리눅스 인터넷 접속하기 하우투 (ISP-Hookup-HOWTO)

Egil Kvaleberg, egilk@sn.no
v1.23, 9 December 1996

번역: 이기동, kidong@shinbiro.com

이 문서는 다이얼업 모뎀을 이용하여 인터넷 서비스(이하 ISP라 한다.) 에 접속하기 위해 리눅스를 사용하는 방법을 설명한다. 기본적인 다이얼업 과정과 IP 설정은 물론, 전자우편과 뉴스를 다루는 방법도 포함한다.

목 차

제 1 절 소개 (Introduction)

이 문서는 어떻게 다이얼업 ISP(인터넷 서비스 제공자) 이용자들이 어떻게 리눅스를 설정하고 사용하는지에 대한 질문과 답변으로 만들어졌다.

처음으로 리눅스 기반 머신에서 ISP로 연결하고자 하는 사람들의 짐을 덜기 위해서, 마주치는 대부분의 문제들을 포괄하는 한가지 시도가 이루어진 것이다. 이렇게 되면 어쩔수 없이 다른 리눅스 하우투 문서, LDP 서적들과 상당부분이 중복될 수 밖에 없다. 참고서적들은 이들 문서를 이해하기 쉽게 하고 더 나은 세부사항을 전하기 위해서 만들어진 것이다.

지금까지 존재하는 많은 문서들은 어느정도 경험이 있는 사용자들을 대상으로 한다. 그리고, 처음 경험하는 사용자들은 때때로 적절한 정보들을 구분하는 데 곤란을 겪을 것이다.

간단히, 주어진 예제들은 다음과 같은 것들을 가정할 것이다:

- User name: `dirk`
- Password: `PrettySecret`
- Internet service provider: `acme.net`
- Email server: `mail.acme.net`
- News server: `news.acme.net`
- Name server: `193.212.1.0`
- Phone number: `12345678`

우리의 `dirk`씨는 그의 머신을 `roderick`이라 부를 것이다.

어떠한 ISP를 사용하든 위의 표에 있는 모든 참고사항을 그대로 바꾸어 넣을 수 있을 것이다. 보통, 다른 ISP의 사용자인 경우 약간의 수정이 필요하다. 이 문서에서 필자는 당신이 접하게 되는 문제에 대한 정보를 알게 되기를 원한다.

1.1 이 문서의 새로운 버전 (New versions of this document)

이 문서의 새로운 버전은 정기적으로 *comp.os.linux.answers*에 게시될 것이다. 거기에는 이곳을 포함하여 정보를 담고 있는 다양한 anonymous FTP 사이트가 추가될 것이다:

`ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO`

덧붙여서, 이 문서를 아래의 리눅스 WWW 홈페이지에서 찾을 수 있을 것이다.

`http://sunsite.unc.edu/mdw/linux.html`

마지막으로, 이 문서의 가장 최신 버전 또한 다양한 포맷으로 제공된다:

`ftp://ftp.sn.no/user/egilk/ISP-Hookup-HOWTO.txt`

`ftp://ftp.sn.no/user/egilk/ISP-Hookup-HOWTO.ps.gz`

`ftp://ftp.sn.no/user/egilk/ISP-Hookup-HOWTO.tar.gz`

`http://home.sn.no/\~{}egilk/ISP-Hookup-HOWTO.html`

1.2 피드백 (Feedback)

모든 설명, 에러 리포트, 추가될 정보나 모든 종류의 비평은 이곳으로 보내주기 바란다:

egilk@sn.no

http://home.sn.no/\~{}egilk/

1.3 경고 (Disclaimer)

이 문서의 내용에 대하여 어떠한 책임도 지지 않는다. 개념의 사용, 예제, 그외 모두 독자가 위험을 감수해야 한다. 덧붙여서, 이것은 오래된 버전이므로, 그에따른 부정확과 오류의 가능성이 있다.

셋업할 수 있는 방법은 다양하지만 그중 하나만 설명할 것이다. 리눅스 세계에서는, 같은 일을 해낼 수 있는 수많은 방법들이 있다. 대안들을 담고있는 단락에서는 **대안:** 으로 표시해 두었다. 또한 FTP-참조란에는 새로운 버전이 나옴에 따라서 변화할 수 있음을 주의한다.

필자가 알고있는 한, 특별한 규정아래서 개인적인 용도로 사용하고 평가할 수 있는 프로그램들만 별도로 서술해 둔다. 대부분의 프로그램들은 GNU 규정을 따라 소스까지 모두 구할 수 있다.

1.4 저작권 정보 (Copyright information)

제 2 절 어떻게 세계와 연결할 수 있는가?

우리는 필수적인 소프트웨어 모듈(예: 슬랙웨어 N-시리즈 중의 필수적인 부분)을 설치했고, /dev/modem이 쓰일 시리얼 포트에 대해 셋업을 마쳤다고 가정한다.

기본적인 설정으로는 root 사용자만이 /dev/modem에 직접 접속 하는 것을 허용할 것이다.

ISP 쉘 계정으로 바로 접속하려면, 그리고 접속 절차를 실험하려면, minicom 프로그램을 써도 된다. 이것은 간단히 사용 할 수 있다.

