIG_FILE = XF86Config.ws004
```

8.2.3 Parametry ekranu dotykowego

USE_TOUCH

Jeśli do stacji roboczej jest podłączony ekran dotykowy, można go włączyć ustawiając wpis na „Y”. Wartością domyślną jest „N”.

X_TOUCH_DEVICE

Taki ekran dotykowy pracuje podobnie jak mysz i zazwyczaj jest podłączany do portu szeregowego

stacji roboczej. W opcji tej należy podać port do którego ekran jest podłączony, np. /dev/ttyS0. Nie istnieje wartość domyślna dla tego wpisu.

X_TOUCH_MINX

Wartość kalibracyjna dla ekranu dotykowego EloTouch. Wartością domyślną jest 433.

X_TOUCH_MAXX

Wartość kalibracyjna dla ekranu dotykowego EloTouch. Wartością domyślną jest 3588.

X_TOUCH_MINY

Wartość kalibracyjna dla ekranu dotykowego EloTouch. Wartością domyślną jest 569.

X_TOUCH_MAXY

Wartość kalibracyjna dla ekranu dotykowego EloTouch. Wartością domyślną jest 3526.

X_TOUCH_UNDELAY

Wartość kalibracyjna dla ekranu dotykowego EloTouch. Wartością domyślną jest 10.

X_TOUCH_RPRDELAY

Wartość kalibracyjna dla ekranu dotykowego EloTouch. Wartością domyślną jest 10.

8.2.4 Parametry dla aplikacji uruchamianych lokalnie.

LOCAL_APPS

Aby mieć możliwość lokalnego uruchamiania aplikacji na stacji roboczej, należy ustawić tę opcję na Y. Pozostałe ustawienia muszą być wykonane po stronie serwera. – Patrz rozdział 9 „Lokalne aplikacje”. Wartością domyślną jest „N”.

NIS_DOMAIN

Jeśli opcja LOCAL_APPS jest ustawiona na „Y”, w danej sieci potrzebny będzie serwer NIS. Wpis NIS_DOMAIN określa nazwę domeny NIS. Musi ona być taka sama, jak ta, która jest zdefiniowana na serwerze NIS. **Domena NIS to nie to samo, co domena internetowa.** Wartością domyślną jest **ltsp**.

NIS_SERVER

Aby stacja robocza nie wysyłała zapytań o serwer NIS, należy w tej opcji ustawić adres tego serwera.

8.2.5 Parametry klawiatury

Wszystkie pliki obsługujące klawiaturę znajdują się w hierarchii ltsroot, tak więc skonfigurowanie obsługi klawiatury, właściwej dla danego języka, jest już tylko kwestią odpowiedniej konfiguracji Xfree 86. Do wyboru jest wiele parametrów konfiguracyjnych.
Nazwy i możliwe wartości dla poniższych parametrów pochodzą z dokumentacji Xfree 86.
Wszystkie wartości opcje dla Xfree 86 są ważne także tutaj.
W przyszłości, autorzy tej pracy, chętnie dołączyliby opisy ustawień potrzebnych do

skonfigurowania poszczególnych narodowych klawiatur. Jeśli uda się Państwu skonfigurować klawiaturę właściwą dla swojego języka, prosimy o poinformowanie autorów.

XkbTypes

Wartością domyślną jest słowo „default”.

XkbCompat

Wartością domyślną jest słowo „default”.

XkbSymbols

Wartością domyślną jest ‘us(pc101)

XkbModel

Wartością domyślną jest ‘pc101’

XkbLayout

Wartością domyślną jest ‘pl’.

8.2.6 Konfiguracja parametrów drukarki

Do stacji roboczej mogą być podłączone maksymalnie trzy drukarki. Kombinacja szeregowych i równoległych drukarek może być skonfigurowana przez następujące wpisy w pliku lts.conf:

PRINTER_0_DEVICE

Urządzenie do którego jest podłączona pierwsza drukarka. Dopuszczalne są nazwy takie jak / dev/lp0, /dev/tty0 lub /dev/ttyS1.

PRINTER_0_TYPE

Typ drukarki. Dopuszczalną wartością dla drukarki równoległej jest „P”, a dla drukarki szeregowej „S”.

PRINTER_0_PORT

Numer portu TCP/IP. Wartością domyślną jest ‘9100’.

PRINTER_0_SPEED

Dla drukarki szeregowej należy określić baud rate. Wartością domyślną jest ‘9600’.

