리눅스에서 PLIP 설치하기 MINI-HOWTO

Andrea Controzzi, controzz@cli.di.unipi.it v2.1, 1998년 3월 12일

김명운, kmw@physics3.sogang.ac.kr 1998년 12월 28일

이 하우투는 여러분에게 병렬라인 인터페이스 프로토콜을 만들고 사용할 수 있는 방법을 알려줄 것이다.

목 차

이 하우투는 새로 나온 것이다. 여기에는 많은 변화들과 여러 가지 내용이 보충되었지만 , 여전히 문법이나 철자에 관한(영어가 모국어가 아닌 관계로) 오류가 있거나 조금 잘못되거나 오래된 정보가 있을 수 있다. 오류가 있다면 모두에게 올바른 정보를 줄 수 있게 나에게 알려주기 바란다.

이 하우투에서 가장 큰 변화들은 다음과 같다:

- SGML 형식
- 가장 많이 질문된 "윈95와 PLIP"에 관한 일반적인 대답 (대답이 부정적이어서, 미안하다.)
- "자주 논의되는 질문과 답변란"이 양과 질적으로 크게 확장 (셀 수 없는 사용자들에게 감사한다 : 그들중 많은 사람들이 내가 약속한 데로 자신들의 의견을 볼수있을것이다.)
- 빠른 PLIP 셋업하기라는 파일이 소용없고 진부한 관계로 삭제
- 도스와 리눅스간의 PLIP 연결에 관한 부분이 향상

어떤 질문이나, 에러수정, 의견 또는 제안이 있다면, 나의 이메일 주소는: controzz@cli.di.unipi.it 이다. 도움을 원한다면 언제나 메일을 보내도 좋다. 때때로 즉시 대답할 수는 없을 것이다. 그러나 대답을 할 것이다. 만약 2주가 지나도록 나에게서 아무 내용도 받지 못했다면, 이것은 우리의 메일 시스템에 문제가 생긴 것이다: 나에게 메일을 다시 보낸는것을 주저하지 말아라. 내가 당신을 돕는 것이 나의 의무이다. 나는 내가 당신의 문제를 풀 수 있다고 보장할 수는 없지만, 시도는 할 것이다.

여러 번 메일을 보낸 후에도 답장을 받지 못한다면, 당신의 수신주소를 확인해보기 바란다. 여러 번 내 대답이 배달 문제 때문에, 대부분 수신인 주소가 root@myhost로 되어있기 때문에 발송되지 못했다.

메일을 보내기 전에 "자주 논의되는 질문과 답변들"부분을 반드시 읽고, 당신이 찾고자하는 답변이 포함되어있다면 이 보다 더 좋은 답변은 없다. 도스나 윈도우즈 95와 PLIP의 관계에 관한 질문은 각 장을 담당한 저자들에게 메일을 보내라. 난 도와줄 수가 없다.

무엇보다도, 많은 기술적인 정보들은 Terry Dawson이 쓴 net-2-HOWTO로 부터왔다. 이 미니 하우투는 net-2-HOWTO의 모든 내용을 포함하거나 대체할 수 있는 내용을 포함하고 있지는 않다 : 내 목적은 당신에게 빨리 PLIP를 영구적으로 설치할 수 있게 하는 것, 단지 그것뿐이다. 다른 대부분의 정보들은 내 개인적인 경험과 나에게 의견과 정보를 제공하는 많은 사용자들의 도움으로 이루어져있다.

네트웍과 내가 바꾸기를 제안한 설정파일들에 관한 일반적인 정보들은 net-2-HOWTO나 다른 문서들을 읽어보아라.

제 1 절 소개: PLIP는 무엇이고 왜 내가 사용해야만 하는가?

여러 호스트들을 연결을 만드는 데에는 여러 가지 방법이 있다. SLIP와 같이 PLIP는 병렬포트를 사용한다는 것을 제외하고, 두 머신간의 지역적인 접속을 허용한다.

병렬포트들은 1초에 한 비트 이상을 전송하며, 이것은 시리얼 인터페이스보다 더 빠른 속도를 낼 수 있다는 것을 의미한다.

얻어질 수 있는 속도은 전체적으로 당신의 하드웨어 (CPU와 병렬 포트)와 시스템 부하에 의존적이며, 대부분 5 Kb/sec에서 심지어 40 Kb/sec까지의 속도을 낸다.

PLIP 인터페이스는 NFS와 같은 tcp/ip 연결을 필요로 하는 형태들을 사용하기에 충분히 빠르다. 그래서, 당신은 리눅스에 관한 모든 것을 갖춘 컴퓨터 한대와 단지 최적의 시스템을 가진 다른 한대를 이용하여, 어디서나 주머신으로부터 나머지 자원들을 모두 마운트할수있다.

하나의 병렬포트만을 가지고있는 대부분의 사용자들의 불이익은 프린트와 PLIP를 동시에 사용할 수 없다는 것이다. 심지어 두개의 병렬 포트를 가지고 있어도 커널의 모듈을 사용하지 않고 프린트와 PLIP를 사용하는 것은 불가능하다.

만약 당신이 두개나 그 이상의 병렬포트를 가지고있다면 이 하우투에서 찾을 수 있는 패치를 가하여 불이익을 해소할 수 있다. 마지막으로, 도스와 리눅스간의 PLIP를 설정하는 좋은 방법을 제공할 수 있다.

나는 충분히 압력을 받고 있지는 않다: 어느 누구도 아직까지 리눅스와 윈도우즈 95와의 성공적인 연결에 관한 내용을 보고한 적이 없다.

제 2 절 PLIP를 사용하기 위한 하드웨어 요구사항

PLIP 인터페이스를 사용하기 위한 하드웨어 요구사항은 분명히 두 머신과 케이블에서 남는 병렬포트가 있어야한다. 만약 당신이 바이오스를 설정할 수 있다면, 적어도 "양방향"모드로 그러나 가능하다면 ECP나 EPP 모드로 설정하는 것이다.

케이블에 관하여, 이 내용은 커널 2.0.33의 소스 안의 plip.c에 쓰여진 내용이다.

```
사용되는 케이블은 공장 표준화 병렬 널 케이블을 사용한다. -- 여러 곳에서
"LapLink"라는 케이블을 판다. 만약 스스로 만들려고 한다면
12-conductor 케이블이 필요할 것이다.
선은:
  SLCTIN      17 - 17
  GROUND      25 - 25
  D0->ERROR   2 - 15          15 - 2
  D1->SLCT    3 - 13          13 - 3
  D2->PAPOUT  4 - 12          12 - 4
  D3->ACK     5 - 10          10 - 5
  D4->BUSY    6 - 11          11 - 6
다른 핀들은 연결하지 말아라. 그 핀들은
  D5,D6,D7인 7,8,9번 핀과
  STROBE인 1번핀, FEED인 14번핀, INIT인 16번 핀과
  여분의 접지들인 18,19,20,21,22,23,24이다.
```

그러나 나는 케이블에 관한 더 많은 정보를 얻기를 원한다면, /usr/src/linux/drivers/net/README1.PLIP와 README2.PLIP를 읽는 것을 적극 추천한다.