2.1 기본적인 설정 (The basic configuration)

네트웍에서 사용하기 위한 이 머신의 설정은 root 사용자로 해야 한다. 더 나아가기 전에, /etc/hosts.deny 파일이 다음과 같은 줄을 담고 있는지 확인하라:

```
ALL: ALL
```

보통은 자신만 접근을 허락하도록 할 것이므로, /etc/hosts.allow에 다음과 같은 줄을 추가한다:

```
ALL: LOCAL
```

또는 이렇게도 할 수 있다:

```
ALL: 127.0.0.1
```

다음에 오는 바와 같이, 이것은 동적 IP 주소를 가지고 PPP를 통해서 연결이 되어 있다는 것에 주목한다. 고정적으로 연결할 수 있는 혜택을 받고 있다면, 약간의 차이가 있을 것이다.

동적 IP 사용자가 선택할 수 있는 이름으로, 연결된 머신에 이름을 지어주는 것은 매우 바람직한 일이다, 그 이름을 /etc/HOSTNAME에 써준다:

```
roderick
```

다음으로 할 일은 /etc/resolv.conf에 네임서버를 셋업하는 일이다:

```
search .
nameserver 193.212.1.0
```

네임서버는 반드시 숫자로 된 IP 주소로 정의되어야 한다, 그리고 ISP에서 ISP까지와는 다를 것이다. 필요하다면, 또다른 세개의 서버들을 가질 수 있다. 그것들은 쓰여진 차례대로 요청될 것이다.

somemachine.acme.net의 약자로 somemachine과 같은 이름을 쓸 수 있기를 원한다면, 첫번째 줄을 다음과 같이 대치한다:

```
search acme.net
```

/etc/hosts에 대해서는 최소한 다음과 같은 것들이 필요하다. 대부분의 사용자들은 이들을 다룰 수 있게 될 것이다:

```
127.0.0.1 localhost
0.0.0.0   roderick
```

고정된 IP 주소를 사용할때는 이것이 분명히 0.0.0.0으로 대치될 것이다.

마찬가지로, /etc/networks중 최소한 설정해야 할 것은:

```
loopback  127.0.0.0
localnet  0.0.0.0
```

그리고 당신은 외부 메일 도메인 이름을 /etc/mailname에 정해야 한다:

```
acme.net
```

ISP에서의 사용자이름과 비밀번호는 /etc/ppp/pap-secrets에 기록한다.

```
dirk * PrettySecret
```

PAP 대신 CHAP를 사용하는 ISP의 경우, 파일이름은 /etc/ppp/chap-secrets이다.

마지막으로, 기본적으로 PPP가 초기화 되기 전에 그 접속 절차가 정의되어야 한다. 이것은 /etc/ppp/chatscript에서 이루어진다:

```
TIMEOUT 5
"" ATZ
OK ATDT12345678
ABORT "NO CARRIER"
ABORT BUSY
ABORT "NO DIALTONE"
ABORT WAITING
TIMEOUT 45
CONNECT ""
TIMEOUT 5
"name:" ppp
```

구체적인 것은 다소 바뀔 수 있다. 세번째 줄에 있는 전화번호는 물론 ISP에 맞게 수정하여야 한다. 몇몇 사용자들은 모뎀 초기화인 ATZ를 사용하는 모뎀에 맞추어 바꾸어 주어야 할 필요가 있다. 마지막 줄은 name:프롬프트를 기대하고 있을 때 응답으로 ppp가 되어야 한다는 것을 나타낸다. 다른 시스템에서는 로그인 절차가 다를 수도 있다.

접속을 실제로 초기화 할때, PPP-프로토콜은 다음과 같은 명령어로 초기화 할 수 있다:

```
exec pppd connect \
     'chat -v -f /etc/ppp/chatscript' \
      -detach crtscts modem defaultroute \
      user dirk \
      /dev/modem 38400
```

우리는 지금부터 잘 지켜보아야 한다. 그리고 프로그램이 Crtl-C를 입력하여 프로그램이 종료될 때까지 머물러 있어야 한다. 접속에 관계된 어떠한 메세지든지 시스템 로그에 덧붙여질 것이다. 이를 읽으려면, 다음과 같이 하면 된다:

```
tail /var/adm/messages
```

PPP가 떠있는 한, 인터넷에 직접 접속을 할 수가 있다, 그리고 ftp, ncftp, rlogin, telnet, finger 등과 같은 프로그램들을 사용할 수 있다. 이들 프로그램들은 네트웍 패키지의 일부이다.

PPP에 관계된 더 많은 정보는 이곳에서 구할 수 있다:

/usr/lib/ppp/README.linux

/usr/lib/ppp/README.linux-chat

마지막으로, 보안에 대해 한가지 덧붙인다. /etc/inetd.conf 파일은 자신의 머신이 외부에 제공하는 모든 서비스를 나열하고 있다. 우리가 만드는 /etc/hosts.deny 파일은, 외부의 어떠한 접근도 허용하지 않는다. 외부에서의 접근을 필요로 하는 사람들에게는, /etc/hosts.allow에 명시된 곳만 접근을 허락하도록 해야 한다. 로컬에서만 허락하기 위해서는 다음과 같이 써준다:

```
ALL: LOCAL
```

man 5 hosts_access 를 참조하도록 한다.

