Configuration HOWTO

저자: Guido Gonzato, *guido@ibogfs.cineca.it* <mailto:guido@ibogfs.cineca.it>

번역: 박민석, *dolman@correl1.snu.ac.kr* <mailto:dolman@correl1.snu.ac.kr> v0.99.3, 4 November 1997

Configuration HOWTO의 목적은 새로 설치한 리눅스 시스템의 세부 설정을 더 빠르고 쉽게 하도록 하는 것이다. 이 문서에는 널리 쓰이는 응용 프로그램들을 위한 설정이 포함되어 있으며, 이를 이용하면 쓰기 쉬운 시스템에서 작업을 시작할 수 있을 것이다.

목 차

제 1 절 소개

필자는 많은 PC에 리눅스를 설치하면서 현재의 배포본들이 매우 훌륭하다는 것을 알았지만, 때로는 기본적인 설정이 빠져 있는 번거로움을 겪기도 했다. 대부분의 프로그램들은 리눅스에서 제대로 작동하지만, 그렇지 않은 것도 있다.

이것은 부끄러운 일이다. 필자는 초보자들이 케케묵은 "백스페이스 문제"를 감내해야 하거나, 이맥스에는 색깔이 없다고 생각해야 하거나, 혹은 디렉터리의 내용을 검은색과 흰색으로만 보야야 할 이유 따위는 어디에도 없다고 생각한다. 필자는 가장 먼저 설정해야 할 것들을 기억하기 위해서 일람표를 작성했는데, 이 HOWTO는 그 일람표를 발전시킨 것이다. 이 문서를 읽는 동안 가장 흔한 응용 프로그램들을 위한 설정의 사례들을 많이 발견할 수 있을 것이다.

이 사례들은 레드햇 배포본을 이용해서 리눅스 시스템을 설치해 온 필자의 개인적인 경험에 기초를 둔 것이다. 따라서 다른 배포본을 사용하고 있다면 주의하여야 한다. 이 문서의 내용을 곧이 곧대로 받아들이지는 않기 바란다. 이 문서의 사례들은 필자의 경우 제대로 먹혀들었던 것들일 뿐이다. 더 많은 것들을 알고 싶다면, 패키지에 딸려온 문서들과 관련 HOWTO들을 읽어보아야만 한다. 이 HOWTO는 단지 간편하고 쓸모있는 일람표이고자 할 뿐이다.

PC의 하드웨어 사양에는 끝이 없지만, 필자의 경험에 따르면 세 개의 파티션(하나는 DOS/Windows, 하나는 리눅스, 하나는 스왑)으로 나뉘어진 대용량 하드디스크와 모뎀, CD-ROM 드라이브와 프린터, 마우스를 장착한 PC가 가장 흔한 듯 하다. 필자는 이런 하드웨어 사양에서 리눅스를 설정한다고 가정하겠다. 하지만 아래에 설명될 팁을 다른 사양에 적용하는 것은 쉬운 일이다. 이후 파일을 편집하거나 수정하거나 해킹할 때에는 루트의 권한을 가지고 있다고 암묵적으로 가정하도록 할 것이다. 자, 그럼 이제 시작해 보자.

제 2 절 일반적인 시스템 설정

2.1 키보드

우선 키보드를 설정하는 방법을 알아야 한다. 설치하는 동안 이 과정을 빠트렸거나 키보드를 바꾸고자 한다면, 아래의 절차를 따라야 한다.

```
- /usr/lib/kbd/keytables에서 적당한 키 테이블을 선택한다.  예를 들어
  이탈리아 키보드라면 it.map을 고른다.
- /etc/sysconfig/keyboard를 편집해서 아래와 같이 되도록 한다.
    KEYTABLE="/usr/lib/kbd/keytables/it.map"
- 키보드의 반복 비율과 지연 시간을 설정하려면, /etc/profile이나 rc
  파일 가운데 하나에 아래의 행을 추가한다.
    /sbin/kbdrate -s -r 16 -d 500  # or whatever
- 끝으로 키보드를 설정할 권한을 얻기 위해 아래와 같이 해준다.
    ~# chmod 666 /dev/port
```

키보드는 다음에 부팅할 때부터 변경된 사항대로 작동하게 된다.

2.2 커널

다음에 해야 할 일은 자신의 시스템에 가장 적합하도록 커널을 재구성하는 것이다. 매우 쉽기는 하지만, 반드시 /usr/src/linux의 README 파일을 읽어보아야 한다. 요령은 다음과 같다.

```
- 자신이 원하는 것이 무엇인지 잘 생각해야 한다.  한 달이 멀다 하고
  재 설정, 재 컴파일 하는 것보다는, 한 번에 적절한 커널을 선택하고
  패치를 끝내는 편이 더 생산적이다.  특히 서버로 쓰는 리눅스
  시스템의 경우에는 이 점이 중요하다.  장차 추가할 가능성이 있는
  모든 하드웨어를 지원하도록 하는 것을 잊어서는 안된다.
- 사이릭스 CPU를 쓰는 PC라면 성능 향상을 위해 적절한 패치를
  가해야 한다.
```

www.linuxhq.com/patch/20-p0591.html `<http://www.linuxhq.com/patch/20-p0591.html>`

```
  에 있는 정보를 참조하기 바란다.
- 노트북 사용자라면 noblink 패치를 써서 화면을 좀 더 읽기 쉽게
  하는 편이 좋다.
- 노트북 사용자가 PCMCIA 모뎀/팩스를 쓸 계획이라면, 시리얼
  지원을 모듈로 컴파일해서는 안된다.  커널 내부에 포함시키지
  않으면 PCMCIA 모뎀이 작동하지 않는다.
```

2.3 부트업 메시지

부트업 메시지를 취향에 맞추고 싶으면, /etc/rc.d/rc.local이 /etc/issue와 /etc/motd를 덮어쓰는 지 확인하여야 한다. 덮어쓰게 되어 있다면 편집해서 바꾸도록 하라.

