# 2.2 호스트 시스템 준비 사항

## 2.2.1 하드웨어

LFS 시스템 CPU에 최소 4개의 코어가 있고 시스템에 최소 8GB의 메모리가 있는 것을 권장합니다. 이러한 요구 사항을 충족하지 않는 구형 시스템에서도 여전히 작동하지만 패키지를 빌드하는 시간이 문서에 기재된 것보다 훨씬 길어집니다.

## 2.2.2 소프트웨어

호스트 시스템에는 최소 버전이 표시된 다음 소프트웨어가 설치되어 있어야 합니다. 이는 대부분의 최신 리눅스 배포판에서는 문제가 되지 않습니다. 또한 많은 배포판에서 소프트웨어 헤더를 별도의 패키지에 배치하며, 종종 `<package-name>-devel` 또는 `<package-name>-dev`의 형태로 배포합니다. 배포판에서 이러한 패키지를 제공하는 경우 반드시 설치하세요.

- **Bash-3.2** (/bin/sh는 bash에 하드 링크 또는 심볼릭 링크되어 있어야 합니다.)
- **Binutils-2.13.1** (버전 2.42이상은 테스트 된것이 아니라서 권장하지 않습니다.)
- **Bison-2.7** (/usr/bin/yacc는 bison과 링크되어 있거나 bison을 실행하는 스크립트여야 합니다.)
- **Coreutils-8.1**
- **Diffutils-2.8.1**
- **Findutils-4.2.31**
- **Gawk-4.0.1** (/usr/bin/awk은 gawk와 링크되어 있어야 합니다.)
- **GCC-5.2**
  C++ 컴파일러 **g++**(13.2.0 이상 버전은 테스트되지 않았으므로 권장하지 않음)를 포함하고 있어야 합니다. C++ 컴파일러가 호스트된 프로그램을 빌드할 수 있도록 C 및 C++ 표준 라이브러리(헤더 포함)도 있어야 합니다.
- **Grep-2.5.1a**
- **Gzip-1.3.12**
- **Linux Kernel-4.19**
  특정 커널 버전을 요구하는 이유는 5장과 8장에서 glibc를 빌드할 때 해당 버전을 지정하기 때문에 이전 커널에 대한 해결 방법이 활성화되지 않고, 더불어 컴파일된 glibc가 약간 더 빠르고 작아지기 때문입니다.
  2024년 2월 현재, 4.19는 커널 개발자가 여전히 지원하는 가장 오래된 커널 릴리스입니다. 4.19보다 오래된 일부 커널 릴리스는 타사 팀에서 여전히 지원할 수 있지만 공식 업스트림 커널 릴리스로 간주되지 않으므로 자세한 내용은 커널 릴리즈를 참조하세요.
  호스트 커널이 4.19 이전 버전인 경우 커널을 최신 버전으로 교체해야 합니다. 이 작업을 수행하는 방법에는 두 가지가 있습니다. 먼저 리눅스 공급업체에서 4.19 이상의 커널 패키지를 제공하는지 확인합니다. 그렇다면 설치하는 것이 좋습니다. 공급업체에서 적합한 커널 패키지를 제공하지 않거나 설치하지 않으려는 경우에는 커널을 직접 컴파일할 수 있습니다. 커널 컴파일 및 부트로더 구성(호스트가 GRUB을 사용한다고 가정할 때)에 대한 지침은 10장에 나와 있습니다.
  호스트 커널이 UNIX 98 의사 터미널(PTY)을 지원해야 합니다. Linux 4.19 또는 최신 커널을 제공하

는 모든 데스크톱 또는 서버 배포판에서 이 기능을 사용하도록 설정해야 합니다. 사용자 지정 호스트 커널을 빌드하는 경우 커널 구성에서 CONFIG_UNIX98_PTYS가 y로 설정되어 있는지 확인하세요.

- **M4-1.4.10**
- **Make-4.0**
- **Patch-2.5.4**
- **Perl-5.8.8**
- **Python-3.4**
- **Sed-4.1.5**
- **Tar-1.22**
- **Texinfo-5.0**
- **Xz-5.0.0**

**중요**
위에서 언급한 심볼 링크는 이 책에 포함된 지침을 사용하여 LFS 시스템을 구축하는 데 필요합니다. 다른 소프트웨어(예: dash, mawk 등)를 가리키는 심볼릭 링크는 작동할 수 있지만 LFS 개발팀에서 테스트하거나 지원하지 않으며, 지침에서 벗어나거나 일부 패키지에 추가 패치가 필요할 수 있습니다. 목록에 있는 소프트웨어 패키지의 이전 버전은 작동할 수 있지만 테스트되지 않았습니다.