마지막 간단한 것으로, POP-프로토콜의 이름이 관계된 부분에 혼란이 존재한다. /etc/services의 이들에 대한 정의는 다음과 같다:

```
pop2    109/tcp   pop-2         # PostOffice V.2
pop3    110/tcp   pop-3 pop # PostOffice V.3
```

대안: chatscript 대신에, 훨씬 융통성있는 dip을 사용할 수도 있다. 그러나 diald와 연결시킬수 없다는 단점이 있다.

대안: 이더넷을 통하여 영구적으로 TCP/IP 연결을 할 수 있는 운좋은 사람에게는 PPP에 대한 어떠한 것도 무시할 수 있다. 그대신, 네트웍 카드를 셋업하는데 집중하는 것이 바람직하다.

대안: PPP를 사용하지 못하는 다른 사람들도 있다. 그러나 SLIP을 대신 사용할 수 있다. 이것은 대부분 PPP와 같은 방법이 지원된다. 다른 가능성은 UUCP이다. 주로 뉴스나 전자우편을 주고받는 사람들에게는 SOUP라는 방법도 가능하다. 뒷부분의 경우에 대한 자세한 문서는 다음에서 찾을 수 있다:

`ftp://ftp.sn.no/user/bjorn/Linux-offline.tgz`

TERM 프로그램은 역시 옵션의 하나이다. *Term-HOWTO*를 참고하라.

제 3 절 어떻게 *서핑* 하는가? (How do I *surf*?)

텍스트를 가장 중요하다고 생각한다면, Lynx 웹브라우저를 사용할 수 있다. 이것은 여기서 구할 수 있다:

`ftp://sunsite.unc.edu/pub/Linux/system/Network/info-systems/lynx-2.3.bin2.tar.gz`

X-윈도우를 설치하였다면, 그래픽을 볼 수 있는 많은 브라우저들을 사용할 수도 있다. Chimera는 여기서 찾을 수 있다:

`ftp://sunsite.unc.edu/pub/Linux/system/Network/info-systems/chimera-1.65.bin.ELF.tar.gz`

`http://www.unlv.edu/chimera/`

모자익 (Mosaic):

`ftp://sunsite.unc.edu/pub/Linux/system/Network/info-systems/Mosaic-2.7b1-aout.tgz`

`ftp://ftp.NCSA.uiuc.edu/Web/Mosaic/Unix/binaries/2.6`

모질라(넷스케이프) (Mozilla (Netscape)):

`ftp://sunsite.unc.edu/pub/Linux/system/Network/info-systems/netscape-v11b3.tar.gz`

`ftp://ftp.cs.uit.no/pub/www/netscape`

(역자 주: 넷스케이프사의 공개정책의 변화에 따라 원문과 다르게 수정하였다.)

이러한 브라우저들은 많은 방법으로 항상 *새*로운 버전을 구할 수 있다.

프로그램에 대한 사용과 평가는 각각의 규정에 따라 다르다. 자세히 읽어보도록 한다.

제 4 절 어떻게 전자우편을 주고 받을 수 있는가? (How do I send and receive email?)

무엇보다도, `sendmail`이 설치되었는지를 확인하라. Sendmail은 내부와 외부로 나갈 메일을 구분하고, 외부로 나갈 메일은 전달할 수 있는 시간이 될 때까지 저장할 것이다.

Sendmail은 `/etc/sendmail.cf` 안의 설정을 기반으로 한다. ISP 사용자에게 알맞은 예제를 여기서 찾을 수 있다:

`ftp://ftp.sn.no/user/egilk/sendmail.cf` 여기에는 *procmail*을 전달자로 설정하였지만, 이것을 deliver로 바꿀 수도 있을 것이다.

This is based on *procmail* as a delivery agent, but may easily be changed to use *deliver*.

외부로 나갈 메일에 쓰일 **공식적인** 도메인 주소를 갖기 위해서 절차가 필요한데, 그에 대한 것은 `/etc/sendmail.cf`에 명시되어 있다:

```
# who I masquerade as (null for no masquerading)
DMacme.net
```

여기서는 여러분이 ISP에서와 같은 사용자 이름을 가지고 있다고 가정한다. 이제 Sendmail은 수신자에게 **직접** 보내도록 설정이 되어 있다. 지루하고 반복된 접속을 피하기 위해서 수신하는 쪽이 느리고 불규칙하게 연결된 곳인 경우, 일반적으로 ISP를 저장 버퍼로 사용하는 것이 좋다. 이것은 DS 명세에 명시될 수 있다:

```
# "Smart" relay host (may be null)
DSmail.acme.net
```

sendmail은 때때로 `sendmail.cf` 안의 탭 구분문자를 다루는데 민감하다는 것에 주의하기 바란다. 이러한 탭 문자들이 그대로 보존되는 것을 확인하고자 한다면 `vi`에디터를 사용하면 된다.