2.4 호스트 이름

"hostname new_host_name"이라고 명령을 내리는 것으로는 충분하지 않다. /etc/sysconfig/network 파일을 편집해서 그 안의 호스트 이름을 바꿔야 한다.

2.5 센드메일에 의한 지체

어떤 시스템에서는 부팅할 때 센드메일 때문에 지체되곤 한다. 이런 경우에는 /etc/hosts 파일에

127.0.0.1 localhost

라는 행이 들어있는지 확인해서 없으면 추가해야 한다.

2.6 디바이스 드라이버

/dev에 들어있어야 할 디바이스가 (혹은 실제 디바이스 드라이버로 연결된 링크가) 없을 수도 있다. 자신의 마우스, 모뎀, CD-ROM 드라이버가 어느 디바이스에 대응되는지 확인한 뒤 아래와 같이 하면 된다.

```
~# cd /dev
/dev# ln -s /dev/cua0 mouse
/dev# ln -s /dev/cua1 modem
/dev# ln -s /dev/hdb cdrom
```

그리고 필요하다면 "chmod 666"으로 모든 사용자들이 이 디바이스들을 마음대로 쓰도록 한다. 팁: 어떤 랩탑 컴퓨터의 경우에는 마우스 디바이스가 /dev/psaux이다. X 윈도우를 설정할 때 이 점도 고려해야 한다.

추가로 "chmod 666 /dev/fd*"로 루트 이외의 사용자가 플로피 디스크 드라이브를 사용할 수 있도록 하고 싶을 수도 있다. 이렇게 하는 것은 보안 문제를 일으킬 수 있지만, 필자는 자세한 내용을 모른다. 이 점에 대한 조언을 부탁한다.

2.7 마우스

gpm 마우스 서비스를 이용하면 콘솔에서 복사/삽입을 하거나 일부 응용 프로그램에서 마우스를 사용하는데 편리하다. /etc/sysconfig/mouse 라는 파일에 아래와 같은 행이 있는지 확인하기 바란다.

```
MOUSETYPE="Microsoft"
XEMU3=yes
```

아울러 /etc/rc.d/init.d/gpm 파일이 있어야 한다. 물론 이 설정이 자기의 마우스에 맞는지 확인해야 한다. 팁: 어떤 랩탑 컴퓨터에서는 MOUSETYPE이 "PS/2"이다.

2.8 마운트 지점

플로피와 CD-ROM, 도스 파티션 및 기타 장치들에 대한 마운트 지점을 가지고 있는 것이 편하다. 예를 들어 다음과 같이 할 수 있다.

```
~# cd /mnt
/# mkdir a: ; mkdir floppy ; mkdir cdrom ; mkdir win ; mkdir zip
```

이렇게 함으로써 MS-DOS 플로피와 ext2 플로피, CD-ROM, 도스 파티션, 병렬 포트를 이용하는 Zip 드라이브의 마운트 지점을 만들 수 있다. 다음에는 /etc/fstab 파일을 편집해서 아래의 내용을 추가한다.

```
/dev/fd0        /mnt/a:         msdos          user,noauto      0 1
/dev/fd0        /mnt/floppy      ext2            user,noauto     0 1
/dev/cdrom      /mnt/cdrom     iso9660        ro,user,noauto    0 1
/dev/hda1       /mnt/dos        msdos          user,noauto      0 1
/dev/sda4       /mnt/zip        vfat             user,noauto     0 1
```

물론 첫 번째 항목에는 정확한 디바이스를 써야 한다. 윈95의 긴 파일 이름을 사용하기 위해 마지막 줄에서는 msdos 대신 vfat를 사용했다. 하지만 (아직은) vfat32에는 이것이 적용되지 않는다.

2.9 LILO와 Loadlin

많은 사용자들이 PC에 리눅스와 도스/윈도즈를 함께 쓰기를 바라고 있고, 또 부팅할 때 어느 운영체제를 쓸 것인지 선택하고 싶어한다. /dev/hda1에는 도스/윈도즈가 있고, /dev/hda2에는 리눅스가 있다고 할 때, 아래와 같이 하면 된다.

```
~# fdisk
Using /dev/hda as default device!
Command (m for help):a
Partition number (1-4): 2
Command (m for help):w
~#
```

이렇게 하면 리눅스 파티션이 부팅 가능하게 된다. 이 단계는 LILO의 QuickInst를 실행할 때, 활성화(activate)에 의해서 수행되어야 하지만, 필자의 레드햇에서는 작동되지 않았다. 아래와 같이 기본적인 /etc/lilo.conf 파일을 쓰도록 한다.

```
boot = /dev/hda2
compact
delay = 50
# message = /boot/bootmesg.txt  # write your own
root = current
image = /boot/vmlinuz
label = linux
other = /dev/hda1
table = /dev/hda
label = dos
```

이제 /sbin/lilo를 수행하면 설정은 끝난다. LILO는 설치과정에서 매우 중요한 부분이기 때문에, 반드시 리로에 딸린 문서를 읽도록 강력히 권고한다. 도스에서 리셋 단추를 누르지 않고 리눅스를 부팅시키려면, LOADLIN.EXE를 도스의 패스에 포함된 (도스 파티션 내의!) 디렉터리에 위치시킨다. 다음에는 자신의 커널을 예를 들어 C:\DOS\VMLINUZ로 복사한다. 이제 아래와 같은 배치 파일로 리눅스를 부팅시킬 수 있다.

```
rem   linux.bat
smartdrv /C
loadlin c:\dos\vmlinuz root=/dev/hda2 r
```

윈도즈 95를 쓴다면, 이 배치 파일의 속성을 도스 모드에서 시작하는 것으로 설정해야 한다.

