Diskless 리눅스 미니 하우투

by Robert Nemkin buci@math.klte.hu v0.0.3, 1996년 9월 12일

kmw@physics3.sogang.ac.kr 1999년 2월 12일

이 문서는 디스크 없는 리눅스 박스를 어떻게 설정하는지를 나타내고 저작권은 Robert Nemkin에게 있다. 저작권은 GPL을 따른다. 나는 Bela Kis <bkis@cartan.math.klte.hu> 가 이 문서를 영어로 번역해준데 대하여 감사한다.

목 차

제 1 절 변화들

- v0.0.3 1996년 9월 12일 : 몇몇 사소한 버그수정

제 2 절 디스크 없는 리눅스 박스를 어떻게 설치할 것인가

이 문서는 디스크 없는 리눅스 박스의 설치에 관한 것이다. 때때로, 하드 디스크나 플로피 드라이브 없이 PC에서 리눅스를 실행할 필요가 있을 것이다. 만약, 네트워으로 연결되어 있다면, 다른 유닉스 시스템과 bootp, tftp, NFS 서버, 그리고 eprom 버너 등을 이용하여 하드 또는 플로피 디스크 없이 리눅스를 설치하고 작동하게 하는 것이 가능하다.

제 3 절 관계된 문서들

- NFS-root 미니 하우투
- Terry Dawson이 쓴 리눅스 NET-2-3-하우투 , 94004531@postoffice.csu.edu.au
- 새로운 커널의 컴파일과 설정에 관한 /usr/src/linux/README

제 4 절 하드웨어

여기에 설명되어진 것들은 아래의 설정에 의해서 확인되었다.

- 부트 서버로서의 Sun-OS 4.1.3
- Slackware 2.3 + Linux 1.2.8 + wd 8013 ethercard.
- Working Ethernet lan

제 5 절 기본적인 아이디어들

기본적인 생각은 다음과 같다. ; PC는 tftp 프로토콜을 경유하여 커널과 초기 IP 주소로서 0.0.0.0을 사용한 bootp 프로토콜에 의한 부트서버로부터 IP주소를 얻을 것이다.[1]

아래에는 이 과정들을 열거한 것이다.

5.1 PC를 설치하기

nfs 패키지 (유명한 리눅스의 미러사이트의 /pub/Linux/system/Linux-boot디렉토리에서 찾을 수 있는)를 얻어라. 여기에서 즉시 사용할 수 있는 wd8013 카드에 관한 booteprom에 관한 이미지가 포함되어있다.

PC를 준비하는데는 여러 가지 방법이 있다. ;

- 만약 당신의 머신에 디스크가 없지 않다면, 작은 도스 프로그램을 사용하라. 또는,
- 같은 패키지 안에는 바이너리 플로피 이미지가 포함되어있다.

만약 당신이 마지막 옵션을 선택한다면, dd 명령을 통하여 플로피에 이미지를 써넣어야만 한 다.

이 이미지들은 bootp와 tftp 클라이언트를 포함하고 있다. 당신은 nfs-root옵션을 포함한 리눅스 커널또한 준비할 필요가 있다.

- 만약 당신이 최근의 안정된 커널인 linux-1.2.13을 사용한다면, 당신은 nfs 패키지 안에 포함된 패치 파일을 통하여 커널을 패치 하여야 한다.[2]

[1](라우터를 경유하여)교차된 세그먼트로 부팅 하는 것은 쉬운 문제가 아니다. 그래서, 서버와 디스크 없는 리눅스 박스를 같은 랜 세그먼트에 두거나 서버의 주소와 관계된 당신의 라우터에서 UDP Helper의 주소를 설정하라. 더자세한 정보에 관해서는 라우터의 매뉴얼을 참고하기 바란다.

[2]패치(1)를 참고하라

- 만약 당신이 최근의 것인 linux-1.3.x 시리즈의 불안정한 커널을 사용한다면, nfs-root 옵션에서 설정하여야만 한다.