호스트 시스템에 적절한 버전이 모두 설치되어 있는지, 프로그램을 컴파일할 수 있는지 확인하려면 다음 명령을 실행하세요.

```
cat > version-check.sh << "EOF"
#!/bin/bash
# A script to list version numbers of critical development tools

# If you have tools installed in other directories, adjust PATH here AND
# in ~lfs/.bashrc (section 4.4) as well.

LC_ALL=C
PATH=/usr/bin:/bin

bail() { echo "FATAL: $1"; exit 1; }
grep --version > /dev/null 2> /dev/null || bail "grep does not work"
sed '' /dev/null || bail "sed does not work"
sort   /dev/null || bail "sort does not work"

ver_check()
{
   if ! type -p $2 &>/dev/null
   then
     echo "ERROR: Cannot find $2 ($1)"; return 1;
   fi
   v=$($2 --version 2>&1 | grep -E -o '[0-9]+\.[0-9\.]+[a-z]*' | head -n1)
   if printf '%s\n' $3 $v | sort --version-sort --check &>/dev/null
   then
```

```
     printf "OK:    %-9s %-6s >= $3\n" "$1" "$v"; return 0;
   else
     printf "ERROR: %-9s is TOO OLD ($3 or later required)\n" "$1";
     return 1;
   fi
}

ver_kernel()
{
   kver=$(uname -r | grep -E -o '^[0-9\.]+')
   if printf '%s\n' $1 $kver | sort --version-sort --check &>/dev/null
   then
     printf "OK:    Linux Kernel $kver >= $1\n"; return 0;
   else
     printf "ERROR: Linux Kernel ($kver) is TOO OLD ($1 or later
required)\n" "$kver";
     return 1;
   fi
}

# Coreutils first because --version-sort needs Coreutils >= 7.0
ver_check Coreutils      sort     8.1 || bail "Coreutils too old, stop"
ver_check Bash           bash     3.2
ver_check Binutils       ld       2.13.1
ver_check Bison          bison    2.7
ver_check Diffutils      diff     2.8.1
ver_check Findutils      find     4.2.31
ver_check Gawk           gawk     4.0.1
ver_check GCC            gcc      5.2
ver_check "GCC (C++)"    g++      5.2
ver_check Grep           grep     2.5.1a
ver_check Gzip           gzip     1.3.12
ver_check M4             m4       1.4.10
ver_check Make           make     4.0
ver_check Patch          patch    2.5.4
ver_check Perl           perl     5.8.8
ver_check Python         python3  3.4
ver_check Sed            sed      4.1.5
ver_check Tar            tar      1.22
ver_check Texinfo        texi2any 5.0
ver_check Xz             xz       5.0.0
ver_kernel 4.19

if mount | grep -q 'devpts on /dev/pts' && [ -e /dev/ptmx ]
then echo "OK:    Linux Kernel supports UNIX 98 PTY";
else echo "ERROR: Linux Kernel does NOT support UNIX 98 PTY"; fi

alias_check() {
   if $1 --version 2>&1 | grep -qi $2
   then printf "OK:    %-4s is $2\n" "$1";
   else printf "ERROR: %-4s is NOT $2\n" "$1"; fi
```

```
}
echo "Aliases:"
alias_check awk GNU
alias_check yacc Bison
alias_check sh Bash

echo "Compiler check:"
if printf "int main(){}" | g++ -x c++ -
then echo "OK:    g++ works";
else echo "ERROR: g++ does NOT work"; fi
rm -f a.out

if [ "$(nproc)" = "" ]; then
   echo "ERROR: nproc is not available or it produces empty output"
else
   echo "OK: nproc reports $(nproc) logical cores are available"
fi
EOF

bash version-check.sh
```

From:
https://www.gamu.kr/dokuwiki/ - **전자 수첩**

Permanent link:
**https://www.gamu.kr/dokuwiki/linuxfromscratch/12.1/016-host_system_requirements?rev=1718085309**

Last update: **2024/06/11 05:55**