2.10 프린터 설정

레드햇에는 훌륭한 설정 도구가 있어서 필자는 HP 데스크젯 400 설정을 멋지게 끝낼 수 있었다. 만약 레드햇을 쓰지 않는다면 아래와 같이 수동 설정을 해야 한다.

포스트스크립트를 지원하지 않는 프린터로 (C 소스 파일 같은) 텍스트 파일과 포스트스크립트 파일을 인쇄하고 싶다고 하자. 포스트스크립트를 인쇄하려면 고스트스크립트를 써야 하므로 미리 설치되어 있다고 가정하기로 한다.

프린터 설정은 몇 가지 단계로 이루어진다.

```
- 병렬 프린터 디바이스가 무엇인지 알아보기 위해, 다음과 같이 하고
     ~# echo "hello, world" > /dev/lp0
     ~# echo "hello, world" > /dev/lp1
  어느 것이 작동하는지 적어둔다.
- 두 개의 스풀 디렉터리를 만든다.
     ~# cd /var/spool/lpd
     /var/spool/lpd/# mkdir raw ; mkdir postscript
- 프린터가 "계단 현상"을 보인다면 (대부분의 잉크젯) 필터가
  필요하다.  아래와 같이 하여 두 줄을 프린트해 본다.
     ~# echo "first line" > /dev/lp1 ; echo "second line" > /dev/lp1
  출력 결과가
     first line
                second line
  이면, 아래의 파일을 /var/spool/lpd/raw/filter라고 저장하고,
```

```
#!/bin/sh
# This filter does away with the "staircase effect"
awk '{print $0, "\r"}'
```

"chmod 755 /var/spool/lpd/raw/filter"로 실행 가능하게 한다.

- 포스트스크립트 에뮬레이션(모방)을 위한 필터를 만들자. 아래와 같은 필터를 /var/spool/lpd/postscript/filter 로 저장한다.

```
#!/bin/sh
DEVICE=djet500
RESOLUTION=300x300
PAPERSIZE=a4
SENDEOF=
nenscript -TUS -ZB -p- |
if [ "$DEVICE" = "PostScript" ]; then
        cat -
else
        gs -q -sDEVICE=$DEVICE \
               -r$RESOLUTION \
               -sPAPERSIZE=$PAPERSIZE \
               -dNOPAUSE \
               -dSAFER \
               -sOutputFile=- -
fi
if [ "$SENDEOF" != "" ]; then
        printf "\004"
fi
```

(여기서는 HP 데스크젯 프린터를 가정했다. 자신의 프린터에 맞추도록 한다.)

- 끝으로 아래의 내용을 /etc/printcap에 추가한다.

```
# /etc/printcap
lp|ps|PS|PostScript|djps:\
        :sd=/var/spool/lpd/postscript:\
        :mx#0:\
        :lp=/dev/lp1:\
        :if=/var/spool/lpd/postscript/filter:\
        :sh:
raw:\
        :sd=/var/spool/lpd/raw:\
        :mx#0:\
        :lp=/dev/lp1:\
        :if=/var/spool/lpd/raw/filter:\
        :sh:
```

더 복잡하거나 특이한 프린팅 설정이 필요하면 Printing-HOWTO를 읽어보기 바란다.

레드햇 사용자들을 위해: 프린트툴이 선택한 GSDEVICE는 제대로 작동할 것이다. 하지만 이것이 자신의 프린터에 가장 알맞은 것은 아닐 수도 있다. postscript.cfg 파일의 사소한 내용들을 살펴보는 것이 좋다. 예를 들어 필자의 경우에는 GSDEVICE를 cdj500에서 djet500으로 바꿨더니 출력 속도가 훨씬 빨라졌다.

제 3 절 소프트웨어 설정

3.1 bash(1)

bash를 길들이려면 다음과 같은 파일들을 편집해야 한다.

```
- /etc/bashrc에는 시스템 규모의 alias들과 함수들이 들어있다.
- /etc/profile에는 시스템 규모의 환경 설정과 시작 프로그램들이
  들어있다.
- $HOME/.bashrc에는 사용자가 정의한 alias들과 함수들이 있다.
- $HOME/.bash_profile에는 사용자가 결정한 환경 설정과 시작
  프로그램들이 들어있다.
- $HOME/.bash_logout에는 로그 아웃할 때 수행될 행동이 들어있다.
- $HOME/.inputrc에는 키 결정(binding)과 기타 내용들이 있다.
```