내 개인적인 생각으로 당신 스스로 병렬 케이블을 만드는 것은 피하기 바란다. 스스로 만든 케이블은 매우 조금의 돈은 절약할 수 있지만, 많은 골칫거리를 낳을 것이다. 만약 당신이 스스로 병렬 케이블을 만들려고 한다면, 당신 스스로 해야 한다는 것을 기억하고 정확히 plip.c에 쓰여진 데로 그러나 나는 이 부분에 대해 어떠한 경고도 줄 수 없다.

케이블 길이에 관한 마지막 말: 긴 케이블 (10 피트 또는 3미터이상)은 라디오 간섭 때문에 문제를 야기한다. 만약 당신이 긴 케이블을 원한다면, 단단하고 질 좋은 케이블을 사용하라. 그러나 그리 긴 케이블을 추천하지는 않는다. : 내가 생각할 때 가장 긴 케이블의 길이는 15미터 (50피트) 정도이다.

아무튼, 누군가가 30미터 (100피트)정도 되는 케이블이 잘 작동한다고 나에게 메일을 보내왔다. 만약 누군가가 사무실과 자기 집을 (200미터거리) PLIP로 연결하기를 원한다면, 또한 그만큼의 여분의 돈을 가지고 있다면 시도할 수 있다. 그것은 자신의 맘이다.

제 3 절 커널을 재 설정하기

당신이 벌써 커널을 재 설정하고 컴파일을 할 수 있다고 가정한다. 다른 한편으로 당신은 좋은 문서들 (kernel-howto 또는 다른 참고 자료들)을 얻어야한다. 커널을 만드는 사람들에 의해서 작업이 매우 쉽게 된 것을 감사한다. 지난 커널을 재 컴파일하는것도 일반사람들에게 쉬운 작업이므로 할 수 있다. 아무튼, 완전성을 기하기 위해 여기 당신이 해야할일들을 요약한다.

주의: 나는 당신이 2.0.xx의 커널 시리즈를 사용하고 있다고 가정한다. 지금은 커널 1.2.xx의 것들은 필요성이 없다. 커널 2.1.xx에 관해서는 개발중인 관계로 어떠한 것도 아직 없다.

나는 당신이 커널 옵션을 설정하기 위해 menuconfig를 사용한다고 가정한다. 다른 툴 들도 동등하다. 나는 menuconfig 명령을 통해서 어떻게 하는지를 알려줄 것이다.:

```
#make menuconfig
```

나는 다음을 선택할 것을 강력히 추천한다.

```
Loadable module support  --->
```

그리고 다음을 설정하라.

```
[*] Enable loadable module support
```

그리고, 가능하다면 (2.0.0대의 모듈을 가지고있다면)

```
[*] Kernel daemon support (e.g. autoload of modules)
```

그리고 뒤로 간후 다음을 선택한다.

```
Networking options  --->
```

적어도 다음의 것들을 선택해야한다.

```
[*] Network firewalls
[*] TCP/IP networking
[*] IP: forwarding/gatewaying
```

그리고 뒤로간후 적어도 다음의 것들을 선택한다.

```
[*] Network device support
<M> PLIP (parallel port) support
```

만약 당신이 모듈을 사용한다면 PLIP를 모듈로서 사용할 것을 추천한다. 만약 당신이 프린터를 사용한다면, 가서

```
Character devices  --->
```

을 선택하고 모듈로서 다음을 설정한다.

```
<M> Parallel printer support
```

지금 당신은 PLIP를 위한 커널 설정을 가능하게 하였다. 당신이 컴파일이 처음이라면 이외의 다른 옵션들을 설정하여 저장하고 나온다.

마지막으로, 다음의 명령으로 컴파일 한다.

```
#make dep ; make clean
#make zlilo
```

그리고, 만약 모듈을 사용한다면

```
# make modules
# make modules_install
```

지금 당신의 시스템을 재 시작하라.

제 4 절 PLIP 인터페이스에 관한 커널의 내용들

당신이 PLIP를 적용할 수 있게 커널을 재 설정하고 컴파일한 후 당신의 시스템을 부팅했을때 만약 커널에서 PLIP를 지원한다면 또는 당신이 PLIP를 모듈로서 컴파일 했다면 또는 PLIP 모듈을 로드할때 다음과 같은 (숫자는 틀릴지라도) 내용을 볼 것이다.

```
NET3 PLIP version 2.2 gniibe@mri.co.jp
plip1: Parallel port at 0x378, using assigned IRQ 7.
```

당신의 klogd와 syslogd의 설정 내용에 따라 PLIP내용들이 당신의 시스템 log파일들로 저장될 것이다. : 만약 이 내용이 보이지 않아도 멈춘 것은 아니다. 만약 당신이 모듈로서 PLIP를 컴파일하고, lsmod를 통하여 plip 모듈이 로드되는것을 보았다면, 그것으로 충분하다.

인터페이스의 이름을 주의 깊게 보아라. 일반적으로 plip1이지만, plip0나 심지어 plip2, plip3등이 될 수 있다. 그것은 입출력 주소에 의존적이다.

제 5 절 설정파일들을 셋업하기

주의: 데비안과 같은 어떤 배포 판들은 다른 설정 파일을 가지고 있다. 만약 당신이 표준적인 설치를 한후 rc.inet* 파일들을 발견하지 못한다면, /etc/init.d의 디렉토리에서 찾아보기 바란다.

무엇보다도 당신이 바꾸려고 하는 파일들을 먼저 백업할 것을 기억하라.

```
#cp rc.inet1 rc.inet1.BACKUP
```

아마 좋은 생각이다.

지금, 만약 당신이 할당된 IP를 가지고 있지 않다면 두머신 간의 IP를 설정하여야만 한다. 나의 경우에는 내가 직접쓰고있는 표준적인 xxx.xxx.xxx.xxx형식을 사용할 것이다.