전자우편 수신은 POP3 프로토콜을 통해서 자주 수행된다. 이것은, 연결이 성립되는 때마다 초기화된다. 이를 시험하기 위한 스크립트는 다음과 같다:

```
sendmail -q
popclient -3 -v mail.acme.net -u dirk -p "PrettySecret" \
       -k -o /usr/spool/mail/dirk
```

이 스크립트는 PPP 연결이 이루어진 후에 시작된다. 이 스크립트는 단지 시험하기 위한 것이라는 것을 주의하기 바란다, 그래서 이 스크립트가 수행되는 동안 자신의 메일박스는 손대지 않고 그대로 놔둔다는 것을 알아둔다. `-k`옵션은 ISP 메일박스에 메일이 그대로 **저장**되어 있다는 것을 의미하고, 간단히 메일의 복사본을 받은 셈이 된다. 물론 셋업이 수행중이라는 것을 확신하게 되면 이 옵션을 제거할 수 있다.

패스워드가 커맨드 라인에서 보인다는 것을 주의하기 바란다. 이것은 가능한한 빨리 수정되어야 할 것이다.

안전하고 더 나아진 버전의 스크립트는 여기서 찾을 수 있다:

`ftp://ftp.sn.no/user/egilk/pop-script.tar.gz`

이 스크립트를 사용하려면 `procmail`을 설치할 필요가 있다, 그러나 어쨌든 아쉬워해야 할 것은 없다.

`ftp://sunsite.unc.edu/pub/Linux/system/Mail/mailhandlers/procmail-3.10-2.tar.gz`

Procmail은 들어오는 전자우편을 광범위한 조건으로 분류할 수 있는 간단하고 유연한 도구이다. 게다가 휴가중 메세지 보내는 것과 같은 일을 자동으로 처리해주는 기능도 가지고 있다.

Note that when we use procmail directly as in this case, the situation is somewhat different from what is described in the procmail documentation. A `.forward` is **not** required, and we also don't need a `.procmailrc`. The latter is only required if we want to sort the mail.

전자우편을 읽고 보내는 사용자 인터페이스는 Pine이나 Elm에서 볼수 있는 것과 같다.

대안: 일반적인 다이얼업 ISP 사용자들에게는 sendmail 데몬을 항상 활성화 시켜두는 것이 필요치 않다. 리소스 사용을 절약하기 위해서, sendmail이 시작되는 부분을 설명문으로 처리할 수 있다, 이것은 보통 `/etc/rc.d/rc.M` 에서 찾을 수 있다.

대안: sendmail 자리에 어떤 이들은 더 간단한 `smail`을 사용하기도 한다. 이에 대한 자세한 설명을 *Linux Network Adminstrator's Guide* 에서 찾을 수 있을 것이다.

대안: 또 깔끔한 `sendmail.cf`를 만들기 위한 m4 매크로 패키지가 있다. 설치가 간단하기 때문에, 존재하는 설정을 수정하는 것과 마찬가지이다.

대안: 또 역시 간단하지만 덜 유연한 것으로 대안이 있다. 예를 들어 Pine은 알맞게 설정하는 한, 스탠드-얼론으로 실행된다. 새로운 버전의 웹브라우저의 메일 클라이언트를 사용하는 것도 가능하다.

대안: 많은 사람들이 Emacs에 따라오는 Gnus를 전자우편, 뉴스 클라이언트로 사용하는데 열광적이다. 더 자세한 정보는 여기서 찾을 수 있다:

`http://www.ifi.uio.no/\~{}larsi/`

대안: `popclient`를 대체할만한 것으로 `pop-perl5`가 있다. 이것은 여기서 구할 수 있다:

`ftp://sunsite.unc.edu/pub/Linux/System/Mail/pop-perl5-1.1.tar.gz`

제 5 절 뉴스 (News)

5.1 어떻게 온라인 뉴스리더를 셋업하는가? (How do I set up an online newsreader?)

PPP가 떠있는 동안, 온라인으로 뉴스를 읽는것이 가능하다. 여기에는 몇가지 가능한 프로그램들이 있다. 간단한 방법으로는 `rtin` 또는 `trn`이 있다.

대부분의 경우 설정할 필요가 있는 것은 다음과 같다. (보통 한번만, ".profile" 안에):

```
export NNTPSERVER=news.acme.net
```

To get the *From*-address correct in postings, some programs *may* require:

```
export NNTP_INEWS_DOMAIN=acme.net
```

5.2 어떻게 오프라인 뉴스리더를 셋업하는가? (How do I set up an offline newsreader?)

오프라인일때 뉴스를 읽을 수 있도록 하면, 전화세를 줄일 수 있고 더 많은 융통성을 가지게 되는데, 한 종류의 로컬 뉴스-스풀을 설정해야 한다. 여기에는 몇가지 설정이 필요하다, 그리고 어느정도 디스크의 여유공간이 필요하다. 초기 셋업이 끝난 후에는, 때때로 주의를 기울여 잘 돌아가는지 실행시켜 보아야 한다.

두가지 해결책을 여기서 설명한다.

5.3 C News를 어떻게 셋업하는가? (How do I set up C News?)

여기서 설명할 해결책은 뉴스서버 **C News**와 NNTP 프로토콜에 기반을 둔다. C News는 원래 다른 종류의 설정을 목표로 하는것이다. 그러나 이것은 우리의 상황도 다룰 수 있을 만큼 유연하다. 새롭고 인기있는 **INN** 뉴스서버를 사용하기를 원할런지도 모르지만, 그것은 리소스 관점에서 더 많은 것을 필요로 한다. 어쨌든, 둘다 설치하지 **않도록** 조심한다; 둘은 쉽게 공존할 수 없다.