예제 파일들을 아래에 적어 두었다. 첫 번째는 가장 중요한 /etc/profile 이다. 이 파일은 이어지는 여러 절에서 보게 되듯이 리눅스 시스템의 많은 특성을 결정하기 위해 쓰인다.

```
# /etc/profile
# 시스템 규모의 환경 및 시작 프로그램.
# 함수와 alias는 /etc/bashrc에 적도록 한다.
# 이 파일은 다음과 같은 특성을 정한다.
#
#   o 패스
#   o 프롬프트
#   o 몇 가지 환경 변수들
#   o 컬러 ls
#   o less의 특성
#   o 키보드 설정
#
#  사용자들은 각자의 $HOME/.bash_profile을 편집해서 이 파일의
#  설정을 무시하게 할 수도 있고, 다른 특성을 추가할 수도 있다.
#  추가 패스 설정(set a decent path)
   echo $PATH | grep X11R6 > /dev/null
   if [ $? = 1 ] ; then   # add entries to the path
     PATH="$PATH:/usr/X11R6/bin:$HOME/bin:."
   fi
# 사용자에게 로그인 쉘인지 아닌지 알려준다. 로그인 쉘이면 푸른 색,
# 아니면 자홍색 프롬프트를 쓰고, 루트는 붉은 프롬프트를 쓴다.
USER=`whoami`
if [ $LOGNAME = $USER ] ; then
   COLOUR=44
else
   COLOUR=45
fi
if [ $USER = 'root' ] ; then
   COLOUR=41
fi
# ^[ 대신 진짜 esc 키를 쓴다.
PS1='^[[$COLOUR;37;1m$HOSTNAME:^[[37;40;1m\w\$ '
```

```
PS2="Continue> "
# core dump를 만들지 않는다.
ulimit -c 0
# umask 설정
if [ `id -gn` = `id -un` -a `id -u` -gt 14 ]; then
   umask 002
else
   umask 022
fi
# 몇 가지 변수들
USER=`id -un`
LOGNAME=$USER
MAIL="/var/spool/mail/$USER"
EDITOR=jed
HOSTNAME=`/bin/hostname`
HISTSIZE=1000
HISTFILESIZE=1000
export PATH PS1 PS2 USER LOGNAME MAIL EDITOR HOSTNAME
HISTSIZE HISTFILESIZE
# 컬러 ls를 사용
eval `dircolors /etc/DIR_COLORS -b`
export LS_OPTIONS='-F -s -T 0 --color=tty'
# less를 취향에 맞게 바꾼다.
LESS='-M-Q'
LESSEDIT="%E ?lt+%lt. %f"
LESSOPEN="| lesspipe.sh %s"
VISUAL=jed
LESSCHARSET=latin1
export LESS LESSEDIT LESSOPEN VISUAL LESSCHARSET
# 키보드도 취향에 맞춘다.
/sbin/kbdrate -s -r 16 -d 500
for i in /etc/profile.d/*.sh ; do
   if [ -x $i ]; then
     . $i
   fi
done
```

다음은 /etc/bashrc 파일의 예다.

```
   # /etc/bashrc
   # 시스템 규모의 함수들과 alias들
   # 환경 설정에 관계된 내용은 /etc/profile에 두어야 한다.
   alias which="type -path"
   alias d="ls"
   alias dir="d"
```

다음은 .bashrc 파일이다.

```
# $HOME/.bashrc
# 시스템 규모로 정의된 것들을 적용한다.
if [ -f /etc/bashrc ]; then
  . /etc/bashrc
fi
# 아래는 사용자에게 비-로그인 쉘에 있다고 알려주기 위해 필요하다.
COLOUR=45
# ^[ 대신 진짜 Esc 문자를 쓰도록 한다.
PS1='^[[$COLOUR;37m$USER:^[[37;40m\w\$ '
# alias들
alias cp='cp -i'
alias l=less
alias lyx='lyx -width 900 -height 700'
alias mv='mv -i'
alias rm='rm -i'
alias x=startx
# 몇 가지 유용한 함수들
inst() # 현재 디렉터리에 .tar.gz 압축 파일을 설치한다.
{ gzip -dc $1 | tar xvf - }
cz() # .zip 압축 파일의 내용 목록을 출력한다.
{ unzip -l $* }
ctgz() # .tar.gz 압축 파일의 내용 목록을 출력한다.
{
  for file in $* ; do
  gzip -dc ${file} | tar tf -
  done
}
tgz() # gzip과 tar로 압축 파일을 만든다.
{
  name=$1 ; tar -cvf $1 ; shift
  tar -rf ${name} $*
  gzip -S .tgz ${name}
}
```

다음은 .bash_profile 파일이다.

```
# $HOME/.bash_profile
# 사용자가 정한 환경과 시작 프로그램들
# 이 파일에는 /etc/profile의 정의에 우선하는 사용자 정의 설정이 있다.
# alias와 함수들을 적용한다.
if [ -f ~/.bashrc ]; then
  . ~/.bashrc
fi
# PS1 설정을 다시 적용한다.
if [ $USER = 'root' ] ; then
  COLOUR=41
else
  COLOUR=44
fi
# ^[ 대신 진짜 Esc를 쓴다.
```

```
PS1='^[[$COLOUR;37;1m$HOSTNAME:^[[37;40;1m\w\$ '
export PS1
```

다음은 .bash_logout 파일이다.

```
# $HOME/.bash_logout
clear
```

다음은 .inputrc 파일이다.

```
# $HOME/.inputrc
# 키 설정
"\e[1~": beginning-of-line
"\e[3~": delete-char
"\e[4~": end-of-line
# (F1 .. F5) are "\e[[A" ... "\e[[E"
"\e[[A": "info \C-m"
set bell-style visible          # beep 음을 내지 않는다.
set meta-flag On               # 8 비트 입력을 허용 (i.e, accented
set convert-meta Off           # 8 비트 문자를 변환하지 않는다.
set output-meta On            # 8 비트 문자를 제대로 표시한다.
set horizontal-scroll-mode On
set show-all-if-ambiguous On
```

xterm을 비롯한 X11 응용 프로그램에서 백스페이스와 delete 키가 제대로 작동하도록 하기 위해서는 아래의 작업이 필요하다.

- 아래의 내용을 자신의 .xinitrc에 넣는다.

```
usermodmap=$HOME/.Xmodmap
xmodmap $usermodmap
```

- 아래의 내용을 자신의 .Xmodmap에 포함시킨다.

```
keycode 22 = BackSpace
keycode 107 = Delete
```

이렇게 해서 콘솔을 제대로 맞춘 다음에는, xterm을 수정한다.

