Using KORNET ADSL Service under Linux

이재광 (leesop@nownuri.net) 1999/12/7

본 글은 리눅스에서 한국통신 코넷 ADSL을 사용하는 방법을 서술하고 있습니다. 최선을 다해서 작성하였으나, 저자들은 이 글을 '있는 그대로만' 제공할 뿐이며, 내용의 정확성이나 동작 여부에 대해 저자는 어떠한 종류의 보증을 하지 않습니다. 일부 가입자의 경우 서버 설정이나 장비 등에 따라 다른 방법을 모색해야 하는 경우가 있음을 미리 밝힙니다. 다만, 잘못된 것이나 다른 자료가 있으면 연락 주시면 새로운 버전에 보완해서 수록하고, 꼭 도움을 받은 분들을 추후의 문서에 수록하겠습니다. 특히 커널 패치 방법을 개발해주신 채윤기님(ezen@hitel.net)과 Jamal Hadi Salim씨 (hadi@cyberus.ca)에게 특별한 감사의 말씀을 드립니다. 이 외에 이 글을 쓰는데 도움을 준 많은 열정적인 리눅서 분들에게 감사드립니다. * 저작/배포권 알림: 이 글은 누구나 열람/자유롭게 배포하실 수 있습니다.

목 록

1 개요

한국통신 코넷 ADSL은 단순한 인터넷 연결 방식이 아니라, PPPoE 라는 방식을 이용해서 인터넷에 접속하도록 되어 있다.

따라서 이를 구현하는 클라이언트가 없이는 인터넷에 접속할 수가 없게 설계되어 있다.

그러나, 현재 대다수의 배포중인 PPPoE 클라이언트는 윈도우즈 기반으로 나와 있으며, 매킨토시 버전도 일부 있으나, 리눅스는 상대적으로 지원에서 후위로 밀려있는 상황이어서 많은 리눅스 이용자들은 이 환경에서 고속 통신의 혜택을 입는데 어려움이 많다.

이 문서에서는 여러 리눅서들의 경험에 비추어 한국통신 ADSL 시스템에 내가 가진 리눅스 박스를 연결시키는 방법을 제시하고, 간단한 원리와 문제 해결 방법을 제시하고자 한다.

이 문서에서는 기본적으로 외장형 ADSL 모뎀을 이용하는 것을 전제로 하고 있으며 (유감스럽게도 내장형 ADSL 모뎀에 대해서는 이 글이 작성되는 시점에서 전세계 누구도 대안을 제시하지 못하고 있다. 하드웨어 벤더들의 실질적인 지원이 필요 하다.) ADSL 설치에 관계된 하드웨어 세팅은 모두 맞춰져 있는 상황이라고 가정한다.

2 설치 방법

* 설치에 들어가기 전에...

레드햇 리눅스 5.2 이상 (Glibc 2.x)

이 글은 x86 머신를 기준으로 했다. 알파나 스팍은 관련 링크에서 소스를 받아다가 맞게 컴파일해 시도해보길 바란다.

커널 2.0 이상

되도록 2.2.x 를 추천한다.

pppd 2.3.7 이상

레드햇 6.1이라면 2.3.10-3 이상을 사용하라.

멀쩡한 랜카드 + 드라이버가 잡혀 있어야 한다. (너무나 당연..)

그리고 진짜 접속을 위해서는 윈도우즈에서 ADSL 접속시에 필요했던 “로그인 창에 쓰는 아이디”와 “암호”를 반드시 알아야 한다.

지역에 따라서 코넷 아이디가 leesop 일때 로그인창에 쓰는 아이디란에는 leesop@local 이라고 적거나 leesop@ppp 라고 적는 경우도 있다고 들었다. 반드시 확인하고 시작하라.

2.1 커널 패치를 하지 않는 방법 (enternet 100 프로그램 사용)

가장 간단하고, 여러가지 커널 버전에 독립적으로 이용할 수 있으며, 보다 안정적이나 이를 이용할 수 없는 환경에 있는 사람들이 있으므로, 주의하기 바란다.