`news`유저로 로그인해서 뉴스의 모든 관리가 모두 해결된다는 것은 아주 결정적인 것이다. 모든 설정파일들은 `/usr/lib/news`에 위치한다. 이것들을 다루는 한가지 방법으로, `root`로 로그인하였을때 `su news; cd`라 입력하는 방법이 있다.

설정에서 가장 중요한 파일들은:

- `active` 파일은 활성화된 뉴스그룹에 대한 목록이다. 이는 addgroup 명령을 실행시킬때 마다 필요할 때 갱신된다. 예를 들어 `addgroup comp.os.linux.networking y` 라고 할 수 있다.
- `organization`은 `Organizaton:` 헤더 필드안에 들어갈 것을 원하는대로 넣어두면 된다. 우리의 경우:

  ```
  Dirk Gently's Holistic Detective Agency
  ```

- `mailname`에는 우리의 경우 `acme.net`을 써주어야 한다.
- `whoami`파일에는 당신의 사이트 이름을 `Path:`에 써준다. 여기서 설명한 셋업에서, `NewsX`을 이용하여, 그 이름이 머신을 떠나게 해서는 안된다, 그래야 그것이 유일할 때 얼마든지 원하는대로 셋팅할 수 있다. 우리의 경우는 `roderick`이라 하겠다.

- sys는 배포되는 뉴스를 가져오고 촉진하는 것을 조절한다. 우리의 경우에는 ISP에 acme.net을 Path에 추가한다는 것을 가정한다, 그리고 이것은 우리가 가지고 있는 유일한 뉴스 소스이다. 주어진 예는 우리에게 도착된 모든 것을 받아들이고, 전에 볼 수 없었던 것만 acme.net에 뉴스를 게시한다는 것을 알려준다. 그리고 게시는 원래 우리가 운영하는 사이트에만 한다. 이렇게 간단하게 된 셋업에서는 모든 그룹은 하나의 소스에서 왔다는 것을 가정한다. /all은 배포를 명시한다, 그리고 반드시 포함되어야 한다. 글자 F는 바깥으로 나가는 뉴스기사가 파일에 쌓인다는 것을 말한다.

```
ME:all/all::
acme/acme.net:all,!junk/all:FL:
```

- 외부로 나가는 뉴스가 들어갈 서브디렉토리는, 우리의 경우 다음과 같이 생성한다:

```
mkdir /var/spool/news/out.going/acme
```

- mailpaths는 moderate groups(역자주: 권한을 가진 사람만 게시할 수 있는 뉴스그룹, 포스팅할때는 그러한 권한을 가진 관리자에게 메일을 보내는 것으로 대신한다. 대표적으로 comp.os.linux.announce가 있다.)의 포스팅에 관한 것을 조절한다. 보통 이 일은 ISP가 맡는다.

C News는 매일 어느정도 유지보수할 필요가 있다. 그러나 이는 한번, news 사용자로서 crontab -e 명령을 통하여 모두 정하는 것이 가능하다. 권장하는 셋업은 다음과 같다; 이를 필요로 한다는 것이 나타나게 된다:

```
# maintain incoming and outgoing batches
10,40 *  * * * /usr/lib/newsbin/input/newsrun

# expire C News, once a day
30 0  * * * /usr/lib/newsbin/expire/doexpire

# monitor and report if needed
00 2  * * sat /usr/lib/newsbin/maint/addmissing
40 3  * * * /usr/lib/newsbin/maint/newswatch
50 3  * * * /usr/lib/newsbin/maint/newsdaily
```

Newsrun은 (한시간마다 두번씩) 기사를 내부와 외부로 옮긴다. doexpire은 오래된 기사들을 지운다.(매일 밤 00:30에), 그리고 나머지 세 명령은 다양한 관리작업과 에러 수정을 수행한다.

역시 이것들이 머신을 시작할때 깨끗이 지워지는지 분명히 해야 한다. root 사용자로, 다음과 같은 줄을 /etc/rc.d/rc.local에 추가한다.

```
su news -c /usr/lib/newsbin/maint/newsboot
```

뉴스는 아마도 **NewsX**라는 프로그램이 뉴스를 NNTP 서버에서 가져옴으로써 모이게 된다. 이 프로그램은 여기서 찾을 수 있다:

ftp://sunsite.unc.edu/pub/Linux/system/News/newsx-0.1.tar.gz Or:

ftp://ftp.sn.no/user/egilk/newsx-0.3.tar.gz

NewsX를 셋업하는 것은 무척 간단하다. 일반적으로 설치하는 순서는 다음과 같다:

```
make
su
make install
exit
```

여기서의 셋업과 함께, 여러분이 해야할 모든 것은 "addgroup" 명령을 사용하여 여러분이 읽기를 원하는 그룹을 생성하는 것이다.

기사를 모으기 위해서, news 사용자는 다음과 같은 명령을 내린다 (PPP나 그와 비슷한 접속이 이루어져 있다고 가정한다.):

```
newsrun
newsx acme news.acme.net
newsrun
```

옵션 -d는 스크린에 연속적인 출력을 행한다. 더 자세한 정보는 NewsX에 따라오는 문서를 참고하도록 한다.

NewsX는 역시 나갈 기사를 포스팅하는 것에도 주의를 기울일 것이다.