양머신의 /etc/rc.d/inet1.rc파일에 이 라인을 추가하자. (맨 끝줄에 쓴다면 더욱 좋다.):

```
/sbin/route add -net ${NETWORK} netmask ${NETMASK}
```

여기에서 NETWORK와 NETMASK는 전에 설정이 되어있어야한다. 만약 당신이 이것에 관해 어떻게 하는지 알지 못한다면, NET-2-HOWTO를 읽어보아라.

route명령을 내린 후에 다음과 같은 내용을 본다면:

```
SIOCADDRT: network unreachable
```

그러면 이 명령을 대신 사용하라.

```
/sbin/route add -net ${NETWORK} netmask ${NETMASK} dev plip1
```

여기에서, 일반적으로, 커널의 내용에의 해서 나오는 이름을 사용해야만 할 것이다. (위의 내용을 보라)

다음과 같은 경우에는 이 값들을 무시해도 상관없다:

만약 당신이 두 머신을 단일 네트웍으로 연결하려고 한다면, 각각에 200.0.0.1과 200.0.0.2의 IP주소를 각각 부여할 수 있다. 이 경우에 당신은 안전하게 NETWORK="200.0.0.0"과 NETMASK="255.255.255.0"으로 놓을 수 있다. 이것들은 빠른 PLIP설치에서 나온 내가 사용하는 IP의 예들이다. (아래를 보라)

주의: 200.0.0.1과 200.0.0.2는 단지 예제 IP이고, 나는 이 번호들이 실제 인터넷 호스트들의 IP주소이므로 사용하지 않기를 바란다.

나는 "개인적인 주소" 구역 사이에서 당신의 주소를 선택하기를 강력히 추천한다.

```
10.0.0.0        -   10.255.255.255
172.16.0.0      -   172.31.255.255
192.168.0.0     -   192.168.255.255
```

두 머신의 /etc/hosts 파일 안에 당신이 PLIP로 연결하려는 머신들의 IP를 추가하여야한다. 나의 경우에 내용은 다음과 같다.

```
200.0.0.1       one             # this is the "one" IP address
200.0.0.2       two             # this is the "two" IP address
```

여기서 one과 two는 두 호스트간의 선택한 이름들이다.

만약, 당신이 NFS를 사용하려고 한다면, 커널을 설정하는 동안 yes라고 설정하였다면, /etc/exports의 파일 안에 당신이 외부와 연결한 디렉토리의 내용을 써넣어라. 나의 경우에, /usr 디렉토리를 마운트하기위해 다음의 엔트리를 추가했다.

```
 /usr                               two (ro)
```

NFS에 관한 더 자세한 정보는 다른 특정한 문서들을 읽어보아라: NFS에 관한 문제들은 나에게 알리지 말라, 나는 도움을 줄 수가 없을 것이다.

지금 시스템을 재 부팅하자.

제 6 절 PLIP연결을 활성화시키기

마지막으로, 루트 권한으로 실행되는 이 명령들은 (물론, 케이블이 벌써 정상적으로 연결되어있다면) PLIP연결을 활성화 할 것이다.

주의: 만약 무언가 잘못된다면, 케이블과 명령 양쪽을 체크하기 바란다. 위와 같은 과정을 모두 수행했는데도 여전히 에러가 있다면, 자주 논의되는 질문과 답변란을 읽어보아라. 필요한 많은 대답들이 있다.

무엇보다도 lp 디바이스가 존재하지 않는다는 것을 확인하라:

```
# cat /proc/devices
```

당신의 lp에 관한 다음의 내용을 볼 수 없을 것이다.

```
 6 lp
```

만약 보게 된다면, lp 디바이스를 지우고 (임시적으로), 만약 PLIP를 사용한다면 나중에 lp를 사용하라. 만약 모듈이라면 rmmod의 명령을 통해서 lp 디바이스를 지울 수 있다. 대신에 커널 컴파일의 과정으로 설정했다면, lp를 모듈로서 (더 현명한 생각) 재컴파일 해야하는것이다.

다시 one과 two의 이름을 사용한 예를 사용한다. one에서 다음의 과정을 거쳐야한다.

만약 당신이 자동 마운트 데몬을 가지고있지 않다면, 당신은 PLIP를 모듈로서 컴파일한후, 마운트 하여야한다.

```
# insmod plip
```

주의: 만약 당신의 병렬포트가 7번과 다르고, 또는 입출력 주소가 0x378과 다르다면 insmod를 통하여 인식시켜야한다. 당신의 실제 irq와 입출력 주소 (도스의 MSD 명령과 비슷한, 그러나 너무 신용하지마라) 를 확인하고 다음과 같은 명령을 써넣어라.

```
# insmod plip io=0x278 irq=5
```

항상 입출력주소가 0x378, 0x278, 0x3bc인 동안에는 irq가 7또는 5이다. 이것은 하드웨어 세팅들 (오래된 보드의 점퍼나 현대 마더보드의 바이오스) 를 통하여 주소와 irq가 맞는지를 확인하는 것이 중요하다.

만약 당신이 정확히 모듈이 로드 되고 있는지를 체크하려면:

```
  # lsmod
Module:        #pages:  Used by:
plip               3           0
```

인터페이스의 이름 (plip0, plip1 등등; 더 자세한 것은 위에서 나온 커널의 내용을 확인하라) 을 확인한 후, PLIP 인터페이스를 셋업 하라.

```
# ifconfig plip1 one pointopoint two up
```

주의: 만약 당신의 병렬 포트가 7번과 다른 irq를 쓰고 있거나 입출력 주소가 0x378과 다른 입출력 주소를 사용한다면 ifconfig를 통해서 인식시켜야한다. 커널 내용에서 나온 irq와 입출력의 주소를 다음과 같이 써야한다.

```
# ifconfig plip1 irq 7
# ifconfig plip1 io_addr 0x3bc
```

입출력 주소가 0x378, 0x278또는 0x3bc인 동안에는 irq는 대부분 7또는 5이다.

지금 잘 작동하는지를 체크한다...

```
# ifconfig

 .....
 .....
 plip1     Link encap:10Mbps Ethernet  HWaddr FC:FC:C8:00:00:01
           inet addr:200.0.0.1  P-t-P:200.0.0.2  Mask:255.255.255.0
           UP POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0
           TX packets:0 errors:0 dropped:0 overruns:0
           Interrupt:7 Base address:0x378
```

two의 경로를 더한다...

```
# route add two plip1
```

그리고, 만약 two와의 NFS연결을 원한다면:

```
# rpc.portmap
# rpc.mountd
# rpc.nfsd
```

"two"의 명령어들과 같게, 그러나 단지 two대신에 one을 사용한다.

당신의 머신들중 one은 단지 PLIP 연결을 하고있고, 만약 이것이 사실이고 그 기계가 two라면, 다음과 같이 타이핑한다:

```
# route add default gw one
```

나의 경우에는, two는 단지 랩탑컴퓨터로서 one과 단지 PLIP만으로 연결되어있고, 그래서 나는 위 라인을 two에서 타이핑했다.

마지막으로 체크하기 위해서

```
# ping two
```

one으로부터 그리고

```
# ping one
```

two로부터 모든 것이 잘 작동하는 것을 볼 수 있다.

물론, 당신은 모든 명령들이 자동으로 스크립트나 부팅할때에 자동으로 실행되기를 원할 것이다. 당신은 이 명령을 실행할 수 있는 스크립트를 만들어야한다. 현재 당신은 필요로할때 루트의 권한으로 할 수 있고 또는 부팅할때에 부를 수 있게 명령을 추가 (/etc/rc.d/rc.inet2) 할 수 있다.