당신은 블록 디바이스 (플로피 또는 하드디스크)를 설정할 필요는 없으나, tcp/ip와 wd-ethernet 카드와 nfs 파일시스템을 선택하여야한다. 그리고 일상적으로 커널을 재 컴파일 하라.

5.2 서버에서 bootpd를 설치하기

만약당신이 `bootpd-2.4.tar.gz` (당신의 주요 리눅스 미러사이트인 `/pub/Linux/system/Network/boot.net` 디렉토리)패키지를 발견한다면, 패키지를 얻고, 컴파일한후 인스톨하라. 만약 당신의 유닉스 박스가 슬랙웨어 리눅스 배포판이라면 bootpd를 포함하고 있으므로 이 과정을 생략 하라. 데몬은 다음 명령에 의해서 즉시 실행된다.

```
            bootpd -s
```

또는 inetd를 사용하여, 이런 경우에는 다음을 편집해야한다.

- /etc/inetd.conf에서 다음의 시작 라인에서 해시 마크를 지워라.

```
         # tftp   dgram   udp     wait    root    /usr/sbin/in.tftpd        tftpd /export
         # bootps dgram   udp     wait    root    /usr/sbin/in.bootpd     bootpd
```

- /etc/services에서 다음의 두라인을 주석을 없애거나 또는 수정하라.

```
         bootps          67/tcp          # BOOTP server
         tftp            69/udp           # TFTP server
```

- 다음의 명령에 의해서 inetd를 재 시작하라.

```
                kill -HUP <process id of inetd>.
```

5.3 서버에서 bootpd를 설정하기

무엇보다도, bootpd는 /etc 디렉토리에서 bootptab이라 불리는 설정 파일을 가지고 있다. 당신은 이 파일에서 게이트웨이, 도메인 네임 서버, 그리고 당신의 디스크 없는 머신의 이더넷 주소 등을 수정하여야만 한다. 다음은 /etc/bootptab의 한 예이다.

```
global.prof:\
        :sm=255.255.255.0:\
        :ds=192.168.1.5:\
        :gw=192.168.1.19:\
        :ht=ethernet:\
        :bf=linux:
machine1:hd=/export/root/machine1:tc=global.prof:ha=0000c0863d7a:ip=192.168.1.140:
machine2:hd=/export/root/machine2:tc=global.prof:ha=0800110244e1:ip=192.168.1.141:
machine3:hd=/export/root/machine3:tc=global.prof:ha=0800110244de:ip=192.168.1.142:
```

global.conf는 일반적인 호스트에 관한 내용들을 가지고있는 파일이고 여기에서,

- sm 부분은 서브넷 마스크를 포함한다.
- ds 부분은 도메인 네임 서버의 주소를 포함한다.
- gw 부분은 기본 게이트웨이의 주소를 포함한다.
- ht 부분은 랜의 하드웨어 형태를 포함한다.

- bf 부분은 부트 파일의 이름을 포함한다.

이것이 끝난 후에, 각 머신들은 다음의 라인을 가지고 있어야 한다. ;

- 처음 부분은 호스트의 이름을 포함하고,
- hd 부분은 부트 파일의 디렉토리 부분을 포함하고,
- 전체 설정은 tc 부분에 포함되어 질 수 있고,
- ha 부분은 이더넷카드의 하드웨어 주소를 포함하고,
- ip 부분은 할당된 ip 주소를 포함한다.

5.4 tftp를 이해하기

TFTP (`Trivial File Transfer Protocol`) 은 ftp와 같은 파일 전송 프로토콜이지만, EPROM에 있어서 코딩을 하는데 훨씬 간단하다. TFTP는 다음의 두 가지 방법에 의해 사용될 수 있다.

- 간단한 tftp: 클라이언트가 당신의 전체 파일 시스템을 액세스 할 수 있음을 의미한다. 이것은 간단하나 보안에 큰 구멍 (누구나 tftp를 통하여 당신의 패스워드 파일을 가질 수 있다.)이다.
- 안전한 tftp: tftp 서버는 chroot를 사용한다. 두 개의 시스템 콜이 자신의 루트 디렉토리를 변경 한다. 새로운 루트 디렉토리를 벗어난 어떤 것도 완전히 액세스 할 수 없을 것이다. chroot 때문에 디렉토리는 새로운 루트 디렉토리가 될 것이고, bootptab 안의 hd 파일은 새로운 상황을 반영한다. 예를 들어 ; 불안전한 tftp를 사용했을 때, hd 부분은 부트 디렉토리로서 /export를 사용하면, /export는 / 가 되고 hd 부분은 /root/machine1이 된다.