PRINTER_0_FLOWCTRL

Dla drukarek szeregowych można określić rodzaj przepływu. Do kontroli programowej (XON/XOFF) służy opcja „S”, do kontroli sprzętowej opcja „H”. Wartością domyślną jest „S”.

PRINTER_0_PARITY

Dla drukarek szeregowych można określić parytet. Do wyboru są: ’E’-Even; ‘O’-Odd i ‘N’-None Wartością domyślną jest ‘N’.

PRINTER_0_DATABITS

Dla drukarek szeregowych można określić ilość bitów danych. Do wyboru są: ‘5’, ‘6’, ‘7’ i ‘8’. Wartością domyślną jest ‘8’.

PRINTER_1_DEVICE

Nazwa urządzenia, do którego podłączona jest druga drukarka.

PRINTER_1_TYPE

Typ drugiej drukarki.

PRINTER_1_PORT

Numer portu TCP/IP drugiej drukarki.

PRINTER_1_SPEED

Prędkość transmisji drugiej drukarki (szeregowej).

PRINTER_1_FLOWCTRL

Kontrola przepływu drugiej drukarki (szeregowej)

PRINTER_1_PARITY

Parity drugiej drukarki (szeregowej)

PRINTER_1_DATABITS

Ilość bitów danych drugiej drukarki (szeregowej)

PRINTER_2_DEVICE

Nazwa urządzenia, do którego podłączona jest trzecia drukarka

PRINTER_2_TYPE

Typ trzeciej drukarki

PRINTER_2_PORT

Port TCP/IP trzeciej drukarki.

PRINTER_2_SPEED

Prędkość transmisji trzeciej drukarki (szeregowej).

PRINTER_2_FLOWCTRL

Kontrola przepływu trzeciej drukarki (szeregowej)

PRINTER_2_PARITY

Parity trzeciej drukarki (szeregowej)

PRINTER_2_DATABITS

Ilość bitów danych trzeciej drukarki (szeregowej)

Rozdział 9 Aplikacje lokalne

Środowisko LTSP daje do wyboru dwa sposoby uruchamiania aplikacji - lokalnie na stacji roboczej lub zdalnie na serwerze.

Łatwiejszą metodą jest uruchamianie aplikacji na serwerze. Oznacza to, że aplikacje klienta uruchamiane są na serwerze, przy użyciu procesora i pamięci serwera, natomiast wynik wyświetlany jest na ekranie stacji roboczej, a do komunikacji z programami używa się myszy i klawiatury, podłączonych do stacji roboczej.

Jest to zasadnicza zaleta X window. Stacja robocza pracuje jako standardowy terminal X Window.

Aby użytkownik mógł uruchomić aplikację na stacji roboczej, stacja potrzebuje uzyskać o nim podstawowe informacje, takie jak:

- ID użytkownika
- podstawowa grupa, do której należy użytkownik
- katalog domowy użytkownika

LTSP używa NIS- Network Information Service (wcześniej zwany Żołtymi Stronami), aby udostępnić te informacje stacji roboczej.

9.1 Korzyści wypływające z lokalnego uruchamiania aplikacji.

-Redukcja obciążenia serwera. W dużych sieciach zawierających pamięciożerne aplikacje, jak np. Netscape, uruchamianie aplikacji na stacji może poprawić wydajność, o ile stacja robocza dysponuje wystarczająco mocnym procesorem i odpowiednią ilością pamięci RAM, aby sobie z tym poradzić.

- uruchamianie procesów (tzw. „runaway") lokalnie nie wpływa na obciążenie serwera i nie ma negatywnego wpływu na pracę innych użytkowników,
- obsługa dźwięku jest dużo łatwiejsza do skonfigurowania, kiedy aplikacja wydająca dźwięki jest uruchamiana na stacji roboczej.

9.2 Co należy wziąć pod uwagę stosując lokalne aplikacje?

Uruchamianie lokalnych aplikacji wymaga więcej pracy.

- Lokalnie uruchamiane aplikacje wymagają zastosowania mocniejszego sprzętu w stacji roboczej. Potrzeba większej ilości pamięci RAM i szybszego procesora. Zalecane jest minimum: 64 MB pamięci RAM.
- NIS - aby uruchomić aplikacje na stacji, należy się najpierw zidentyfikować. Do tego jest potrzebne hasło potwierdzające autentyczność. Metodą autentykacji użytkowników w sieci jest NIS.