PLIP를 쉽게 설정하기 위해서, 당신은 plipconfig를 사용할 수 있고, 더 자세한 정보를 원한다면 man 페이지를 보기 바란다.

PLIP를 종료하기 위해서는, 당신은 단지 다음과 같이 하면 된다:

```
# ifconfig plip1 down
```

또한 route의 엔트리를 지울 수 있다. 만약 당신이 자동 마운트 데몬을 가지고 있지 않다면, 다음과 같이 모듈을 지우면 된다:

```
# rmmod plip
```

제 7 절 자주 논의되는 질문과 답변들.

이 부분은 당신의 문제를 풀 수 있게 할 것이다. 만약 당신이 어떤 질문이 있다면, 어느 때나 나에게 메일을 보내주기 바란다.

- 나는 부팅할때에? 또는 plip모듈을 로드할때) 이런 내용을 보았다.

  ```
  SIOCSIFADDR: No such device
  SIOCADDRT: Network is unreachable
  ```

  그리고 내가 위에서 쓰여진 데로 연결을 설정할 때에, 다음과 같은 에러내용을 보았다.

  ```
  SIOCSIFADDR: No such device
  SIOCSIFDSTADDR: No such device
  SIOCADDRT: Network is unreachable
  mount clntudp_create: RPC: Port Mapper failure - RPC: Unable to send
  ```

  - 어떤 이유로 커널에서 plip support가 설정되지 않았다. 이것은 아마 다음가 같은 이유 때문이다:
    * 커널을 설정하는 동안 "PLIP support?"에 yes라고 대답하지 않았다.
    * 커널을 설정중 "printer support?"에 yes라고 대답했다.
    * 당신은 모듈로서 PLIP를 컴파일 했고, 그래서 그것을 로드해야한다.
    * 당신은 잘못된 포트의 주소를 설정했다. 즉, 당신은 예를 들어 plip0 대신에 plip1을 써야한다.

- 모듈을 제외하고 두개의 병렬포트를 통해서 PLIP와 LP를 함께 설정할 방법이 있는가?

- 그렇다. 적어도 두 가지 방법이 있는데 "PLIP와 LP를 함께 만들기 위한 패치들"에서 설명되었다:
  - 커널에서 두 가지를 설정하기 위한 패치를 하여야한다.
  - PLIP와 다른 LP를 위한 병렬포트를 사용하기 위해서 커널에 다른 패치를 하여야한다.

- 나의 두 컴퓨터를 연결할 스크립트를 만들어왔다. 나는 NFS를 설정하고 연결을 확립할 수 있는 스크립트라 불리는 나의 rc.inet2 파일을 통해 자동적으로 설정하게 되어있다. 나의 "two"라는 호스트는 one의 디렉토리들을 마운트하고있다. 나는 "two"의 정확한 내용들을 /etc/fstab에 추가하였다. 만약, "one"이 다운되는 동안에 "two"를 부팅 한다면, "two"는 "mounting remote file system..."라는 내용을 몇분동안 내며 중지된다.
  - 이 현상은 "two"가 "one"의 파일시스템을 마운트하기위해 기다리기 때문이다. 그러나 만약 "one"이 다운된다면, 당신은 "two"가 기다리기를 끝낼 때까지 기다려야한다. 이 경우를 피하기 위해서 당신은:
    - 원격 파일 시스템을 마운트하기 위해서 rc.inet2의 명령의 내용을 주석 처리한다.
    - /etc/fstab에서 "two"의 내용을 삭제하고, 당신이 필요로 할 때에 마운트를 하면 된다.
    - 더 낳은 해결책은 부팅할때마다 "one"이 살아있는지를 점검하고 살아있다면 파일시스템을 마운트하는 것이다. 이것은 rc.d내에 다음과 같은 명령으로서 대체할 수 있다.

```
if ping -c 5 one ; then
  mount one:/.....
fi
```

- 연결이 설정되었으나, ping이 실패한다. 나는 다음과 같은 커널 내용을 받는다.

```
plip1: timed out (1, 89)
```

  또는 유사한 내용들
  - 이것은 "당신 쪽과의 연결이"작동되고, 당신의 머신이 신호를 보내고 반대편은 대답을 하지 않거나 고유한 irq/입출력 주소를 기다리지 않는다. 이것은 가장 큰 일반적인 문제이고, 케이블이 나쁘거나 잘못된 irq나 입출력주소를 사용하는 것이 대부분의 경우이다. 잘못된 irq는 문제의 60%이상의 원인이며, irq를 바꾸는 것은 문제를 해결하는 가장 좋은 방법이다. 여기에는 가능한 문제의 자세한 내용이 있다.
    - 케이블이 완전히 접속되지 않았거나 선이 끊어졌거나 잘못되었다. 가능하다면, 벌써 PLIP가 작동하고있는 두 리눅스 호스트들간의 것들을 체크하라. 만약 그것이 불가능하다면, 적어도 테스터 기를 가지고 케이블을 테스트하라. 케이블이 도스나 윈도우주 95에서 잘 작동할지는 보장할 수 없다.
    - 다른 쪽의 머신이 PLIP가 설정되지 않았다.
    - 당신은 노트북을 가지고 완전한 병렬포트가 아닌 것을 사용했고, 아래를 보자.
    - 당신은 정말로 싼 간단히 프린터 포트라 불리는 병렬포트를 사용해서, 보낼 수는 있으나 받을 수는 없다.
    - 당신의 병렬포트는 양-방향으로 설정되지 않았다. 바이오스에서 이것을 설정하라. EPP나 ECP와 같은 고급 병렬포트 설정도 좋다.
    - 병렬포트들은 다른 irq를 가지고있고, 그래서 다른 irq로 plip 모듈을 로드해야한다. "PLIP 연결을 활성화시키기"의 내용으로 돌아가서 다른 irq를 선택하라.
    - 어떤 다른 디바이스들은 irq (대부분 7번)을 공유하며, 아마 사운드카드일 것이다. MSD와 같은 도스 프로그램들을 신용하는 대신에 다른 irq로 plip 모듈의 로드를 시도하라.

- 나는 올바른 IRQ와 입출력 주소를 설정했으나, 여전히 작동하지 않는다. 나는 MSD 명령으로 주소를 얻었다.

- 나는 MSD가 다른 잘못된 주소를 주고 있다는 내용을 받았다. 이 프로그램을 사용해보라 : http://www.cs.caltech.edu/ huny/para13.zip.

- 나의 연결은 성공적이고, ping도 잘 작동한다. 때때로 커널로부터 다음과 같은 메시지를 받는다.

```
plip1: timed out (1, 89)
```

또는 이와 유사한 메시지들

  - 이것은 상대편이 timeout이 되기 전에 응답이 없음을 의미한다. 만약 모든 것이 동작한다면, 이 메시지를 무시하라.: 때때로 상대편이 당신보다 더 느림을 뜻하거나 오래된 하드웨어임을 나타내는 것을 의미한다. plipconfig 명령을 사용하여 PLIP를 알맞게 설정하기 바란다.