거의 모든 유닉스의 구현에 있어서 tftp 서버를 포함하고, 아마 스스로 인스톨할 일은 없을 것이다.

5.5 원격 서버에서 최소의 리눅스 설정을 설치하기

이것은 슬랙웨어 배포 본에서 a, ap, n과 x의 패키지에 포함되어있다. 모두 인스톨하는 것은 좋다. ; 그러나 위의 패키지들은 디스크 없는 X 터미널로서의 목적은 충분하다. 인스톨에 관해서, 당신은 리눅스 시스템을 이용해야한다. 원격 머신의 디스크 공간을 확인하고 그것은 읽기-쓰기로 export 하라. 리눅스 박스의 파일시스템에서 어떤 (/mnt)같은 export된 디렉토리를 마운트한다. 리눅스 설치를 시작하고 /로부터 /mnt로의 설치에서 루트 옵션을 변경하라. 그후 일방적으로 패키지를 설치하라. 만약 당신이 하나의 디스크 없는 리눅스 박스만을 실행하기를 원한다면, 바꿀 필요가 아무 것도 없다. 다른 한편으로, 만약 당신이 하나의 디스크 없는 머신 이상을 사용하기를 원한다면, 한 머신에 관한 개인적인 파일과 디렉토리만을 포함하므로 위의 설치는 작동하지 않을 것이다. 문제는 /usr (개인적인 데이터를 포함하지 않은 부분)을 옮기고 각각의 디스크 없는 머신의 디렉토리를 설정하므로 피할 수 없게된다. 예를 들어, 만약 /export/linux/machine/ 이 /mnt에 마운트되어 있다면, 초기의 설치를 한 후에 디렉토리 구조가 다음과 같을 것이다.

```
/export/linux/machine1/bin
/export/linux/machine1/sbin
/export/linux/machine1/lib
/export/linux/machine1/etc
/export/linux/machine1/var
/export/linux/machine1/usr
```

설정을 변경한 후에는 다음과 같을 것이다.

```
/export/linux/machine1/bin
/export/linux/machine1/sbin
/export/linux/machine1/lib
/export/linux/machine1/etc
/export/linux/machine1/var
/export/linux/usr
```

지금 다른 머신에 관한 서브디렉토리를 만들라. 지금, 당신의 디스크 없는 머신이 machine1, machine2, machine3등이라고 가정한다. 그러면 당신의 다른 디렉토리를 설정할 때에 다음의 bash 스크립트를 사용할 수 있다.

```
cd /export/linux
for x in machine2 machine3 ; do
        mkdir $x; cd $x
        (cd ../machine1; tar cf - *) | tar xvf -
done
```

그러면 아래의/export들은

- /export/linux/usr 모두에게 읽기만
- /export/liunx/machine1 단지 머신1에게 루트와 읽기 쓰기만
- /export/liunx/machine2 단지 머신2에게 루트와 읽기 쓰기만
- /export/liunx/machine3 단지 머신3에게 루트와 읽기 쓰기만

아래에서 [3]

```
# This file is /etc/export
# for remote linux X terminals by Buci
# this line is only once
/export/root/usr             -access=linuxnet
# these lines once for every host
/export/root/machine1       rw=machine1,root=machine1
/export/root/machine2       rw=machine2,root=machine2
/export/root/machine3       rw=machine3,root=machine3
```

exportfs -a 하는 것을 잊지 말아라.

5.6 tftp 서버를 설정하기

지금 tftp 서버를 설정할 때이다. 만약 당신이 안전한 tftp를 필요로 하지 않는다면 당신의 클라이언트에 관해 모든 것이 간단하여 /export 디렉토리로부터 부팅할수 있을 것이다.

