

Bash

```

#   가
TARGET_ARCH=${LFS_TARGET:-x86_64-lfs-linux-gnu}
#
FILE="ffmpeg-8.0.1.tar.xz"
DIR_NAME=${FILE%.tar*} # DIR_NAME="ffmpeg-8.0.1"
#
SAFE_NAME=${RAW_NAME// /_} # " " -> "_"
#
# ##   가   "/"   .   가   "."
# %   "/"   .   가   "/"
PATH="/usr/lib/pkgconfig/sdl3.pc"
echo ${PATH##*/} #   /   -> sdl3.pc
echo ${PATH%/*} #   /   -> /usr/lib/pkgconfig
==== pkgconf   ====
Bash Script
<codeprism lang=bash>
### Check installed packages
TARGET=""

### merge args
for UARGS in "$@"; do
    TARGET="${TARGET}|${UARGS}"
done

### remove first '|'
TARGET="${TARGET#|}"

### '-print0' and 'xargs -0' for file name with white space
find /usr/lib/pkgconfig -name "*.pc" | \
    grep -Ei "${TARGET}" | \
    xargs -0 pkgconf --print-provides

```

Ruby Script

```

#!/usr/bin/env ruby

# 1.
target = ARGV.empty? ? "." : ARGV.join('|')

# 2.   가
search_path = "/usr/lib/pkgconfig/*.pc"

# .pc   (   )
package_names = Dir.glob(search_path).map do |file|
  File.basename(file, ".pc") # "/usr/lib/pkgconfig/zlib.pc" -> "zlib"
end.select do |pkg|

```

```

    pkg =~ Regexp.new(target, Regexp::IGNORECASE)
end

# 3. pkgconf
if package_names.any?
  #
  # : pkgconf --print-provides zlib libpng
  system("pkgconf", "--print-provides", *package_names)
else
  puts "          가          ."
end

```

Grub auto detect

AMD 5800X

```

sudo -s
cp /usr/lib/firmware/amd-ucode/microcode_amd_fam19h.bin /boot/amd-ucode.img
grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
Found theme: /usr/share/grub/themes/starfield/theme.txt
Found linux image: /boot/vmlinuz-6.18.10-1st
Found initrd image: /boot/amd-ucode.img
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done

```

Perl Module

Perl

```

#!/bin/sh

# Begin ~/bin/pml.sh

# List of installed Perl Modules

# grep "$pattern" - $1          (.),      $1
# /"Module"/ - "Module" 가
# !seen[$NF]++ - awk
# print $NF -          (      )
# sort --ignore-case (-f) |

pattern="${1:-.}"

```

```
perldoc perllocal | awk '/"Module"/ {if (!seen[$NF]++) print $NF}' | sort --
ignore-case | grep --color=always -E -i "$pattern"
```

```
# End ~/bin/pml.sh
```

```
pml.sh "tiny|test"
Capture::Tiny
Class::Tiny
Test::Deep
Test::Exception
Test::Fatal
Test::Needs
Test::Warnings
Try::Tiny
```

가 .

ExtUtils::Install module

```
#!/bin/sh

# Begin ~/bin/pml.sh

# List of installed Perl Modules

#      가
if [ -z "$1" ]; then
    echo "Usage: $(basename $0) [pattern]"
    echo "Example: $(basename $0) build"
    exit 1
fi

QUERY=$1

echo "---- Searching installed Perl modules for: $QUERY ----"

# ExtUtils::Installed
# grep -i

perl -MExtUtils::Installed -e '
    my $inst = ExtUtils::Installed->new();
    foreach my $mod ($inst->modules()) {
        my $ver = $inst->version($mod) || "N/A";
        print "$mod | Version: $ver\n";
    }
' | grep --color=auto -E -i "$QUERY"

#      가
```

```
if [ ${PIPESTATUS[1]} -ne 0 ]; then
    echo "Result: No Matching Modules found."
fi

# End ~/bin/pml.sh
```

```
pml.sh text
---- Searching installed Perl modules for: text ----
Text::BibTeX | Version: 0.91
Text::Diff | Version: 1.45
Text::Glob | Version: 0.11
```

Python3 Module

```
pip3 list
```

Qemu Screen Dump

1. <Ctrl>+<Alt>+<2>
 <Ctrl>+<A>, <C> -nographic
2. screendump filename.ppm
 -f 가
 screendump screenshot.png -f png ppm
3. <Ctrl>+<Alt>+<1> 가

xwindow screen saver & dpms

```
xset s 600 600
xset dpms 600 600 600
```

DPMS , , ,

for, echo and sed

```
for i in $(echo "ldapadd, ldapcompare, ldapdelete, ldapexop, ldapmodify,
slapschema, slaptest" | sed 's/,/ /g') ; do find /usr/bin -type f -name $i ;
done
```

Grep & Tar

가

“python3_11” \$HOME

```
grep -rل python3_11 | tar -T - -acf ~/ibus-hangul.tar.zst
```

grep

- r
- l

tar

- T
- -
- a
- c
- f

Configure Options #1

Fail

가

-docdir=/usr/share/doc/<package name>

\$HOME/.bashrc 가

```
if [ -d $HOME/.bashrc.d ] ; then
  for conf in $HOME/.bashrc.d/*.conf ; do
    source $conf
  done
fi
```

```
cat > $HOME/.bashrc.d/config_opts.conf << EOF
alias doc-dir='echo "--prefix=/usr --enable-shared --disable-static --
```


```
sysconfdir=/etc --docdir=/usr/share/doc/${(basename $(pwd))} --enable-
dependency-tracking" '
EOF
```

```
./configure $(doc-dir)
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
... ..
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating Makefile
config.status: creating src/Makefile
config.status: creating tests/Makefile
config.status: creating config/Makefile
config.status: creating config/gdlib.pc
config.status: creating src/config.h
config.status: executing depfiles commands
config.status: executing libtool commands
make
sudo make install
```