2.1.1 PPPoE 프로그램을 다운로드 한다.

가장 신뢰성 있는 PPPoE 클라이언트는 윈도우즈 등의 다양한 플렛폼에서 제품을 내놓고 있는 NTS 사의 Enternet 100 의 Linux 버전이다. 이 Enternet 이라는 제품은 원래 일반이용자들에게 판매되는 소프트웨어가 아니고, 특수한 PPPoE 서버가 반드시 필요한 제품이기 때문에 기업이나 ISP를 대상으로 OEM 납품만 한다. 따라서 이를 구하려면 이 제품이 OEM 으로 들어간 ISP의 고객 지원/다운 로드 사이트에서 가져오는 방법 밖에는 없다.

캐나다의 Sympatico 사에서 이 제품을 라이센스해서 자사 가입자들에게 Access Manager 라는 이름으로 배포하고 있다. 몇몇 리눅서들이 자신의 홈페이지에서 관련 프로그램셋을 제공하고 있다.

우리는 여기서 다운로드 한다. <http://sidhe.folkwolf.net/~chris/pppoe-howto/enternet100.bin.tar.gz> 레드햇 5.2용의 바이너리 파일이며, 레드햇 6.0 이상에서도 잘 동작한다.

이 홈페이지 아래에 enternet100 의 소스도 있으나 라이브러리 등이 더 있어야 하며 테스트해본 결과 컴파일이 한번에 잘 되지는 않았고, 괜히 초보의 경우 머리만 복잡해지는 단점이 발생했다.

나중에 성공하고 난 뒤에 할일이 없거나, 여러가지 이유로 실행이 되지 않을 경우에만 받아서 해보길 권한다.

2.1.2 적당한 위치에 압축을 푼다.

이 파일을 아무 디렉토리에나 푼다. 다음의 두 파일이 나올 것이다.

```
pppoe
pty-redir
```

다른 파일이 하나도 나오지 않지만 당황하지 마라.

2.1.3 1차 테스트를 해본다.

이 1단계 테스트 과정에서 실패한다면 이 방법으로는 현재 환경에서 접속이 되지 않음을 의미한다. 2-2 방법으로 전환하거나, ADSL 담당자와 문의하기 바란다.

일단 랜카드를 올린다.

```
ifconfig eth0 up
```

파라미터 등은 중요하지 않다.

linuxconf 등에서 IP는 192.168.0.10 식으로 아무렇게나 설정하고, 게이트웨이는 설정하지 마라. 호스트 네임 등은 전혀 중요하지 않다. 바꿔봐야 도움이 안된다.

아래는 이를 올렸을때의 필자의 상황이다. 참고하라.

필자는 IP 마스커레이딩을 시도하기 위해 랜카드를 두개 설치했다. 이중에 eth0 이 ADSL 모뎀과 연결되어 있다.

```
[leesop@www pppoe]$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:C0:26:11:8D:BD
          inet addr:192.168.0.10  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:10 Base address:0xe400

eth1      Link encap:Ethernet  HWaddr 00:00:E8:ED:20:A0
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:9 Base address:0xe000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
```

```
        UP LOOPBACK RUNNING  MTU:3924  Metric:1
        RX packets:32 errors:0 dropped:0 overruns:0 frame:0
        TX packets:32 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
```

랜카드를 올린 다음, 아까의 pppoe 프로그램을 구동한다.

```
[leesop@www pppoe]$ ./pppoe
```

이를 입력하고 몇초간을 기다려 본다. 이때 모뎀으로 일반 PPP 접속시에 나오는 쓰레기문자가 일정 간격으로 반복해서 출력되면 성공이다.

그렇지 않다면 지난번 접속이 비정상 종료되어 있어 일시적으로 회선이 잠가져 있는 것이거나, 아니면 센터쪽의 머신이 PPPoE 를 제대로 지원하지 않는 것이다. 이때는 뒤의 2-2의 커널 패치 방법을 시도해 보거나 센터와 협상을 해야 한다.

윈도우즈 머신이 있다면 테스트전에 미리 한번 연결해 보고서 시도해 보기를 권한다.

2.1.4 접속 스크립트를 만든다.