- 아래의 내용을 .Xdefaults에 넣는다.

```
xterm*VT100.Translations: #override
        <Key>BackSpace:       string(0x7F)\n\
        <Key>Delete:           string(0x1b) string("[3~")\n\
        <Key>Home:            string(0x1b) string("[1~")\n\
        <Key>End:              string(0x1b) string("[4~")\n\
        Ctrl<Key>Prior:         string(0x1b) string("[40~")\n\
        Ctrl<Key>Next:         string(0x1b) string("[41~")
nxterm*VT100.Translations: #override
        <Key>BackSpace:       string(0x7F)\n\
        <Key>Delete:           string(0x1b) string("[3~")\n\
        <Key>Home:            string(0x1b) string("[1~")\n\
        <Key>End:              string(0x1b) string("[4~")\n\
        Ctrl<Key>Prior:         string(0x1b) string("[40~")\n\
        Ctrl<Key>Next:         string(0x1b) string("[41~")
```

정보가 더 필요하면 bash(1)과 readline(3)의 매뉴얼 페이지를 보기 바란다. 모든 응용 프로그램이 정확히 작동하리라고는 기대하지 말아야 한다. 예를 들어 xterm에서 joe를 실행시키면 몇 가지 키가 작용하지 않을 것이다. rxvt에서도 마찬가지다. 이런 것은 termcap의 문제라는 소문이 있다.

3.2 ls(1)

서로 다른 파일 유형을 강조하기 위해, 컬러 ls로 디렉터리 목록을 출력할 수 있다. 이렇게 하려면 아래의 행들을 /etc/profile에 추가해야 한다.

```
eval `dircolors /etc/DIR_COLORS -b`
export LS_OPTIONS='-F -T 0 --color=tty'
```

이 두 줄은 /etc/DIR_COLORS에 준비된 색상표를 포함하는 LS_COLORS 환경 변수를 설정하는 것이다. 주의: 이 방법은 rxvt의 특정 버전에서는 제대로 작동하지 않는데, 필자도 그 까닭은 모른다. 대신 xterm 같은 것을 쓰도록 하자. 아마 rxvt 내에 어떤 환경에서는 환경을 정확히 물려받지 못하도록 하는 버그가 있는 듯 하다.

3.3 less(1)

이 탁월한 페이저(pager)는 일반 텍스트 파일 뿐 아니라 gzip으로 압축된 파일이나, tar와 zip으로 묶여진 파일, 매뉴얼 페이지 등도 볼 수 있다. less의 설정은 몇 단계로 이루어진다.

```
   - less에서 이동 키를 쓰려면, 아래와 같은 .lesskey라는 아스키 파일이
     자신의 홈 디렉터리에 있어야 한다.
        ^[[A    back-line
        ^[[B    forw-line
        ^[[C    right-scroll
        ^[[D    left-scroll
        ^[OA   back-line
        ^[OB    forw-line
        ^[OC    right-scroll
        ^[OD   left-scroll
        ^[[6~     forw-scroll
        ^[[5~     back-scroll
        ^[[1~     goto-line
        ^[[4~     goto-end
        ^[[7~     goto-line
        ^[[8~     goto-end
     다음에는 lesskey라는 명령을 실행한다.  이 명령은 키 설정을 담은
     .less 라는 이진 파일을 만들어 낸다.
   - 아래 파일을 /usr/bin/lesspipe.sh로 쓴다.

#!/bin/sh
# 이 파일은 'less'의 전처리기이다.
# 이 파일은 LESSOPEN="|lesspipe.sh %s" 라는 환경 변수가 설정되었을
# 때 사용된다.
lesspipe() {
case "$1" in
   *.tar) tar tf $1 2>/dev/null ;; # .tar와 .tgz 파일의 내용을 본다.
   *.tgz|*.tar.gz|*.tar.Z|*.tar.z) tar ztf $1 2>/dev/null ;;
```

```
   *.Z|*.z|*.gz) gzip -dc $1  2>/dev/null ;; # 압축 파일을 제대로 본다.
   *.zip) unzip -l $1 2>/dev/null ;; # 압축 파일을 본다.
   *.arj) unarj -l $1 2>/dev/null ;;
   *.rpm) rpm -q -p -i -l $1 2>/dev/null ;;
   *.1|*.2|*.3|*.4|*.5|*.6|*.7|*.8|*.9|*.n|*.man) FILE=`file -L $1`
   FILE=`echo $FILE | cut -d ' ' -f 2`
     if [ "$FILE" = "troff" ]; then
        groff -s -p -t -e -Tascii -mandoc $1
     fi ;;
   *) FILE=`file -L $1` ; # 이진 파일의 경우 보기 위해 점검한다.
   FILE1=`echo $FILE | cut -d ' ' -f 2`
   FILE2=`echo $FILE | cut -d ' ' -f 3`
   if [ "$FILE1" = "Linux/i386" -o "$FILE2" = "Linux/i386" \
     -o "$FILE1" = "ELF" -o "$FILE2" = "ELF" ]; then
        strings $1
   fi ;;
   esac
}
lesspipe $1
```

다음에는 chmod 755 lesspipe.sh해서 실행 가능하게 만든다.

```
   - 다음 몇 줄을 /etc/profile에 추가한다.
        LESS="-M-Q"                    # long prompt, silent
        LESSEDIT="%E ?lt+%lt. %f"      # edit top line
        LESSOPEN="| lesspipe.sh %s"     # filter
        VISUAL=jed                      # 기본 에디터-원하는 것으로.
        LESSCHARSET=latin1             # 필요하면 강조 문자를 쓴다.
```

필자가 이탈리아에 살고 있고 ISO 8859/1 문자 집합을 쓰고 싶기 때문에 LESSCHARSET 변수를 라틴으로 했다. 다른 나라 사람들은 이렇게 설정하지 않는 것이 좋다.

3.4 emacs(1)

몇 가지 이맥스 배포본들은 색깔과 구문 강조(syntax highlighting)를 위한 설정이 되지 않은 상태로 설치된다. 이 경우 아래의 내용을 자기 홈 디렉터리의 .emacs 파일에 써야 한다.

```
   (global-font-lock-mode t)
   (setq font-lock-maximum-decoration t)
```

이 설정은 X11에서만 작동한다. 이맥스를 각자의 필요에 맞게 길들이는 방법을 찾기 위해서는 스스로 이맥스 관련 문서들을 찾아보기 바란다. 이맥스를 제대로 설정하려면 여러 달 걸릴 수도 있다.