- 나는 PLIP 연결을 설정했으나 100% 데이터 손실이 있다. 나는 내 데스크탑을 노트북과 연결했다.

  - 어떤 노트북의 병렬포트들은 PLIP에 좋지 않다. 왜냐하면, 그들은 단지 프린터 포트들 즉, 그들은 단지 데이터를 전달만하고 받지는 못한다. 지금까지 그것들을 작동할수있게하는 방법을 알지 못한다. 유일한 희망은:

    * 노트북의 설정에 보면, 아마 병렬 포트를 프린터포트대신에 병렬포트로 사용하기 위한 방법이 있다. 대부분 "병렬 enhanced mode"라고 불린다.
    * plip 모드 0를 시도하라. 나는 이것을 어떻게 하는지 또는 잘 작동하거나 지난 커널에서 여전히 이용되고있는지 알지 못한다.

- PLIP를 통하여 어떤 것들이 속도에 영향을 미치나?

  - 이것은 대답하기 어려운 질문이다. 왜냐하면, 당신의 performance와 깊게 연관되는 많은 요소들이 있기 때문이다. :

    * 서로간의 CPU 속도
    * 병렬포트의 형태와 설정
    * 시스템 부하
    * PLIP를 사용하여 무엇을 하는가.

    단지 올바른 생각으로, 당신은 40Kbytes/sec의 속도를 얻을 수 있고, 적어도 low-level 이더넷 카드나 어떤 시리얼 인터페이스보다 빠르다.

- plip1에 관하여 ifconfig up과 ifconfig down을 여러 번 했을 때, 무슨 일이 일어날까?

  - 당신이 arp를 통하여 ifconfig 명령을 추가할 것을 보아라. 처음으로 부팅할때를 제외하고, 나는 필요성이 없으나 아마 누군가는 필요성이 있을 것이다.

- 나는 개인적으로 할당된 IP 주소들을 읽었고 200.0.0.1과 200.0.0.2의 영역의 범위는 아니다. 이 IP범위들을 변경할 수 있는가?

  - 그렇다. 변경할 수 있다. 그러나 나는 이 번호들이 단순성 때문에 이 IP주소들을 설정할 때는 주의를 하여야한다. net-2-howto의 내용을 인용하면 :

```
 RFC1597은 특별히 몇몇 IP주소들을 개인적인 네트웍을 위해
할당하였다. 당신이 만약 인터넷에 연결되어 있다면, 아래의 번호들을
사용하지 말아야한다. 이 예약된 주소들은 :
  10.0.0.0       -   10.255.255.255
  172.16.0.0     -   172.31.255.255
  192.168.0.0    -   192.168.255.255
```

- 소스코드들의 수정 없이 PLIP 파라미터들을 알맞게 조절하는 방법이 있습니까?

  - 그렇다, 방법이 있다. /sbin/ifconfig명령을 사용하기 바란다. 자세한 사항은 man 페이지를 보기 바란다.

- 나는 데비안 GNU/Linux를 사용하고 있는데, /etc/rc.d/rc.inet1과 2가 존재하지 않는다. plip 설정들을 어디에 써야하나?
  - 데비안 GNU/Linux에서는 rc.inet1과 2대신에 모든 명령들을 줄 수 있는 /etc/init.d/network를 편집하면 된다.
- 나는 plip로 두 호스트를 연결하는데 문제가 있다. 첫 번째는 지난 커널을 가지고있고, 두 번째로 PLIP의 버전은 여전히 1.0.x를 사용하고있다. 이것이 문제가 되는가?
  - 그렇다, 가능하다면 같은 버전의 PLIP를 사용하는 것이 훨씬 좋다. plip.c에 쓰여진 데로 실제적으로 1.0.xx대의 PLIP는 작동하지 않는다.
- PLIP는 4비트로 작동을 하고, 커널의 문서를 읽기로는 8비트 모드로 동작을 한다는데 무슨 내용입니까? 나는 그것이 모드 1이라고 생각하는데요
  - 이 미니하우투는 설정에 관한 것이고, 기술적인 정보는 /usr/src/linux/drivers/net/README*.PLIP 파일 또는 원저자와 상이하기 바란다. 내가 아는 것은 단지 이것뿐이다.: 표준적인 PLIP는 "널 프린터" 케이블을 사용하고 모드 0 (plip0등의 인터페이스 이름과 혼동하지 말 것) 로서 4비트로 작동한다. 모드1은 8비트를 사용하고, 벌써 작동하고 있으나 손으로 만들어야하며 단지 두 호스트간에만 가능하다. 나는 이런 케이블을 가진 적이 없고, 모드 1의 PLIP연결을 설정하는 것에 관하여 알지 못한다. 만약 누군가가 알고있다면, 나에게 알려달라.

제 8 절 PLIP와 LP를 함께쓰기위한 패치들

PLIP와 LP를 함께 사용하는 가장 좋은 방법은 커널 모듈을 사용하는 것이다.: 당신은 plip.o를 로드한후 프린터나 이외의 일을 할 때에 로드하지 않을 수 있다. 당신이 진정으로 PLIP와 LP를 함께 사용하려고 한다면 다음의 패치들을 따르십시오.

8.1 같은 포트에 PLIP와 LP 사용하기

어떤 이유로 당신이 PLIP와 LP를 커널에의해서 함께 사용하려고 한다면, 이 패치들을 사용하기 바란다.

당신은 아래의 코드를 수정하여야하며, 그전에 이전 파일들을 반드시 백업하기 바란다:

```
******** modifications to linux/drivers/char/lp.c ***********************
struct lp_struct lp_table[] = {
        { 0x3bc, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL,
NULL, },
/*      { 0x378, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL,
NULL, },
        { 0x278, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL,
NULL, },
*/
};
#define LP_NO 1

******** modifications to linux/drivers/net/Space.c ********************
#if defined(PLIP) || defined(CONFIG_PLIP)
    extern int plip_init(struct device *);
    static struct device plip2_dev = {
        "plip2", 0, 0, 0, 0, 0x278, 2, 0, 0, 0, NEXT_DEV, plip_init, };
    static struct device plip1_dev = {
```

```
        "plip1", 0, 0, 0, 0, 0x378, 7, 0, 0, 0, &plip2_dev, plip_init, };
/*    static struct device plip0_dev = {
        "plip0", 0, 0, 0, 0, 0x3BC, 5, 0, 0, 0, &plip1_dev, plip_init, };
*/
#   undef NEXT_DEV
#   define NEXT_DEV     (&plip1_dev)
#endif  /* PLIP */
```

물론, 이것은 표준적인 방법이 아니다: *나는 이 패치들을 받고 내가 했던 데로 썼다. 이것은 당신 스스로 해보아야함을 의미한다.* 아무튼, 당신의 가장 큰 문제는 원 파일을 복원하고 재 컴파일 하여야 한다는 것이다.