만약, 안전한 ftp를 사용한다면 (단일의 실제 커널과 다른 머신들과 심볼릭 링크된것)아래 /export/linux 디렉토리 구조를 만들것이다. 또는, /export 디렉토리를 안전한 tftpd의 부트 디렉토리 가 되게 하여라. 그리고 유사하게, 다른것들을 위한 단위의 커널과 심볼릭 링크들의 원래 디렉토리 구조를 필요로하게된다. 당신의 다음을 타이핑 함으로서 이 과정을 수행할 수 있다.

```
mkdir -p /tftpboot/export/linux/machine1
cd /tftpboot/export/linux/machine1
cp /export/linux/machine1/<name of the kernel> .
```

[3]아래의 형식은 SunOS 4.1.3의 exports의 파일 규칙을 따를 것이다.

그후 다음을 타이핑하라.

```
mkdir -p /tftpboot/export/linux/machine2
cd ../machine2
ln -s ../machine2/<name of the kernel>
```

5.7 마지막 작업

마지막으로, 당신은 다음을 수정하여야 하고

```
/sbin/mount nfs_server:/export/linux/usr /usr
```

as the first line of

```
/export/linux/<machinex>/etc/rc.d/rc.S
```

여기에서 <machinex> 는 머신1, 머신2, 머신3등을 나타낸다.

제 6 절 메모리와 디스크 공간 요구사항; 속도

나는 이것을 단지 슬랙웨어 2.3에서 테스트했다.; 다른 배포판/버전에서는 다음과 다를 수 있다.

- 디스크 공간: 28MB + 6.5MB/machine
- 램 : 나는 8메가로서 X를 사용한다. 단지 4메가의 스왑만이 필요하고, 각각의 머신에서 추측하기 로는 /tmp이다. mkswap을 실행하는 것을 잊지 말라.
- 속도 : 8메가의 486 DX2/66에서 아무런 문제가 없다.

제 7 절 가능한 에러들

- 나는 이상한 에러를 발견했다.; SunOS의 /dev 서브디렉토리에서 디바이스의 내용들이 부서졌고 그래서 나는 리눅스 박스의 서브디렉토리를 마운트하여 MAKEDEV를 다시 실행시켰다. (이유는 리눅스의 nfs와 SunOS의 nfs가 다르기 때문이다.; 모두 Major와 Minor의 32비트의 디바이스 수 를 사용하지만, SunOS는 Major에 관하여 14비트가 확장된 필드를 사용하고, Minor 디바이스의 수로서 18비트의 확장된 필드를 사용한다.
- 디스크가 없는 리눅스가 부팅 되었을 때, tftp 서버에 관한 라우팅 테이블 내에서 단지 하나의 라우터만이 있고 그래서 당신은 라우팅 테이블을 정확하게 설치하여야만 한다. 당신은 여기에서 두 가지 선태권이있다.
  - 손으로 각 머신에 관한 rc.S파일을 설정하기
  - bootp 클라이언트 패키지를 사용하고 일반적인 설치 스크립트를 쓰기

제 8 절 에러들과 가능한 이 문서들의 확장

- 관련된 문서의 올바른 인용
- SunOS는 BSD기반이다. 서버 설정에 관계된 SVR4 (Solaris등)를 포함시켜야한다.
- 비록 리눅스가 매우 SunOS와 bootp/tftp 서버로서 유사하지만, 서버에 기반한 리눅스의 예는 매우 유용하다.

- 현재 etherboot 패키지에 관한 이 문서의 업데이트
- 커널 버전 1.2.13의 nfs 루트 패키지와 새로운 nfs-root 패치를 적용한 1.3.x 커널 사이에 차이점 을 보아라.
- wd8013 보다는 다른 카드를 사용할 필요가 있다.
- bootpc에 관한 설정 정보를 포함하여, 리눅스에 관한 bootp 클라이언트는 올바른 라우팅 테이블을 설치해야한다.
- Typos 와 다른 에러들: `buci@math.klte.hu` 에게 보내달라. 감사한다.