오래된 기사들을 처리하기 위해서, explist 파일이 필요하다. 예제의 설명은 우리가 무엇을 원하는지 설명하고 있다:

```
# hold onto history lines 14 days, nobody gets >120 days
/expired/           x    14   -
/bounds/            x    0-1-120 -

# retain these for 2 months
comp.sources,comp.os.linux.all      x    60   -

# noise gets thrown away fast
junk,control              x    2    -

# default:  14 days, no archive
all                 x    14   -
```

대안: 작은 뉴스-스풀에서, 뉴스그룹 control이 필요 없을 때도 있다. 유용성에 비해서 **대단한** 부하가 걸린다. 주된 점은 기사들이 취소된다는 것이고, 그리고 그룹들은 자동으로 만들어질 것이다. newsgroup을 담고있는 제어 메세지들이 뒤죽박죽 섞이는 것을 막기 위해 newgroupperm이라 불리우는 파일에 우리가 허가할 것들을 정의한다:

```
comp.os.linux  tale@uunet.com yv
all       any       nq
```

이러한 예에서, comp.os.linux 아래 있는 적당한 그룹들을 모두 생성하려면 (y), news 사용자가 통지받기를 원한다면 (v)를 써준다. 나머지 모든것들을 조용히 이루어지게 하거나 (q) 무시할 수 있다 (n). 마지막 줄은 모든 뉴스그룹을 직접 만들기를 원한다면 충분할 것이다.

대안: NewX의 대안으로써 suck이나 postit과 결합한 slurp이 있다. Slurp는 많은 뉴스 서버에 커다란 부하를 가하는 NNTP NEWNEWS 를 사용한다.

5.4 Leafnode를 어떻게 셋업하는가? (How do I set up Leafnode?)

다른 해결책으로는 통합 패키지인 leafnode를 설치하는 것이다. 이것은 모든 개인 뉴스 스풀에 필요한 모든 작업을 다룰 것이도, 설정하기도 쉽다. leafnode는 여기서 구할 수 있다:

`http://www.troll.no/freebies/leafnode.html`

C News의 경우, news라는 사용자가 모든 뉴스 유지보수를 하여야 한다.

leafnode의 홈디렉토리는 /usr/lib/leafnode에 있다. 설치하기 위해서는 다음과 같이 입력한다:

```
cd /usr/lib/leafnode
tar -xzvf leafnode-0.8.tgz
cd leafnode-0.8
make
su
make install
```

아직 root로 로그인한다면, /etc/inetd.conf의 NNTP를 제어하는 줄을 다음과 같이 바꾸어준다:

```
nntp  stream  tcp  nowait  news  /usr/sbin/tcpd /usr/local/sbin/leafnode
```

Activate it by:

```
killall -HUP inetd
```

exit를 입력함으로써 news 사용자로 돌아온다. /usr/lib/leafnode/config에서는 NNTP 서버를 정의하고 있는 줄을 바꾸어준다. 우리의 경우에는:

```
server = news.acme.net
```

Leafnode는 news 사용자로 crontab -e에 다음과 같은 명령을 첨가함으로써 자신을 관리할 것이다:

```
# expire Leafnode, once a day
0 4 * * * /usr/local/sbin/texpire
```

뉴스 교환은 다음과 같은 명령으로 news사용자를 통해서 이루어진다. (PPP가 떠있고 실행됨을 가정) The news exchange is also done as user news by the following command (assuming PPP is up and running):

```
/usr/local/sbin/fetch
```

사용자들은 뉴스를 읽기를 원하는 사용자들은 *How do I set up an online news-reader?*에 나온 방법을 사용하여야 하는데, 단 로컬 머신에서 설정할때는 제외한다, 예를 들면:

```
export NNTPSERVER=localhost
```

지금까지 나온 것이 해야 할 모든 것이다. 첫번째 fetch는 이용할 수 있는 뉴스그룹의 리스트를 전달한다. Leafnode는 사용자가 어떠한 그룹을 요청하였는지 감시하고, **다음** 번에 그 그룹이 활성화되면 받아들이게 된다.

대안: leafnode의 대신할 만한 것으로는 nntpcache가 있는데, 여기서 구할 수 있다:

ftp://ftp.suburbia.net/pub/nntpcache/nntpcache.tgz

제 6 절 어떻게 접속 절차를 자동화하는가? (How do I automate the connection procedure?)

리눅스에서 뉴스와 전자우편을 자동으로 처리하는 것을 구현하는 것은 매우 쉽다.

먼저 무엇보다 ISP로의 연결을 초기화해 주는 /usr/lib/ppp/ppp-on을 만들어야 한다. 보통, 이 파일은 간단히 다음과 같은 것을 담고 있다:

```
/usr/sbin/pppd
```

더 자세한 설정은 /etc/ppp/options에서 수행된다:

```
connect "/usr/lib/ppp/chat -v -f /etc/ppp/chatscript"
crtscts
modem
defaultroute
asyncmap 00000000
user dirk
/dev/modem 38400
```

연결을 끝내기 위해서, /usr/lib/ppp/ppp-off가 들어있는 버전을 사용하면 된다.

이 두 스크립트의 기능을 시험하고 나면, 여러 작업을 수행하는 스크립트를 작성해야 한다. 이 스크립트는 앞에서 설명한 메일들을 가져오는 일을 한다, 그리고 이것이 /home/dirk/pop에 있다는 것을 가정한다.

