

<b>Linux From Scratch - Version 12.1-systemd</b>		
<b>Chapter 2. Preparing the Host System</b>		
	/	
LFS		

## 2.2

### 2.2.1

LFS CPU 4 가 , 8GB 가 .

### 2.2.2

가  
 , <package-name>-devel <package-name>-dev .

- **Bash-3.2** (/bin/sh bash .)
- **Binutils-2.13.1** ( 2.42 .)
- **Bison-2.7** (/usr/bin/yacc bison bison .)
- **Coreutils-8.1**
- **Diffutils-2.8.1**
- **Findutils-4.2.31**
- **Gawk-4.0.1** (/usr/bin/awk gawk .)
- **GCC-5.2**  
 C++ g++(13.2.0 )  
 . C++ 가 C C++
- **Grep-2.5.1a**
- **Gzip-1.3.12**
- **Linux Kernel-4.19**

5 8 glibc  
 , glibc가  
 2024 2 , 4.19 가 가 . 4.19

4.19  
 가 가 . 4.19

( 가 GRUB 가 ) 10  
 UNIX 98 (PTY) . Linux 4.19

CONFIG\_UNIX98\_PTYS가 y

- **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**



LFS  
가 가

( :dash, mawk ) 가

LFS

```
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 ~/.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 T00 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 T00 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/> -

Permanent link:

[https://www.gamu.kr/dokuwiki/linuxfromscratch/12.1/016-host\\_system\\_requirements?rev=1718370430](https://www.gamu.kr/dokuwiki/linuxfromscratch/12.1/016-host_system_requirements?rev=1718370430)

Last update: **2024/06/14 13:07**