8.2 다른 포트에 PLIP와 LP 사용하기

이 패치를 활용하기 위해서는 적어도 2개의 병렬포트를 가지고 한쪽에는 PLIP를 다른 편은 LP를 사용해야한다.

1. drivers/char/lp.c에서 다름 소스의 라인을 주석 처리하라.

```
struct lp_struct lp_table[] = {
{ 0x3bc, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, },
{ 0x378, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, },
/* { 0x278, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, }, */
};
    3 -> 2
```

2. 커널 설정

```
PLIP (parallel port) support (CONFIG_PLIP) [n] y

Parallel Printer support [y] y
```

3. 부팅할때의 커널의 메시지

```
lp1 at 0x0378, using polling driver
.....
NET3 PLIP version 2.0 gniibe@mri.co.jp
plip2: Parallel port at 0x278, using assigned IRQ 5.
```

다시 내용은 8.1과 같다.

제 9 절 도스와 리눅스간의 PLIP 연결

처음 이 미니하우투가 나온 이후로 많은 사람들이 리눅스와 도스 (또는 윈도우즈) 에 관한 많은 정보를 썼다. 일반적인 재미로 이장을 더하게 됐으며 모두에게 도움이 되기를 바란다.

이 부분은 James McDuffie의 리눅스 가제트에서 발췌했다. 여기서는 리눅스와 도스컴퓨터를 윈도우즈와 트럼펫 윈속을 사용하여 설치하고 윈도우에서 X윈도우즈의 프로그램을 실행시킬 수 있는 방법을 제공할 것이다.

최신 버전은 James Vahn에 의해서 추가되었다. *jvahn@short.circuit.com* `<mailto:jvahn@short.circuit.com>` 그는 어떻게 이 연결을 설정하고 많은 문제들에 대한 해결책을

이 문제들에 관한 부분들은 내가아닌 그에게 물어보기 바란다.

9.1 도스와 리눅스 연결

나는 당신이 반대편의 리눅스 머신과 PLIP가 거의 설정되고 케이블이 올바르게 연결되어 있다고 가정한다. 그렇지 않다면, 전의 내용들을 다시 보기 바란다.

지금, 도스 쪽에서, 무엇보다도 패킷 드라이버를 설치해야한다. 그것은 이 사이트에서 찾을 수 있다.

ftp://ftp.crynwr.com/drivers/plip.zip `<ftp://ftp.crynwr.com/drivers/plip.zip>`

이 프로그램은 도스에서 실행되며 Ethernet 패킷 드라이버처럼 동작한다. 만약 윈도우즈에서 사용하기를 원한다면, 트렘펫 윈속을 사용하라. 이것은 TCP/IP 인터페이스처럼 동작한다. 다른 한편으로, 당신은 도스용 TCP/IP프로그램을 찾을수있을것이다.

지금 리눅스 컴퓨터로 돌아가서 도스 컴퓨터의 주소를 /etc/hosts에 써넣어라. 만약 당신의 도스 컴퓨터가 할당된 IP를 가지고있지 않다면 원하는 주소를 고르기 바란다. (3장의 IP에 관한 경고를 기억하라) 지금 리눅스 컴퓨터의 이름은 리눅스, 도스 컴퓨터의 이름은 도스라고 가정한다.

당신은 다음과 같이 타이핑 하여야한다.

```
ifconfig plip1 linux pointopoint dos arp up
route add dos
```

물론, 당신이 매번 부팅될때마다 리눅스 컴퓨터에서 이를 실행하기를 원한다면 /etc/rc.d/rc.inet1에 다음의 내용을 기입하라.

```
/sbin/ifconfig plip1 linux pointopoint dos arp up
/sbin/route add dos
```

이것은 인터페이스를 설정하고 경로를 추가한 것이다. 물론, 두 번째 병렬포트를 사용한다면 대신 plip2를 써야 한다.

도스/윈도우즈 컴퓨터로 되돌아가서 다음의 라인을 autoexec.bat파일에 추가하라.

```
c:\plip\plip.com 0x60
c:\tcpip\winsock\winpkt.com 0x60
```

물론, 나는 c:\plip.com 디렉토리 안에 plip.com (패킷 드라이버) 이 있다고 가정하고 c:\tcpip에 winpkt.com이 있다고 가정하여 올바른 경로가 설정되어 있다고 가정한다.

plip.com의 프로그램은 패킷 벡터를 0x60으로 설정하고 같은 벡터에서 트럼펫 윈속을 시작할 winpkt.com을 로드 한다. 만약 케이블이 lpt1이 아니라면 plip.com 프로그램에 irq와 입출력 주소를 알려주어야 한다. 또한, winpkt.com은 윈도우즈에서 패킷 드라이버를 이용하기 위해 필요하다. 여기서부터 우리는 실제적인 트럼펫 윈속에 관한 설정을 하게된다. SLIP와 PPP에 관한 것들을 선택하지 말고, 패킷드라이버의 기입란에 60을 기입하라. 그후 당신이 가지고있는 IP 주소를 기입하고, 리눅스 컴퓨터의 IP주소는 기본 게이트웨이로서 그리고 만약 당신의 컴이 IP나 ISP의 주소를 가지고 네임서버의 역활을하고 인터넷과 맞물려있다면 네임서버의 역할을 할 것이다. 설정을 끝낸 후 윈속을 재실행한다. 윈속을 시작 그룹에 포함시키면 모든 설정이 자동적으로 실행되게 된다.

만약 당신이 윈도우즈 컴퓨터를 당신의 리눅스 컴퓨터를 통하여 인터넷에 접속하려고 한다면, IP-masquerading이 필요하며 이에 대해서는 NET-2-HOWTO를 보기 바란다. 이것은 간단히 윈도우즈 컴퓨터를 당신의 리눅스 컴퓨터의 IP로 속여주는 것을 말한다.

또한 나는 윈도우즈에서 X윈도우즈 프로그램을 실행시키는 프로그램을 발견했다. 주소는 다음과 같다.

http://www.tucows.com/ `<http://www.tucows.com/>`

지정한데로 모든 설정을 하고 윈도우즈 컴퓨터에서 리눅스 컴퓨터로 telnet을 한후 display를 윈도우즈 컴퓨터 (DISPLAY=duncan:0.0) 로 한후 원하는 프로그램을 실행시켜라. 윈도우즈에서 xv를 실행시키는 것보다 더 좋은 것은 없다. 이것이 도움이 되었기를 바란다.