메일을 주고받기 위한 스크립트는 /root/mail에 생성된다:

```
#! /bin/sh
#
# exchange mail
# 10 minutes timeout:
TIMEOUT=600
DT=10

# kick sendmail:
sendmail -q &

# retrieve mail:
su dirk -c /home/dirk/pop

# wait for sendmail to terminate:
t=0
while ! mailq | grep -q "Mail queue is empty"; do
    t=$[$t+$DT]
    if [ $t -gt $TIMEOUT ] ; then
     echo "sendmail -q timeout ($TIMEOUT).."
     exit 1
    fi
    sleep $DT
done

exit 0
```

뉴스를 주고받기 위한 스크립트는 /usr/lib/news/news에 위치한다:

```
#!/bin/sh
#
# exchange news
# must be run as news:
cd /usr/lib/news

#update the outgoing batch (C News):
/usr/lib/newsbin/input/newsrun < /dev/null

#exchange news:
/usr/lib/newsbin/newsx acme news.acme.net
```

```
#and flush the incoming batch:
/usr/lib/newsbin/input/newsrun < /dev/null
```

A script to connect the various bits and pieces remains, and can be placed in `/root/news+mail`:

```
#!/bin/sh
#
# exchange news and email
# must be run as root
#
if ! /usr/lib/ppp/ppp-on; then
    exit 1
fi
trap "/usr/lib/ppp/ppp-off" 1 2 3 15

#exchange news+mail:
/root/mail &
su news -c ~news/news
wait

#disconnect..
/usr/lib/ppp/ppp-off

#update the incoming batch (C News):
su news -c /usr/lib/newsbin/input/newsrun < /dev/null &

exit 0
```

외부로 나갈 메일이나 뉴스가 있는 경우 접속하도록 확장하는 것은 매우 쉽다. 이것을 `/root/news+mail`라 하자, 그리고 외부로 나갈 뉴스-스풀의 이름이 알맞게 갱신되어야 한다는 것을 새겨둔다:

```
#!/bin/sh
#
# exchange news and email, only if outgoing news or mail
# (C News spool)
if [ -s /var/spool/news/out.going/acme/togo ] ||
    ! ( mailq | grep -q "Mail queue is empty"); then
     /root/news+mail
fi
```

이제 남은 것은 언제 내보내야 할지 정하는 일 뿐이다. 이것은 root로 `crontab -e` 명령어로 수행된다. 우리는 항상 뉴스와 메일을 아침 07:00 에 주고받고, 외부로 나가는 메일과 뉴스를 매 네시간 마다 내보낸다고 가정하자.

```
00 7      * * *     /root/news+mail
00 11,15,19,23 * * *     /root/news+mail.cond
```

각각의 부분들을 같이 연결하기 전에 충분히 시험해볼 것을 권한다. 나중에 다른 일들을 첨가할 수도 있다. 예를 들어 시간을 맞춘다던가 (ntpdate 를 이용), 로컬로 운영되는 WWW의 자동적인 갱신(미러링), 그리고 FTP 파일을 ISP로 업하는 것(make와 ftp를 이용) 등이 있다.

대안: 취향에 따라서, 프로세스가 엉망이 될 수도 있다. PPP 링크가 초기화 될때마다, `/etc/ppp/ip-up` 스크립트가 시작된다. 여기에 전자우편과 뉴스를 주고받기 위해 필요한 어떠한 magic이든 추가시킬 수 있다. 자세한 사항은 `man pppd`를 참조하기 바란다.

대안: 네트웍 트래픽 요구가 감지되었을 때마다 자동으로 PPP 접속을 할 수도 있다. 여기에는 많은 정밀한 방법들이 있지만, 자주 접속이 이루어지는 것을 막기 위해서 좋은 설정에 크게 의존한다. 더 많은 정보는 여기서 구할 수 있다:

http://www.cs.toronto.edu/\~{}schenk/diald.html

diald 유틸리티는 여기서 구할 수 있다:

ftp://sunsite.unc.edu/pub/Linux/system/Network/serial/diald-0.13.tar.gz

역시 그곳에서 다른 다양한 PPP 접속 방법을 발견하게 될 것이다.

제 7 절 마지막으로 (Final words)

7.1 알아야 할 또다른 것들은? (Other things I should know about?)

- 시스템의 많은 에러 메세지들은 보통 내부적인 전자우편에서 발생된다. 이것들을 읽을수 있도록 하기 위해서, /etc/aliases를 만들어야 한다. 위의 파일을 바꿀때마다 newaliases 명령을 실행해야 한다는 것을 기억하라. 대부분의 경우를 포괄하는 한가지 예이다:

```
PostMaster: root
ftp: root
news: root
usenet: root
FaxMaster: root
fax: root
WebMaster: root
MAILER.DAEMON: root
```

- 많은 리눅스용 프로그램들은 **Sunsite**에서 찾을 수 있다, 그렇지만 그곳은 매우 붐빈다. 다행히도 많은 미러들이 있어서, 이들은 항상 ftp://sunsite.unc.edu/pub/Linux/..을 참조한다. 자신에게 가까운 미러를 사용하도록 해야 한다. 예로 ftp://ftp.nvg.unit.no/pub/linux/sunsite/..
- Yarn으로 부터 옮겨야 할 일이 생겼을 때, 아래에서 구할 수 있는 yarn2mf를 사용하여 표준 폴더를 이들로 바꿀 수가 있다:

ftp://ftp.sn.no/user/egilk/yarn2mf.zip

제 8 절 ISP 관련 정보 (ISP specific information)

