11.5. LFS 완료한 후

11.5.1. 다음에 수행할 작업 결정하기

이제 LFS가 완료되고 부팅 가능한 시스템이 생겼으니 이제 무엇을 해야 할까요? 다음 단계는 이 시스템을 어떻게 사용할지 결정하는 것입니다. 일반적으로 워크스테이션 또는 서버라는 두 가지 범주를 고려할 수 있습니다. 실제로 이러한 범주는 상호 배타적이지 않습니다. 각 카테고리에 필요한 애플리케이션을 하나의 시스템에 결합할 수 있지만 지금은 따로따로 살펴보기로 하겠습니다.

서버는 상대적으로 간단한 카테고리입니다. 일반적으로 Apache HTTP 서버와 같은 웹 서버와 MariaDB와 같은 데이터베이스 서버로 구성됩니다. 하지만 다른 서비스도 가능합니다. USB 드라이브 내장된 운영체제가 이 범주에 속합니다.

반면에 워크스테이션은 훨씬 더 복잡합니다. 일반적으로 기본 그래픽 환경을 기반으로 하는 LXDE, XFCE, KDE 또는 Gnome과 같은 그래픽 사용자 환경과 Firefox 웹 브라우저, Thunderbird 이메일 클라이언트 또는 LibreOffice 오피스 제품군과 같은 여러 그래픽 기반 애플리케이션이 필요합니다. 이러한 애플리케이션에는 원하는 기능에 따라 수백 개 이상의 지원 애플리케이션 및 라이브러리 패키지가 필요합니다.

위의 애플리케이션 외에도 모든 종류의 시스템에 대한 시스템 관리를 위한 애플리케이션 세트가 있습니다. 이러한 애플리케이션은 모두 BLFS 책에 포함되어 있습니다. 모든 환경에 모든 패키지가 필요한 것은 아닙니다. 예를 들어 dhcpcd는 일반적으로 서버에는 적합하지 않으며 wireless_tools는 일반적으로 노트북 시스템에만 유용합니다.

11.5.2. 기본 LFS 환경에서 작업하기

처음 LFS로 부팅하면 추가 패키지를 빌드할 수 있는 모든 내부 도구가 제공됩니다. 안타깝게도 사용자 환경은 매우 빈약합니다. 이를 개선할 수 있는 몇 가지 방법이 있습니다:

11.5.2.1. 루트에서 LFS 호스트에서 작업하기

이 방법은 모든 기능을 갖춘 브라우저와 복사/붙여넣기 기능을 사용할 수 있는 완전한 그래픽 환경을 제공합니다. 이 방법을 사용하면 호스트의 wget 버전과 같은 애플리케이션을 사용하여 루트 환경에서 작업할 때 사용할 수 있는 위치로 패키지 소스를 다운로드할 수 있습니다.

루트에서 패키지를 제대로 빌드하려면 가상 파일 시스템이 아직 마운트되지 않은 경우 마운트하는 것도 기억해야 합니다. 이를 위한 한 가지 방법은 호스트 시스템에서 스크립트를 만드는 것입니다:

```
cat > ~/mount-virt.sh << "EOF"
#!/bin/bash

function mountbind
{
   if ! mountpoint $LFS/$1 >/dev/null; then
     $SUDO mount --bind /$1 $LFS/$1
     echo $LFS/$1 mounted
   else
     echo $LFS/$1 already mounted
   fi
```

```
}

function mounttype
{
   if ! mountpoint $LFS/$1 >/dev/null; then
     $SUDO mount -t $2 $3 $4 $5 $LFS/$1
     echo $LFS/$1 mounted
   else
     echo $LFS/$1 already mounted
   fi
}

if [ $EUID -ne 0 ]; then
  SUDO=sudo
else
  SUDO=""
fi

if [ x$LFS == x ]; then
  echo "LFS not set"
  exit 1
fi

mountbind dev
mounttype dev/pts devpts devpts -o gid=5,mode=620
mounttype proc    proc   proc
mounttype sys     sysfs  sysfs
mounttype run     tmpfs  run
if [ -h $LFS/dev/shm ]; then
  install -v -d -m 1777 $LFS$(realpath /dev/shm)
else
  mounttype dev/shm tmpfs tmpfs -o nosuid,nodev
fi

#mountbind usr/src
#mountbind boot
#mountbind home
EOF
```

Note that the last three commands in the script are commented out. These are useful if those directories are mounted as separate partitions on the host system and will be mounted when booting the completed LFS/BLFS system.

The script can be run with bash ~/mount-virt.sh as either a regular user (recommended) or as root. If run as a regular user, sudo is required on the host system.

Another issue pointed out by the script is where to store downloaded package files. This location is arbitrary. It can be in a regular user's home directory such as ~/sources or in a global location like /usr/src. Our recommendation is not to mix BLFS sources and LFS sources in (from the chroot

environment) /sources. In any case, the packages must be accessible inside the chroot environment.

A last convenience feature presented here is to streamline the process of entering the chroot environment. This can be done with an alias placed in a user's ~/.bashrc file on the host system:

alias lfs='sudo /usr/sbin/chroot /mnt/lfs /usr/bin/env -i HOME=/root TERM="$TERM" PS1="\u:\w\\\$ " PATH=/bin:/usr/bin:/sbin:/usr/sbin /bin/bash –login' This alias is a little tricky because of the quoting and levels of backslash characters. It must be all on a single line. The above command has been split in two for presentation purposes.

11.5.2.2. Work remotely via ssh This method also provides a full graphical environment, but first requires installing sshd on the LFS system, usually in chroot. It also requires a second computer. This method has the advantage of being simple by not requiring the complexity of the chroot environment. It also uses your LFS built kernel for all additional packages and still provides a complete system for installing packages.

You may use the scp command to upload the package sources to be built onto the LFS system. If you want to download the sources onto the LFS system directly instead, install libtasn1, p11-kit, make-ca, and wget in chroot (or upload their sources using scp after booting the LFS system).

11.5.2.3. Work from the LFS command line This method requires installing libtasn1, p11-kit, make-ca, wget, gpm, and links (or lynx) in chroot and then rebooting into the new LFS system. At this point the default system has six virtual consoles. Switching consoles is as easy as using the Alt+Fx key combinations where Fx is between F1 and F6. The Alt+← and Alt+→ combinations also will change the console.

At this point you can log into two different virtual consoles and run the links or lynx browser in one console and bash in the other. GPM then allows copying commands from the browser with the left mouse button, switching consoles, and pasting into the other console.

[Note] Note As a side note, switching of virtual consoles can also be done from an X Window instance with the Ctrl+Alt+Fx key combination, but the mouse copy operation does not work between the graphical interface and a virtual console. You can return to the X Window display with the Ctrl+Alt+Fx combination, where Fx is usually F1 but may be F7.