9.2 도스와 리눅스간의 연결경험

주의: 나는 이 문서를 James Vahn *jvahn@short.circuit.com* `<mailto:jvahn@short.circuit.com>`. 에서 받았고 이것을 하나도 변경하지 않고 썼다. 이것은 **이 부분에 관하여 어떠한 질문이라고 나보다는 그에게 능력이 더 많으므로 그에게 보내는 것이 적합하다**는 것을 의미한다. 단지 플로피만 있는 도스 컴퓨터와 리눅스 컴퓨터와의 PLIP 연결 경험은 일반적인 문제들에 관한 아주 훌륭한 예이다.

1996년 6월 11일 마지막 업데이트

나의 플로피-단지 도스 박스는 PLIP를 통하여 리눅스 컴퓨터의 두 번째 프린터 포트와 네트웍으로 연결되어 있다. 처음 리눅스의 프린터 포트는 프린터를 위한 것이고, 둘다 영구적으로 연결되어있고 도스 박스는 리눅스 쪽으로 telnet되어있다. 이것들이 내가 성취하려고 하는 것들의 요점이다.

커널에서 프린터 포트를 찾았을 때, 만약 포트를 찾았을 때 지우지 않았다면 모두를 찾을 것이다. 다른 한편으로 PLIP는 아무 것도 얻지 못할 것이다. 첫 번째 목적은 필요할 때마다 모듈로서 드라이버를 로드 한다.

<gniibe@mri.co.jp> 썼다:

나는 커널 모듈로서 PLIP/LP를 사용하기를 추천한다. 왜냐하면

- 모듈은 설정을 바꾸기에 매우 유연하다.
- 초보 사용자들에게 커널 재 컴파일은 어려운 작업이다.
- PLIP와 LP의 공존은 모듈에 의해서 쉽게 다루어 질 수 있다.

PLIP/LP를 커널 모듈로서, 당신은 한쪽 포트에는 PLIP를 그리고 다른 쪽 포트에는 LP를 인식시킬 수 있다. 여기의 예다:

```
# insmod lp.o io=0x378
# insmod plip.o io=0x278 irq=2
```

심지어 당신은 2개의 병렬 포트를 사용할 수 있다:

```
# insmod plip.o io=0x278,0x3bc irq=2,5
```

위의 예에서,

plip0는 0x278번지에 할당되고 irq는 2번이다.

plip1은 0x3bc번지에 할당되고 irq는 5번이다.

모듈을 사용하는 것은 확실히 앞으로 나아가는 방향처럼 보인다. 다음 방법은 모듈의 사용 없이 서로 다른 포트에 PLIP와 LP를 사용하기 위한 커널 패치를 어떻게 하는지를 보여준다. 만약 당신이 모듈의 개념에 친숙하지 않다면, 빨리 설정하는 방법을 발견할 것이다.

당신은 커널의 소스구조에서 두 파일을 변경할 필요를 요한다. 나는 커널 1.2.13을 사용하고 나의 시스템에 알맞게 ../linux/drivers/net/Space.c안에서 필요한 부분을 변경하였다. 당신의 포트의 irq가 맞는지를 확인하기 위해서 PLIP를 정의한 205라인 주변을 살펴보고 어떤 드라이버 (plip0, plip1, plip2) 가 사용되는지를 확인하라. 나의 경우에는 0x278 포트가 irq 5번 (카드에 점퍼를 꽂는 형식으로)을 사용하고, 그러나 Space.c에는 irq 2번을 정의했다. 나는 박스를 열거나 점퍼를 바꾸기보다는 이 소스를 변경했다. 유동적인 것은 후에 ifconfig 명령으로 잘못된 irq를 할당할 것이며 그것은 당신은 귀찮게 할 것이다. 이것은 단순성(한 문자로) 변화이다.

다음으로, 더 어려운 과정인 :

../drivers/char/lp.c의 파일 안에서 38번째 라인의 부분을 찾아보아라.

```
struct lp_struct lp_table[] = {
    { 0x3bc, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, },
    { 0x378, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, },
/*  { 0x278, 0, 0, LP_INIT_CHAR, LP_INIT_TIME, LP_INIT_WAIT, NULL, NULL, },
 * 0x278 reserved for plip1
 *
 * };
 * #define LP_NO 3
 */
};
#define LP_NO 2
```

변경은 한 포트의 주석 처리를 한 것이고, 그래서 현재 두포트만이 정의되어있다. 0x3bc 포트는 PLIP에서 작동을 안하고 - 아마 irq라인은 오래된 단일 어댑터(MDA)에서 보았을 때 irq라인이 이 포트들에서 문제를 일으킨다.

당신은 이 파일들을 변경하기 전에 반드시 백업하여라. 맞지 않는가? 지금 프린터와 net , dummy 그리고 plip를 위한 새로운 커널을 만들어라.

시스템을 설정하라. 이것은 나의 /etc/rc.d/rc.inet1 파일이다. :

```
#!/bin/bash
#
/sbin/ifconfig lo 127.0.0.1
/sbin/route add -net 127.0.0.0

/sbin/ifconfig dummy 200.0.0.1
/sbin/route add -net 200.0.0.0 netmask 255.255.255.0
/sbin/ifconfig plip1 arp 200.0.0.1 pointopoint 200.0.0.2 up
/sbin/route add 200.0.0.2
/sbin/ifconfig dummy down
```

arp는 도스와 리눅스간의 연결을 통하여 사용되고 분명히 리눅스와 리눅스간의 연결에는 사용되지 않는다.

그리고 /etc/hosts 파일 안에 다음 두 라인, 단지 두머신의 이름을 추가하면 된다:

```
200.0.0.1 console1
200.0.0.2 console2
```

도스박스는 console2이다. Andrea의 경고를 생각하고, 공식적인 이름을 정하는 규칙을 따르는 것이 더 좋다.

모든 변화들이 적용되기 위해서는 재부팅을 하고 새로운 커널이 영향을 미칠 것이다. 만약 당신이 패치를 가하는동안, 부팅순서 (또는 dmesg를 실행하는 동안) 또는 모듈이 로드 되고 있는 동안 :

```
lp0 at 0x03bc, using polling driver
lp1 at 0x0378, using polling driver
[....]
NET3 PLIP version 2.0 gniibe@mri.co.jp
plip1: Parallel port at 0x278, using assigned IRQ 5.
```

”route”명령은 다음을 보여준다:

```
Kernel routing table
Destination     Gateway         Genmask         Flags MSS    Window Use Iface
console2        *               255.255.255.255 UH    1436   0      136 plip1
loopback        *               255.0.0.0       U     1936   0      109 lo
```

그리고 "ifconfig plip1"은 다음과 같다:

```
plip1     Link encap:10Mbps Ethernet  HWaddr FC:FC:C8:00:00:01
          inet addr:200.0.0.1  P-t-P:200.0.0.2  Mask:255.255.255.0
          UP POINTOPOINT RUNNING  MTU:1500  Metric:1
          RX packets:132 errors:0 dropped:0 overruns:0
          TX packets:136 errors:0 dropped:0 overruns:0
          Interrupt:5 Base address:0x278
```

/etc/inetd.conf 파일을 보고 만약 telnet이 된다면 보아라. 당신이 /etc/hosts.allow와 /etc/hosts.deny 파일을 이용하기를 원한다면 tcpd에 관한 매뉴얼 페이지를 읽어보아라.