더 많은 특정한 ISP에 관계된 정보는 다양한 곳에서 구할 수 있다:

Demon Internet (demon.co.uk)

ftp://ftp.demon.co.uk/pub/unix/linux/Demon/slack3.0.help.tgz

Easynet TBA

Netcom

http://www.netcom.com/bin/webtech/NetCruiser/Operating_Systems/Linux/linux.cfg.html

PowerTech, Schibstednett, Telenor Online

http://home.sn.no/\~{}egilk/no-isp.html

Primenet TBA

Stanford

http://www-leland.stanford.edu/\~{}wkn/Linux/network/network.html

여기에 나오지 않은 ISP에 대한 정보를 알고 있다면, 연락을 부탁드린다.

8.1 어떻게 더 배울수가 있는가? (How do I learn more?)

*Linux Network Adminstrator's Guide*라 불리우는 *리눅스 문서 프로젝트* (Linux Documentation Project)책은 정말 TCP/IP와 인터넷을 포함한 모든것을 셋업, 유지보수하는 누구에게나 필수적인 책이다:

ftp://sunsite.unc.edu/pub/Linux/docs/linux-doc-project/network-guide/nag-1.0.ascii.tar.gz

각각의 소프트웨어 패키지에 따라오는 문서들은 필요로 하는 자세한 정보를 제공할 것이다. 매뉴얼 페이지는 가장 먼저 보아야 할 것이다. 예를들어 다음을 한번 시도해보라:

```
man pppd
```

여러 프로그램에 대한 또다른 문서들을 /usr/doc에서 찾을 수 있을 것이다. 항상 잘 구조가 짜여져 있는 것은 아니지만...

다음과 같은 HOWTO들은 깊이 관련된 것들이다:

- **Installation-HOWTO**로는 기초를 쌓을 수 있다.
- **NET-2-HOWTO**에는 NET 코드에 대한 설치와 셋업에 관하여 대단히 자세하게 설명하고 있다. 여기에 있는 대부분은 표준 리눅스 배포본을 사용하는 경우에는 (예를 들어 Slackware, RedHat, Debian) 이미 설정되어 있다. 그러나 설치와 문제해결에 관한 많은 부분들은 상당히 가치있는 것들이다.
- **Mail-HOWTO**는 많은 도구들을 어떻게 설정하는지 설명하고 있다. 마찬가지로, 여기에 있는 대부분은 표준 리눅스 배포본을 사용하는 경우 이미 설정되어 있다.
- **News-HOWTO**는 (통상적인) 뉴스 스풀을 셋업하는데 쓰인다.
- **Tiny-News**는 뉴스를 수집하는 "slurp"에 대한 대안을 포함한다.
- **PPP-HOWTO**에는 PPP 접속을 셋업하면서 겪게 되는 여러 문제에 관한 자세한 설명이 있다.
- **Serial-HOWTO**는 시리얼 포트를 셋업하는데 알아야 할 필요가 있는 모든 것들을 담고 있다.
- **Mail-Queue** tells you how to send up *sendmail* to always queue remote mail but deliver local mail at once.

8.2 감사드릴 분들 (Thanks to)

여기에 있는 정보들은 많은 소스로부터 모아진 것이다. 직접, 간접적으로 공헌한 다음의 많은 분들께 감사드린다:

```
Adam Holt <holt@graphics.lcs.mit.edu>
Arne Coucheron <arneco@oslonett.no>
Arne Riiber <riiber@oslonett.no>
Arnt Gulbrandsen <agulbra@troll.no>
Bjorn Steensrud <bjornst@powertech.no>
Gisle Hannemyr <gisle@a.sn.no>
Hans Amund Rosbach <haro@sesam.dnv.no>
Hans Peter Verne <hpv@ulrik.uio.no>
Harald T Alvestrand <Harald.T.Alvestrand@uninett.no>
```

```
Harald Terkelsen  <Harald.Terkelsen@adm.hioslo.no>
Haavard Engum <hobbes@interlink.no>
Johan S. Seland <johanss@sn.no>
John Phillips <john@linux.demon.co.uk>
Jorn Lokoy <jorn@oslonett.no>
Kenneth Tjostheim <kenneth.tjostheim@asplanviak.no>
Kjell M. Myksvoll <kjell.myksvoll@fou.telenor.no>
Kjetil T. Homme <kjetilho@math.uio.no>
Michael Meissner <meissner@cygnus.com>
N J Bailey <N.J.Bailey@leeds.ac.uk>
Nicolai Langfeldt <janl@math.uio.no>
Ove Ruben R Olsen <Ove.R.Olsen@ub.uib.no>
R. Bardarson <ronb@powernet.net>
Steinar Fremme <steinar@fremme.no>
Sverre H. Huseby <sverrehu@ifi.uio.no>
Trond Eivind Glomsrod <teg@stud.imf.unit.no>
Tommy Larsen <tommy@mix.hive.no>
```