앞의 두 파일만 덩그만하게 나와 있는 것에 당혹해 하는 분들이 많았을 것이다.

필자가 이를 실행하기 위한 간단한 구동 스크립트를 제공한다.

```
[leesop@www pppoe]$ ls -al
total 243
drwxr-xr-x   2 leesop   users        1024 Nov 30  1999 .
drwxr-xr-x   3 leesop   users        1024 Nov 28 18:59 ..
-rwxr-xr-x   1 leesop   users      238004 Jun 25 06:18 pppoe
-rwxr-xr-x   1 leesop   users        5320 Nov 30  1999 pty-redir
-rwxr-xr-x   1 leesop   users         228 Nov 30  1999 start

[leesop@www pppoe]$ cat start
#!/bin/sh -x
USER=leesop@local
pty=`./pty-redir ./pppoe -I eth0`
if [ "X$pty" = "X" ]; then
        echo >&2 "cannot start pppoe"
        exit 1
fi
/usr/sbin/pppd $pty noipdefault defaultroute passive persist name $USER lcp-echo-interval 30
```

앞의 USER 부분의 아이디를 윈도우즈에서 로그인 창에 쓰는 것으로 교체하라. 앞으로 접속을 위해서는 이 디렉토리로 와서 ./start 라고 치면 접속이 될 것이다. 그러나, 당장 급하게 아직 시도하지는 마라. 설정할게 더 있다.

2.1.5 pap-secrets 를 설정한다.

앞에 아이디는 넣는 곳이 있는데, 암호를 입력하는 란이 없다고 이상하게 생각할 수도 있다. 걱정하지 마라. 암호는 여기에 넣는다.

/etc/ppp/pap-secrets 파일을 연다.

이 파일의 끝에 다음과 같이 적는다.

```
leesop@local * 12345
```

앞의 # 으로 시작하는 문장들은 주석이다. ppp 프로그램은 이걸 읽으려고 시도조차 하지 않는다. 따라서 이걸 제거하고 안하고는 아무런 의미가 없다. 혹시 안된다고 이걸 지워보고 하는 것은 시간 낭비이다.

구성은 보시는대로 앞에는 앞서 스크립트의 USER 항목에 적었던 로그인창에 적는 아이디와 별표 그리고 암호이다. (예로 12345라고 적었다. 정말로 필자가 암호를 12345로 설정하고 사용중이라고 믿지는 말기 바란다. ^_^) 세가지는 모두 공백이나 탭 등으로 띄워놓기만 하면 된다.

2.1.6 설정은 완료, 2차 테스트를 해보자.

터미널 창을 하나 더 연다.

새 창에서 tail -f /var/log/messages 라고 입력한다.

이 창에서는 이제 pppd가 보내오는 메시지가 출력된다.

먼저 있던 창에서 ./start 라고 입력해 아까 만들었던 스크립트를 실행시킨다.

길게 잡아 10초 정도 기다린다.

이때 메시지 창에 다음과 같이 IP를 받았다고 메시지가 나오면 성공이다.

```
Nov 30 17:46:33 home kernel: PPP: version 2.3.7 (demand dialling)
Nov 30 17:46:33 home kernel: PPP line discipline registered.
Nov 30 17:46:33 home kernel: registered device ppp0
Nov 30 17:46:33 home pppd[659]: pppd 2.3.10 started by root, uid 0
Nov 30 17:46:33 home pppd[659]: Using interface ppp0
Nov 30 17:46:33 home pppd[659]: Connect: ppp0 <--> /dev/ttyp0
Nov 30 17:46:34 home pppd[659]: local  IP address 128.134.107.16
Nov 30 17:46:34 home pppd[659]: remote IP address 128.134.107.1
```

이중에서 Connect 까지만 떨어지고 먹통이라면, 이는 스크립트를 잘못 만들었을 경우나 아까의 pap-secrets 파일의 아이디와 스크립트의 아이디가 다른 경우 등으로 생각할 수 있다.

중간에 warning 이나 error가 떨어진다면 스크립트가 잘못되었을 가능성이 높다.

2.1.7 접속을 확인한다.