리눅스 쪽에서는 모든 것이 끝났고 , 지금은 도스 쪽이다. 다시, one이 존재한다면 0x3BC의 포트를 확인하라.

나는 NCSA의 telnet을 사용하고 Crynwr의 PLIP드라이버를 다음의 사이트에서 사용한다.

ftp://ftp.ncsa.uiuc.edu/Telnet/DOS/ncsa/tel2308b.zip <ftp://ftp.ncsa.uiuc.edu/Telnet/DOS/ncsa/tel2308b.zip>

ftp://ftp.crynwr.com/drivers/plip.zip <ftp://ftp.crynwr.com/drivers/plip.zip>

NCSA의 telnet 버전 2.3.08의 사용과 Crynwar의 PLIP 드라이버의 11.1버전을 사용하는지를 확인하라. Crynwar의 SUPPORT.DOC 파일을 읽어보기 바란다.

CONFIG.TEL파일이다. 대부분은 기본적이고 필요에 따라 수정할 수 있는 공간을 남겨두라. 이 머신의 두 번째 포트는 0x278에 irq는 5번으로 설정되어있다.

```
myip=200.0.0.2
netmask=255.255.255.0       # subnetting mask
hardware=packet             # network adapter board (packet driver interface)
interrupt=5                 # IRQ which adapter is set to
ioaddr=60                   # software interrupt vector driver is using
#
#[...lots unchanged...]
#
# at the end of the file, put this line:
name=console1 ; hostip=200.0.0.1 ; nameserver=1 ; gateway=1
```

(console1은 리눅스 머신의 이름이고, 어디에서나 사용할 수 있다.)

나는 두 머신사이에 12핀의 널 케이블을 만들었고, 그리고 (처음에는 잘못 만들었다.) 거기에는 아무런 문제도 없다. 표준적인 11-wire 널 케이블도 잘 작동한다. 리눅스의 plip.c 소스는 선의 연결을 보여준다. 비록 내 케이블이 17-17번이 연결되어 있지만, 더이상 사용되지 않고 만들어진 케이블도 현존하지 않는다.

```
@echo off
plip.com 0x60 5 0x278
telbin -s console1
```

위 내용은 리눅스 박스의 /dev/ttyp와 연결하도록 한 것이다. NCSA의 telnet은 8개의 가상 스크린을 제공하고 ftp 서버처럼 작동한다. PLIP 인터페이스는 매우 질이 좋고 나의 경우에는 파일 전송에 6.5K/s의 전송속도를 얻었다. 더 좋은 속도가 나오길 바란다. :-)

제 10 절 리눅스와 윈도우즈 95와의 PLIP 연결

이 부분은 비어있다. 나는 윈도우즈 95를 게임이외에는 사용하지 않고, 그래서 시도해 보지도 않았고 리눅스와의 PLIP 연결에도 관심이 없다. 이런 연결에 관한 질문들은 대부분의 답변 contest에서 이길 것이며 그래서 나는 여기에 직설적인 답변을 내놓는다.

어느 누구도 나에게 리눅스와 윈도우즈 95와의 성공적인 연결을 보고한 적이 없다. 만약 누군가가 이 연결에 성공한다면 나에게 즉시 알려주기 바란다. 수많은 PLIP 사용자가 이 소식을 기다리고있다.

제 11 절 질문? 의견? 나에게 보내달라.

어떤 질문이나 의견들에 관해서 당신은 나의 이메일 주소인 *controzz@cli.di.unipi.it* `<mailto:controzz@cli.di.unipi.it>` 을 알 것이다. 되돌림은 환영하고, 어떤 에러보고들은 매우 중요하다. 만약 당신이 질문을 보내고, 물론 당신 스스로 답변을 얻게 된다면, 다음 버전에서 자주 논의되는 질문과 답변란이 방대해질 것이다. 자주 논의되는 질문과 답변란에 존재하는 질문들은 보내주지 말기를 바란다. 나에게 물을 것이 있다면, 적어도 당신을 도울 수 있는 정보들을 보내달라. ; 커널 버전, 사용된 명령, 당신이 사용하고있는 케이블과 PLIP에 관련된 다른 시스템 메시지들.

도스 / 윈도우즈 3.1/윈도우즈 95에서 PLIP와 관련된 어떠한 질문도 보내지 말기를 바란다. 나는 당신을 도울 수가 없다. 이런 질문들은 *jvahn@short.circuit.com* `<mailto:jvahn@short.circuit.com>`**으로 보내주기 바란다. 또한 그는 나에게 도스에 대한 추가 내용을 보냈다. 다시 : 그나 나에게 윈도우즈 95와 관련된 PLIP 질문들은 소용이 없다.**

제 12 절 새로운 버전의 미니 하우투는 어디서 얻는가.

이 미니-하우투는 HOWTO 제작자에 의해서 관리되고 월 간격으로 *comp.os.linux.answers* `<news:comp.os.linux.answers>`에서 내용이 올라오고 sunsite와 그 미러링 사이트의 HOWTO 디렉토리에서 찾을 수 있다.

이 미니 하우투를 발견하는 다른 방법은 나의 홈페이지에서이다.

http://www.cli.di.unipi.it/˜controzz/intro.html `<http://www.cli.di.unipi.it/~controzz/intro.html>` (italian language)

http://www.cli.di.unipi.it/˜controzz/intro_e.html `<http://www.cli.di.unipi.it/~controzz/intro_e.html>` (english language)

제 13 절 Credits.

많은 분들에게 감사를 드린다:

- Rick Lim <ricklim@freenet.vancouver.bc.ca> PLIP와 LP를 함께 사용하기 위한 패치에 관해서
- Takeshi Okazaki <GBA03552@niftyserve.or.jp> 다른 포트에 PLIP 와 LP를 사용하기 위한 패치에 관해서
- Jim Van Zandt <jrv@vanzandt.mv.com>충고에 관해서 "tutorial" 이 하우투의 부분
- Fernando Molina <fmolina@nexo.es> irq와 입출력주소에 관한 유용한 의견들에 관해서
- James Vahn <jvahn@short.circuit.com> 리눅스와 도스의 PLIP 연결에 관한 부분의 추가에 관해서

- PLIP와 관련된 내용을 올려준 리눅스 뉴스그룹들과 나에게 메일을 보내준 모든 분들께 감사한다. 정보와 의견들에 관해서 나를 도와준 많은 사람들의 리스트는 이 미니하우투의 내용보다 더 길다.: 모두에게 감사한다.

제 14 절 저작권 내용.