3.5 joe(1)

/usr/bin 밑에 jmacs나 jstar, jpico 등이 joe로 심볼릭 링크되어 있는지 아니면 따로 이진 파일로 되어 있는지 확인해 보아야 한다. 별도의 이진 파일로 되어 있다면, 이 파일들 대신 심볼릭 링크를 써서 디스크 공간을 아낄 수 있다.

```
   ~# cd /usr/bin
   /usr/bin# ln -sf joe jmacs ; ln -sf joe jstar ; ln -sf joe jpico
```

다음에는 /usr/lib/joe/joerc를 자기 홈 디렉터리에 .joerc라고 복사하고, 적절히 편집하면 된다.

3.6 jed

필자는 이 에디터를 좋아한다. 필자가 바라는 기능이 있고, 이맥스에 비해 작고 설정도 쉽다. 또 다른 에디터들을 상당히 쉽게 흉내낼 수 있다. 필자가 다니는 대학의 많은 사용자들은 VMS의 시스템 에디터인 EDT 비슷하게 jed를 쓰고 싶어한다.

```
.jedrc와 /usr/lib/jed/lib/* 가 설정 파일이다.  .jedrc는 /usr/lib/jed/lib의
jed.rc를 복사/편집해서 쓰면 된다.
   - jed가 몇 가지 키를 제대로 쓰도록 하려면 /usr/lib/jed/lib/defaults.sl
     파일을 만들어서
        () = evalfile("linux");
     라는 한 줄만 넣어주면 된다.
   - /usr/lib/jed/lib/linux.sl을 편집한다.  Info_Directory = "/usr/info"
     라는 행의 주석문 표시를 지우고, /bin/mail을 UCB_Mailer = 뒤에
     추가한다.
   - jed가 EDT(혹은 다른 에디터)를 흉내내도록 설정하는 것은 쉽다.
     .jedrc 내의 두 줄만 편집하면 된다.  숫자 키 패드의 '+'로 글자
     하나 대신 여러 낱말을 지우도록 하려면, 아래 내용을 .jedrc의 "() =
     evalfile("edt")"라는 행 아래에 추가한다.
        unsetkey("\e0l");
        unsetkey("\e0P\e0l");
        setkey("edt_wdel", "\e0l");
        setkey("edt_uwdel", "\e0P\e0l");
   - xjed가 숫자 키 패드를 써서 EDT를 흉내내도록 하려면 아래의
     내용을 .Xmodmap에 삽입한다.
        keycode 77  = KP_F1
        keycode 112 = KP_F2
        keycode 63  = KP_F3
        keycode 82  = KP_F4
        keycode 86  = KP_Separator
     다음에는  /etc/X11/XF86Config에 아래 두 줄이 포함되어 있는지
     확인하도록 한다.
        #    ServerNumLock  # must be commented out
        XkbDisable
     이 설정은 XFree 3.2에 적용된다.  하지만 표준 미국식 키보드를
     쓰지 않는다면, "XkbDisable"로 설정했을 경우 몇 가지 작은 문제가
     생긴다.  이 문제는 각자 해결해야 한다.
   - xjed의 색상을 취향에 맞게 하려면 아래와 같은 행을 .Xdefaults에
     추가하면 된다.
        xjed*Geometry: 80x32+150+50
        xjed*font: 10x20
        xjed*background: midnight blue
```

이 밖의 정보들은 패키지에 포함된 xjed.doc에 있다.

3.7 TeX 및 관련 프로그램들

TeTeX 배포본을 가지고 있다고 가정하도록 하자. 여기서는 몇 가지만 얘기한다.

- 자기가 쓰는 언어에 따른 하이픈 규칙을 설정하려면, /usr/lib/texmf/texmf/tex/generic/config/language.dat 파일을 편집하고, 다음 명령을 실행시킨다.
  ~# texconfig init ; texconfig hyphen
- LaTeX 패키지를 추가하려면 /usr/lib/texmf/texmf/tex/latex/ 아래에 파일을 추가한 다음에 texhash 명령을 수행해서 TeTeX이 새로운 패키지를 인식할 수 있도록 해야 한다.
- dvips 설정을 바꾸려면 /usr/lib/texmf/texmf/dvips/config/config.ps. 파일을 편집해야 한다. 기본 해상도에 대한 항목은 xdvi에도 영향을 준다는 점을 염두에 두기 바란다. xdvi를 실행시킬 때마다 폰트를 만드는 것이 귀찮다면, .Xdefault 안에 아래 한 줄을 넣으면 된다.
  XDvi*mfmode:

3.8 PPP

커널이 PPP + TCP/IP를 지원하도록 컴파일 되어 있고, 루프백(loopback)이 가능하며, pppd 패키지를 제대로 설치했다고 하자. (이 요구사항들은 기본적으로 되어 있어야 한다) PPP가 작동하도록 하는 방법은 가)수동으로 설정하거나 나)자동 설정 프로그램을 쓰는 두 가지가 있다. 먼저 수동 옵션을 보자.

자신의 ISP(인터넷 서비스 제공자)가 아래와 같다고 하자.

- 전화 번호: 1234567
- 네임 서버: 123.231.112.111
- 메일 서버: mbox.supernet.edu
- 도메인: supernet.edu
- 원격 호스트 이름:<url url="http://www.supernet.edu" name="www.supernet.edu">
- 사용자 이름: John
- 패스워드: _Loo%ny
  을 수동으로 설정하기 위해서는 아래와 같은 것을 해야 한다.
- 아래 내용을 ISP에 연결하기 위해 사용될 /usr/local/bin/ppp-on 파일로 저장한다.

```
#!/bin/sh
# ppp-on: #ISP에 연결하는 스크립트
INIT=ATX3  # 자기 모뎀에 맞게 쓴다.
PHONE=1234567
/usr/sbin/pppd connect "/usr/sbin/chat '' $INIT OK ATDP$PHONE
CONNECT '' " user John
```

  다음에는 chmod 755 /usr/local/bin/ppp-on 해서 실행 가능하게 한다.