- Do uruchamiania lokalnych aplikacji należy wyeksportować dodatkowe katalogi do

zamontowania przez NFS.

- Aplikacje uruchamiają się wolniej, ponieważ muszą być odczytywane przez NFS, co dodatkowo powoduje wzrost obciążenia sieci. Ponieważ każda kopia programu jest uruchamiana na własnym procesorze, rezygnuje się z korzyści wspólnych segmentów programu w pamięci RAM. Wspólne segmenty programu umożliwiają szybsze uruchamianie kolejnych instancji programu

9.3 Konfiguracja serwera dla lokalnych aplikacji.

9.3.1 Wpisy w lts.conf.

W pliku lts.conf należy dokonać kilku wpisów:

LOCAL_APPS

Wartość tą należy ustawić na Y. – Spowoduje to uruchamianie następujących procesów:

- Katalog /home zostanie zamontowany przez NFS.
- Na stacji roboczej zostanie utworzony katalog **/var/y/nicknames**.
- Uruchomi się portmapper.
- Uruchomi się xinetd.
- Zostanie utworzony plik /etc/yp.conf
- Zostanie wykonane polecenie domainname zgodnie z wartością NIS_DOMAIN w pliku lts.conf.
- Zostanie uruchomiony ypbind

NIS_DOMAIN

NIS – serwer i wszystkie przynależące stacje robocze muszą należeć do tej samej domeny NIS (nie jest to w żaden sposób powiązane z domeną DNS).

NIS_SERVER

Stacja robocza spróbuje połączyć się z określonym w tej opcji serwerem NIS, albo wyśle do sieci broadcast i będzie czekać na odpowiedź serwera. Chcąc wskazać określony serwer NIS, należy wprowadzić w tym miejscu jego adres IP.

9.3.2 Network Information Service - NIS

NIS jest usługą typu klient-server. Na serwerze działa demon, który przyjmuje żądania od klientów (stacji). Ten proces nazywa się **ypserv**.

Na stacji przebiega proces określany jako ypbind. Kiedy stacja szuka informacji o użytkowniku, takich jak weryfikacja hasła lub odnalezienie katalogu domowego, wtedy użyje ypbind aby ustanowić połączenie z demonem ypserv na serwerze.

Jeśli w sieci jest już używany NIS, nie ma potrzeby konfigurowania serwera LTSP jako dodatkowego serwera NIS. Należy tylko w pliku lts.conf dopisać wartości dla NIS_DOMAINNAME i NIS_SERVER, pasujące do danej sieci.

Jeśli w sieci nie był do tej pory używany NIS, należy skonfigurować i uruchomić ypserv. Wyczerpujące informacje na temat instalacji serwera NIS można znaleźć w The Linux NIS(YP) NYS/NIS+HOWTO. Dodatkowe informacje znajdują się w tekstach źródłowych wymienionych na końcu tej pracy.

9.4 Konfigurowanie aplikacji

Aby aplikacja uruchamiała się na stacji roboczej, należy umieścić wszystkie jej składniki w miejscu miejscu osiągalnym dla stacji.
W przypadku starszych wersji LTSP (2.08 i wcześniejsze), wiele katalogów serwera było eksportowanych przez NFS i montowanych na stacji – min.: /bin, /usr/bin, /lib i /usr.
Z powyższym postępowaniem łączy się pewien problem, mianowicie funkcjonuje ono tylko wówczas, gdy stacja i serwer mają taką samą architekturę. Nawet taka różnica, gdy serwer jest Pentium II (i686) a stacja jest Pentium (i586) może stanowić problem, ponieważ serwer będzie wolał mieć biblioteki i686, a nie i386, i486 czy i586.
Dlatego najprostszym rozwiązaniem dla stacji klienckiej jest, posiadanie kompletnego drzewa katalogów, ze wszystkimi programami i bibliotekami, których potrzebuje, niezależnego od programów i bibliotek serwera.
Wszystkie części, których potrzebuje lokalna aplikacja, muszą znajdować się w odpowiednich katalogach drzewa. Jednym z pakietów dostępnych do ściągnięcia ze strony LTSP jest lokalny pakiet netscape, instalujący dużą ilość plików w katalogu /opt/ltsp/i386/usr/local/netscape. Znajdują się tam: java classes, pliki pomocy, wykonywalne pliki binarne i skrypty.
Netscape nie wymaga żadnych dodatkowych bibliotek systemowych, tak więc nie ma konieczności dodawania czegokolwiek do katalogu /opt/ltsp/i386/lib. Inne aplikacje mogą jednak wymagać dodatkowych bibliotek.