메시지 창에서 보듯 이렇게 IP를 받아왔다면 접속은 성공한 것이다.

네트웍 상태를 보기 위해 ifconfig 를 다시 해보자.

```
[leesop@www pppoe]$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:C0:26:11:8D:BD
          inet addr:192.168.0.10  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1400  Metric:1
          RX packets:15 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:10 Base address:0xe400

eth1      Link encap:Ethernet  HWaddr 00:00:E8:ED:20:A0
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 txqueuelen:100
          Interrupt:9 Base address:0xe000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:32 errors:0 dropped:0 overruns:0 frame:0
          TX packets:32 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0

ppp0      Link encap:Point-to-Point Protocol
          inet addr:128.134.107.5  P-t-P:128.134.107.1  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1492  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
```

맨 아래에 ppp0 이라는 접속 어댑터가 생겼다. 인터넷 연결이 잘 된 것이다.

참고로 인터넷 사용 전에 /etc/resolv.conf 파일을 편집해서 네임 서버 주소를 넣어줘야 www.hitel.net 같은 식으로 인터넷을 사용할 수 있다. (재부팅이나 접속 단절 등은 전혀 필요 없이 그냥 vi 등으로 에디팅 해주고 파일만 저장하면 바로 효력이 발생한다.)

2.1.8 행복한 마음으로 접속을 끊는다.

ps -aux 라고 치면, 프로세스 목록이 나온다.

이중에 pppd 를 kill로 죽여주면 된다.

아까의 메시지 창에 접속이 성공적으로 종료되었다는 말과 함께, 전송한 바이트 수와 사용 시간이 나올 것이다.

2.2 커널 패치를 통해 하는 방법 (pppoed 0.4 사용)

많은 경우에 성공을 했다는 사례가 보고되는 방법이고, 오버헤드가 앞의 방법 보다 적어 더 좋은 성능을 낸다고 알려져 있다. 그러나 커널 버전이 맞아야 하고, 커널 컴파일에 약간의 지식이 있어야 한다는 단점도 있다.

1. 준비 <http://www.davin.ottawa.on.ca/pppoe/>에 접속해 pppoed0.4.tgz 파일을 다운로드 한다.

   <ftp://ftp.kernel.org/pub/linux/kernel>에서 깨끗한 커널 소스를 받아오거나 커널 소스 패키지를 설치한다.

   현재 앞의 0.4 버전의 pppoed 프로그램은 커널 2.0.38 2.2.13 2.3.29 만을 공식으로 지원하고 있다.

2. 커널을 패치한다.

   적당한 디렉토리에 pppoed 프로그램을 풀면, 커널버전-pppox.gz 라는 파일이 생겨난다.

   이 파일을 커널 소스에 적용한다.

   편의상 커널 소스를 /usr/src/linux 에 풀어놓았다고 가정한다.

   ```
   cd /usr/src/linux
   zcat /압축을 푼 디렉토리.../커널버전-pppox.gz | patch -b -p1
   ```

3. 커널을 설정한다.

   시스템에 맞게 적당하게 커널을 설정한다.

   이때 pppoe를 사용하기 위해서 network devices 항목 안에 있는 “tty support for pppox” 를 켠다.

   커널 2.3.x 에서는 ‘PPP support for sync tty ports’ 항목도 켠다.

4. 컴파일을 한다.

   앞서 설정한 커널을 컴파일/설치 한다.

   pppoed 를 컴파일 한다.

   디렉토리로 들어가서

   ```
   make realclean
   make dev
   make depend
   make
   make install
   ```

   을 순서대로 수행한다.

   시스템을 재시작 한다.

5. 랜 카드와 pppd를 설정한다.

   랜 카드에서 IP 는 0.0.0.0 , 기본 게이트웨이는 부여하지 않는다.

   `/etc/ppp/options` 파일을 다음과 같이 한다.