혹시 자신의 ISP가 PAP 프로토콜을 쓰지 않는다면, 아래의 스크립트를 써야 한다.

```
#!/bin/sh
# ppp-on: #ISP에 연결하는 스크립트
INIT=atx3  # 자기 모뎀에 맞게 쓴다.
PHONE=1234567
/usr/sbin/pppd connect "/usr/sbin/chat '' $INIT OK ATDT$PHONE \
CONNECT '' 'ogin' John 'word' _Loo%ny ''" /dev/modem 38400 \
modem defaultroute
```

- 아래 내용을 ppp 연결을 종료하기 위해 쓰일 /usr/local/bin/ppp-off 파일에 쓴다.

```
#!/bin/sh
```

```
# ppp-off
DEVICE=ppp0
if [ -r /var/run/$DEVICE.pid ]; then
        kill -INT `cat /var/run/$DEVICE.pid`
        if [ ! "$?" = "0" ]; then
                echo "removing stale $DEVICE pid file."
                rm -f /var/run/$DEVICE.pid ; exit 1
        fi
        echo "$DEVICE link terminated" ; exit 0
fi
# link not active
echo "$DEVICE link is not active" ; exit 1
```

역시 chmod 755 /usr/local/bin/ppp-off로 실행 가능하게 한다.

- /etc/ppp/options에 아래와 같이 쓴다.

```
# /etc/ppp/options
/dev/modem
38400
lock
crtscts
modem
asyncmap 0
remotename <url url="http://www.supernet.edu" name="www.supernet.edu">
defaultroute
```

- ISP가 PAP 프로토콜을 쓰면 (대부분이 쓴다), /etc/ppp/pap-secrets 파일에 아래와 같이 쓴다.

```
# /etc/ppp/pap-secrets
# username    remotehost          secret
  John        <url url="http://www.supernet.edu" name="www.supernet.edu">    _Loo%ny
```

- /etc/resolv.conf를 편집한다.

```
# /etc/resolv.conf
search supernet.edu
nameserver 123.231.112.111
```

- /etc/sendmail.cf를 편집한다. DS로 시작하는 줄을 찾아서 아래와 같이 만든다.

```
DSmbox.supernet.edu
```

휴! 운이 좋다면, 이 정도로 PPP를 쓸 수 있다. 하지만 PPP-HOWTO를 공부할 준비를 해두기 바란다. 수동 설정은 할 일이 너무 많다. 하지만, 이 온갖 너절한 일들을 옛 일로 만들 수 있는 멋진 패키지가 있다. 아래 주소에서 ezppp 패키지를 당장 다운받도록 하자. *www.serv.net/˜cameron/ezppp/index.html* <http://www.serv.net/~cameron/ezppp/index.html>

3.9 POP 클라이언트

POP 서버에서 메일을 받아오려면 fetchpop나 fetchmail 같은 POP 클라이언트를 써야 한다. 자신의 PPP 서버가 LAST 명령에 문제를 보인다면 fetchmail을 써야할 것이다. 이 클라이언트들을 설정하는 방법은 아래와 같다.

- fetchpop: 처음 실행시킬 때 몇 가지 안내를 볼 수 있다. 질문에 대답하면 설정이 끝난다.
- fetchmail: 아래의 .fetchmailrc 예제를 쓰면 된다.

```
# $HOME/.fetchmailrc
poll mbox.supernet.edu with protocol pop3;
user john there with password _Loo%ny is john here
mda "/usr/bin/procmail -f fetchmail"
```

마지막 줄은 필요 없을 수도 있다. chmod 600 .fetchmailrc로 허가권을 주어야 하며, 허가권이 다르게 되어 있으면 fetchmail이 실행되지 않는다.

3.10 X 윈도우 시스템

비디오 카드 설정 등을 통해서 X 윈도우가 제대로 작동하게 만들고 나면, 수많은 설정이 가능하게 된다. 어떤 설정을 쓰는지는 각자 사용하는 윈도우 매니저에 따라 다르다. 어느 경우에나 자신의 홈 디렉터리에 있는 하나나 그 이상의 아스키 파일을 편집함으로써, 설정을 마치게 된다. 몇 가지 윈도우 매니저의 경우에는 아래와 같다.

```
 - fvwm: /etc/X11/fvwm/system.fvwmrc를 홈 디렉터리에 .fvwmrc라고
   복사하고, 그 내용을 살펴보면서 자기에게 맞춰 가도록 한다. 미리
   들어있는 system.fvwmrc는 약간 지나치게 단순해서 fvwm의 기능을
   충분히 활용하지 못한다.
 - fvwm95-2: /etc/X11/fvwm95-2/fvwm2rc95를 자신의 홈 디렉터리에
   .fvwm2rc95라고 복사한 뒤 편집한다. 이 설정은 상당히 훌륭하다.
 - TheNextLevel: 이 매니저는 설정하기가 비교적 까다롭다.
   /etc/X11/TheNextLevel/.*를 홈 디렉터리에 복사하고, 조심해서
   살펴본 뒤에 자기에게 맞도록 바꾸도록 한다. 가장 먼저 살펴볼
   파일은 .fvwm2rc.defines이다.

아울러 올바른 .xinitrc를 가지고 있어야 한다. 아래 예를 보자.
  #!/bin/sh
  # $HOME/.xinitrc
  # 몇 가지 키를 적절하게 설정한다.
  usermodmap=$HOME/.Xmodmap
  xmodmap $usermodmap
  xset s noblank  # 스크린 세이버를 끈다.
  xset s 300 2    # 5분 후에 스크린 세이버를 작동한다.
  xsetroot -solid "medium blue" &
  # rxvt는 메모리를 적게 쓰지만, 버그가 몇 가지 있다.
  #  - home, end 키가 인식되지 않는다.
  #  - backspace와 delete가 콘솔에서 처럼 작동하지 않는다.
  # - 환경이 색상을 제대로 상속받지 못한다.
  # - 일반적으로 환경에 문제가 있다.
  # 따라서 많은 경우에 xterm이 더 낫다. 하지만 mc 같은 몇 가지
  # 컬러 응용 프로그램을 수행시키기에는 rxvt가 최고다.
  xterm -ls -bg black -fg white -sb -sl 500 -j -ls -fn 10x20 \
    -fb 10x20bold -title "Color xterm" -geometry 80x25+150+0 &
  fvwm95-2
```