Aby określić, które biblioteki są konieczne - można wykorzystać polecenie ldd.
W poniższym przykładzie aplikacją uruchamianą lokanie będzie gaim. gaim jest programem typu Instant Massenger i służy do komunikowania się z innymi użytkownikami w Internecie.
Pierwszą rzeczą jaką należy zrobić jest odnalezienie wykonywanego pliku binarnego gaim. w dystrybucji Redhat 7.2, znajduje się on w katalogu /usr/bin. W poszukiwaniu mogą być pomocne narzędzia takie jak type i which.
Po zlokalizowaniu pliku wykonywalnego gaim, można za pomocą ldd wyszukać potrzebne biblioteki:

```
[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
      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)
```

Powyższy listing pokazuje wszystkie biblioteki, dołączane dynamicznie z programem gaim. Większość programów wykorzystujących wspólne biblioteki, wymaga loadera ld-linux, aby lokalizować i ładować poszczególne biblioteki. W niektórych programach biblioteki ładuje się ręcznie za pomocą funkcji dlopen(). Dla tych aplikacji, polecenie ldd nie pokaże bibliotek. W takim przypadku, można wykorzystać polecenie strace, które pokaże wszystkie wezwania dlopen() z nazwami bibliotek.
Kiedy lista bibliotek będzie gotowa, wymagane biblioteki należy skopiować do właściwego miejsca w drzewie /opt/ltsp/i386.

9.5 Uruchamianie lokalnych aplikacji.

Programy X window, zazwyczaj uruchamiają się tam, gdzie działa menadżer okien. Tak więc, jeśli menadżer okien działa na serwerze i efekty są wyświetlane na stacji, wtedy każda aplikacja będzie uruchamiała się również na serwerze i wysyłała efekty na stację.
Cała sztuka w przypadku aplikacji lokalnych polega na tym, że serwer nakazuje stacji roboczej, aby uruchomiła program. Zazwyczaj dzieje się tak po zastosowaniu polecenia rsh.
Poniższy przykład pokazuje jak uruchomić program gaim na stacji roboczej:

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

Powyższy przykład może być wprowadzony w oknie xterm, lub też umieszczony w skrypcie powłoki który może być uruchamiany przez ikonę na pulpicie.

Uruchamianie lokalnego Netscape odbywa się w podobny sposób, jednak przed uruchomiemiem należy ustawić zmienną środowiskową.

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

Rozdział 10 Przykłady konfiguracji

Prawie wszystkie cechy stacji roboczej mogą być ustawione za pomocą ustawień w pliku lts.conf. Plik ten znajduje się zwykle w /opt/ltsp/i386/etc

10.1 Mysz szeregowa

Przykładowy wpis w pliku lts.conf dla dwuklawiszowej myszy szeregowej.

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

10.2 Mysz PS/2 z kółkiem

Przykładowy wpis w pliku lts.conf dla myszy PS/2 z kółkiem.

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

10.3 Drukarka USB przy ThinkNic

Stacja robocza ThinkNic posiada port USB, który może być używany dla drukarki. Przykładowe wpisy dla takiej drukarki w lts.conf to:

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

10.4 Konfiguracja X serwera z wykorzystaniem Xfree86 3.3.6

Przez ustawienia domyślne konfiguruje się Xfree86 4.1.0 dla stacji roboczej. Dla starszych kart graficznych, które nie są już wspierane przez wersję 4.1.0, należy najpierw zainstalować wersję 3.3.6 X serwera. Następnie w pliku lts.conf opcji XSERVER należy wprowadzić wpis jak w poniższym przykładzie:

```
XSERVER             = XF86_SVGA
```

Rozdział 11 Inne źródła informacji.

11.1 W Internecie

- Strona projektu LTSP
  www.LTSP.org
- Diskless−Nodes HOW−TO document for Linux
  www.linuxdoc.org/HOWTO/Diskless−HOWTO.html
- Strona domowa projektu Etherboot
  etherboot.sourceforge.net
- Serwis Rom−O−Matic
  www.Rom−O−Matic.net
- XFree86−Video−Timings−HOWTO
  www.linuxdoc.org/HOWTO/XFree86−Video−Timings−HOWTO.html
- The Linux NIS(YP)/NYS/NIS+ HOWTO
  www.linuxdoc.org/HOWTO/NIS−HOWTO.html

11.2 Publikacje drukowane

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