   ```
   lock
   local
   nocrtscts
   noauth
   #be careful with mtu/mru if you are masquareding.
   # Look at Kal Lin's page at http://www.cs.toronto.edu/~kal/hse/resource.html
   mru 1490
   mtu 1490
   #please make sure you have noaccomp for now
   noaccomp
   #the construct below is needed by sympatico
   name "leesop@local"
   #you might want to change defaultroute if you have more
   #than one pppoe session
   defaultroute
   hide-password
   sync
   #it might be a good idea to uncoment the debug below
   #debug
   #kdebug 7
   #if you use the -R option to make it persistent
   #then uncomment the next two lines below
   #lcp-echo-interval 240
   #lcp-echo-failure 3
   #nodetach
   ```

   위의 name 항목에는 앞의 윈도우에서 WinPoET으로 접속할때 로그인창에 쓰던 아이디를 넣는다.

   `/etc/ppp/pap-secrets` 파일을 다음과 같이 한다.

   ```
   # Secrets for authentication using PAP
   # client        server  secret                  IP addresses
   leesop@local * 12345
   ```

   역시 앞의 name 항목에 쓴 것과 같은 접속에 사용되는 아이디와 암호를 적는다.

6. 접속해보기

   pppoed 프로그램을 실행한다.

   자동으로 접속이 되면서 연결이 된다.

   옵션이 몇가지 있는데, 이는 pppoed -help 라고 입력하면 얻을 수 있다.

기타 설치에 대한 자세한 사항은 내장된 INSTALL 문서를 참조하기 바란다.

3 사용 방법

3.1 접속 방법

root 아이디로 그냥 간단하게 아까의 디렉토리에서 만든 스크립트 (enternet 100 사용시) 나 pppoed (커널 패치법에서)를 실행하면 된다.

리눅스에서 네트웍 접속/단절 등은 일반 이용자 아이디로 되지 않는다.

자동으로 부팅때마다 실행하게 하는 방법은 /etc/rc.d/ 아래의 파일 등을 간단하게 손대주면 된다. 여기서는 접속과는 무관한 일이기 때문에 언급을 생략한다.

다만, 아래의 관련 링크에 언급된 FAQ 페이지에 이를 다루고 있는 곳이 있다.

3.2 접속 종료

랜카드를 죽일때와 동일하다.

```
/sbin/ifconfig ppp0 down
```

다른 방법으로는 불편하지만, 설치 과정에서 보듯, ps -aux 로 프로세스 목록을 보고, pppd를 kill 로 죽이면 된다.

다만, kill -9 해서 강제로 죽이지는 말것.

시스템 종료할때는 시스템이 알아서 자동으로 죽여주므로 걱정할 필요가 없다.

4 동작 원리

ADSL 모뎀을 설치하고 전원을 켜면, 다음과 같은 세 컴퓨터가 붙어있는 가상의 네트웍이 형성된다.

[내 컴퓨터]—-랜케이블—-[ADSL 모뎀]—–전화선—–[전화국의 내 서버 카드]

그리고, 전화국의 메인 서버가 전화국의 내 서버 카드에 IP를 할당하고, 이 패킷을 적절히 이 폐쇄 네트웍 안으로 포워딩해 줌으로써 인터넷 연결이 유지되는 것이다.

즉, 일부 분들의 설명과는 달리, 전화국내 서버 카드가 단순 라우터이거나, 인터넷으로 네트웍이 열려 있다는 식은 옳지 않다.

이전에 고정 IP 일때는 전화국의 내 서버 카드에 고정적으로 IP를 할당하였고, 중간에 PPPoE로 패킷을 패키징 하지 않았기 때문에 랜인것 처럼 내 컴퓨터가 인터넷을 사용할 수 있었다.

하나로 통신의 경우에는 이 세 컴퓨터가 붙은 네트웍 내에 Microsoft 계열의 VPN 프로토콜을 이용하게 되어 있다.

어쨌든, 이때 PPPoE 클라이언트를 돌리면 다음과 같은 순서로 연결이 일어난다.