3.11 포트란

필자의 경험에 따르면, 포트란이 필요하다면 g77 대신 포트란-C 번역 프로그램인 f2c와 프런트-엔드인 yaf77을 쓰는 것이 좋다.

yaf77은 *ftp.unc.edu/pub/Linux/devel/languages/fortran/yaf77-1.4.tgz* `<ftp://ftp.unc.edu/pub/Linux/devel/languages/fortran/yaf77-1.4.tgz>`

와 그 미러 사이트에서 구할 수 있다.

3.12 개별 사용자를 위한 설정

새로운 사용자가 처음 로그인 했을 때를 위해 몇 가지 설정 파일들을 준비해 두는 것이 좋다. bashrc, bash_profile, bash_logout, inputrc, less, xinitrc, fvwmrc, fvwm2rc95, Xmodmap, Xdefaults, jedrc, joerc, emacs 파일들을 /ect/skel에 두도록 하자.

(주의: 포맷 문제 때문에 필자는 각 파일 이름 앞의 점('.')을 떼어냈다.) .pinerc는 완전히 편집할 수는 없다는 것에 주의해야 한다. 최소한 user-domain, smtp-server, nntp-server 란은 제대로 설정되었는지 확인해 두자.

제 4 절 끝

4.1 저작권

다른 언급이 없다면, 리눅스 HOWTO 문서들의 저작권은 각 문서의 저자들에게 있다. 리눅스 HOWTO는 이 저작권이 유지되는 한 모든 물리적 매체나 전자 매체를 통해 문서 전체 혹은 부분으로서 재생산 및 배포될 수 있다. 상업적인 재배포도 허용되며 장려된다. 하지만 이런 경우에는 저자에게 알려주기 바란다.

모든 번역이나 파생적인 작업들, 리눅스 HOWTO 문서들을 결합하는 작업 들은 이 저작권 안내에 따라야 한다. 이는 여러분이 HOWTO를 이용한 작업을 하고 나서 그 배포에 부가적인 제한조건을 달아서는 안된다는 뜻이 다. 어떤 경우에는 이 규칙에 대한 예외도 허용될 수 있으므로, 아래 주소 를 통해서 리눅스 HOWTO 조정자와 접촉하기 바란다.

간단히 말해서 우리는 이 정보를 가능한 한 많은 경로를 통해서 보급하기를 바란다. 하지만 우리는 HOWTO 문서에 대한 저작권이 유지되기를 바라며, HOWTO를 재배포하는 모든 계획을 통보받기를 바란다.

의문이 있다면 리눅스 HOWTO 조정자인 Greg Hankins와 아래 주소로 접 촉하기 바란다. *gregh@sunsite.unc.edu* `<mailto:gregh@sunsite.unc.edu>`

4.2 의견

다른 HOWTO들 못지 않게 이 글도 독자의 제안이나 비평, 기여를 환영한다. 독자의 의견은 환영 정도가 아니라 필수적이기까지 하다. 이 글에서 뭔가가 빠져있다고 생각한다면, 필자에게 전자메일을 보내주기 바란다. 레드햇 외의 배포본을 쓰고 있고 자신의 설정 파일이 이 글에서 설명한 것과 다르거나 다른 디렉터리에 위치하고 있다면, 그 팁을 포함시킬 수 있도록 필자에게 알려주기 바란다. 필자의 목적은 리눅스를 쓰는 것이 가능한 한 편해지도록 하는 것이다.

리눅스에는 엄청난 숫자의 패키지들이 있기 때문에, 그 패키지들을 위한 디렉터리를 모두 포함시키기란 불가능하다. 따라서 요청 사항이나 제안은 "가장 적절한" 프로그램들에 대해서만 해주기 바란다. 구체적인 판단은 각자의 상식에 맡긴다.

4.3 책임에 대한 포기

"Configuration HOWTO"는 Guido Conzato, *guido@ibogfs.cineca.it* `<mailto:guido@ibogfs.cineca.it>`에 의해 쓰여졌다. 필자가 뻔뻔스럽게 도용했던 모든 다른 HOWTO의 저자들과 매뉴얼 페이지의 필자 및 그 유지관리에 애쓰는 사람들에게 깊이 감사한다. 이 글은 "있는 바와 같이" 제공된다. 필자는 할 수 있는 한 정확하게 쓰려고 대단한 노력을 기울였지만, 여기 제시된 정보를 이용하는 것은 각자의 책임 하에 해야 한

다. 필자는 이 문서를 사용함으로써 생길 수 있는 일체의 손해에 대해서 어떤 경우에도 책임지지 않는다. 이 문서가 유용하길 바란다. 필자가 새로운 컴퓨터에 리눅스를 설치할 경우에는 늘 그랬다. 그럼 즐기기를, Guido =8-)