1. 내 컴퓨터에서 망 전체에 "누구 PPPoE 서버 없냐?" 하고 외쳐본다. "혹은 이런 서비스명을 가진 PPPoE 서버는 없냐?" 하고 물어본다.
2. 전화국 카드가 "여기 있다. 내 서비스명은 이거, 카드 ID는 이거다." 라고 말한다. (참고로 필자의 Access Concentrator ID는 13021119814491-youngdong_adsl 이다.)
3. 그렇다면 그 서버로 TCP/IP 연결을 한다. (앞서 랜카드 설정에서 IP는 좋은데, 그래도 라우터는 설정하지 말아야 한다는 이유가 여기에 있다. 라우터 설정이 이상하면 이 단계서 저쪽과 연결을 못한다.)
4. 서버가 전화 접속의 PPP와 같은 데이터를 보내주고, 여기서 인증을 통해 접속한다. (인증은 일반적인 모뎀에서 쓰는 pap 인증을 사용중이다. 비슷한 chap 인증도 있지만 여기선 사용하지 않는다.)

이렇게 연결되면, 계속 내 컴퓨터와 전화국 카드는, TCP/IP 패킷을 PPP 프로토콜로 포장을 해서 넘기고 받는 일을 반복한다. 물론, 이때 전화 PPP에 사용하는 IP 헤더 압축 전송이나 PPP 고유의 데이터 압축 테크닉 등을 그대로 이용할 수 있다. (실제 지금의 한국통신 코넷 PPPoE 서버가 어디까지 지원하는지는 불확실하지만.)

리눅스에서는

"프로그램 –[pppd]–> ppp0 –[pty-redir]–> ttyp0 –[pppoe]–> eth0 -> 네트웍" 또는 "프로그램 –[pppd]–> ppp0 –> pppox0 –> eth0 -> 네트웍" (커널 패치시에) 의 순서로 패킷이 움직인다.

연결을 끊을때는 마찬가지로 PPP 종료 신호를 보내고 TCP/IP 연결을 끊는다. (아까의 스크립트 구성에서 포트를 먼저 올리고, PPP를 올리는 이유가 여기에 있다. 다른 스크립트 중에는 PPP를 먼저 올리고 포트를 나중에 올리는데, 이렇게 해버리면 종료시에 포트가 먼저 사라져 버린다. 정상 종료를 할 수 없게 된다.)

문제의 많은 부분은, pppoe 프로그램과 pppd가 연결되는 과정이 제대로 되지 않아 발생하는 것으로 보인다. 사실 인증자체는 pppd가 알아서 다 처리하므로 별로 신경 쓸 여지가 없다. 앞으로도 지속적인 안정화가 필요한 상황...

5 결어/관련 사이트

이상으로 간단하게 사용법을 정리해 보았다.

사실, 여기에 소개된 방법 이외에 많은 방법들이 있으니 혹시 실패하더라도 여러가지 방법을 시도해 보기 바란다.

참고로, 곧 발표될 리눅스 커널 2.4 에는 앞의 구현처럼 아예 커널에 ppp 처럼 PPPoE를 공식으로 내장한다니, 속도나 호환성 문제는 더 많이 개선될 것으로 전망한다.

끝으로 이 문서 작성에 참고한 인터넷 웹 사이트 몇개를 나열한다.

부디, 독자 여러분의 생활에 이 글이 보탬이 되었기를 바란다.

- `<http://www.sympaticousers.org/>`

  캐나다의 ISP인 Sypatico 사의 서비스의 이용자 모임 ADSL 관련 자료와 각종 자료들이 많이 누적되어 있다.

  한국통신의 ADSL 망은 이 회사의 것과 아주 유사하다. 수 많은 세팅법과 팁, FAQ, 포럼이 모여 있으므로 꼭 리눅스 유저가 아니어도 돌아보기 바란다.

- `<http://www.sympaticousers.org/linux_pppoe.htm>`

  리눅스에서 PPPoE 를 사용하는 방법을 정리하고 있다. 필자의 글은 많은 부분을 여기와 여기에 연결된 링크에서 참고했다.

- `<http://sidhe.folkwolf.net/~chris/pppoe-howto/>`

  역시 또 하나의 완벽한 FAQ 페이지, 필요한 프로그램은 모두 구할 수 있다. 앞서도 프로그램은 모두 여기서 구해왔다.

- <http://www.cs.toronto.edu/~kal/hse/resource.html>
  링크와 관련 자료가 정리되어 있는 페이지, 역시 추천한다.