Find script

가

: [ANSI Code Generator](#)

 git clone <https://github.com/fidian/ansi.git>

```
baecy-lfs@m5pro-lfs < ~ > $
double-check libGLEW
/usr/lib/libGLEW.so
/usr/lib/libGLEW.so.2.2
/usr/lib/libGLEW.so.2.2.0
I searched for libGLEW you wanted in th /usr you specified
baecy-lfs@m5pro-lfs < ~ > $
```

```
function file-finder() {
  local FindWord=${1}
  local FindDir=$2
  if [ -z $FindDir ] ; then
    FindDir=/usr
```

```

    fi
    find $FindDir -name $FindWord 2> /dev/null | sort | grep -E --
color=always $1
    echo "Found file $(ansi --green ${1}) in directory $(ansi --yellow
${FindDir}) (refer to the list above)."
}
export -f file-finder

```

Patch with Wget

```
wget -q -O- <patch url> | patch <options>
```

1

docbook-xsl-nons-1.79.2 patch

```
wget -q -O-
https://www.linuxfromscratch.org/patches/blfs/12.1/docbook-xsl-nons-1.79.2-s
tack_fix-1.patch \
| patch -Np1

```

2

libpng patch

```
wget -O-
https://downloads.sourceforge.net/sourceforge/libpng-apng/libpng-1.6.40-apng
.patch.gz \
| gzip -cd | patch -p1

```

PS1

```

\[\033[01;34m\]\u\[\033[00m\]@\[\033[01;36m\]\h
\[\033[01;34m\]\w\[\033[00m\] \$

```

```
\[\033[01;35m\]\u\[\033[00m\]@\[\033[01;35m\]\h \[\033[01;37m\]\w
\[\033[01;31m\]# \[\033[00m\]
```

2

```

PS1_ROOT='\[$(tput setaf 203)\][ \[$(tput sgr0)\]\w \[$(tput setaf 203)\]]
\[$(tput sgr0)\]\$\n'
PS1_USER='\[$(tput setaf 75)\][ \[$(tput sgr0)\]\w \[$(tput setaf 75)\]]
\[$(tput sgr0)\]\$\n'
PS1_HEAD='\[$(tput setaf 229)\]\u\[$(tput setaf 199)\]@\[$(tput setaf
215)\]\h '
if [ "$(awk '$5=="/" {print $1}' </proc/1/mountinfo)" != "$(awk '$5=="/"
{print $1}' </proc/$$/mountinfo)" ] ; then
    if [ $(id -u) -eq 0 ] ; then
        PS1="<Chroot>${PS1_HEAD}${PS1_ROOT}"
    else
        PS1="<Chroot>${PS1_HEAD}${PS1_USER}"
    fi
else
    if [ $(id -u) -eq 0 ] ; then
        PS1="${PS1_HEAD}${PS1_ROOT}"
    else
        PS1="${PS1_HEAD}${PS1_USER}"
    fi
fi

```

1

```

# chroot
if [ "$(awk '$5=="/" {print $1}' </proc/1/mountinfo)" != "$(awk '$5=="/"
{print $1}' </proc/$$/mountinfo)" ] ; then
    PS1='<Chroot> \[$(tput setaf 229)\]\u\[$(tput setaf 199)\]@\[$(tput setaf
215)\]\h \[$(tput setaf 203)\][ \[$(tput sgr0)\]\w \[$(tput setaf 203)\]]
\[$(tput sgr0)\]\$\n'
else
# root
    if [ $(id -u) -eq 0 ] ; then
        PS1='\[$(tput setaf 229)\]\u\[$(tput setaf 199)\]@\[$(tput
setaf 215)\]\h \[$(tput setaf 203)\][ \[$(tput sgr0)\]\w \[$(tput setaf
203)\]] \[$(tput sgr0)\]\$\n'
    else

```

```

                PS1='\[$(tput setaf 229)\]\u\[$(tput setaf 199)\]@\[$(tput
setaf 215)\]\h \[$(tput setaf 75)\][ \[$(tput sgr0)\]\w \[$(tput setaf
75)\]] \[$(tput sgr0)\]\$\n'
                fi
fi

```

```
touch $LFS/etc/lfs-chroot
```

```
~/bashrc
```

```

if [ -f /etc/lfs-chroot ]; then
    PS1='\[$(tput setaf 75)\]<LFS> \[$(tput setaf 229)\]\u\[$(tput setaf
199)\]@\[$(tput setaf 215)\]\h \[$(tput setaf 75)\]\w \[$(tput sgr0)\]\$\n'
else
    PS1='\[$(tput setaf 75)\]\[$(tput setaf 229)\]\u\[$(tput setaf
199)\]@\[$(tput setaf 215)\]\h \[$(tput setaf 75)\]\w \[$(tput sgr0)\]\$\n'
fi

```

Debian 가

LFS

```

sudo apt install build-essentials bison gawk m4 texinfo texinfo ##
sudo apt install gettext libisl-dev ##
sudo ln -sf bash /usr/bin/sh && file /usr/bin/sh ##
/usr/bin/sh: symbolic link to bash

```

```
time { command 1 && ... && command N; } 2>&1 | tee <log-file>
```

Qemu

가

Qemu 가

가

```
mkswap -L <label> <partition>
e2label <device> <label>
```

/etc/fstab

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system>          <mount point>  <type>          <options>
<dump> <pass>
# / was on /dev/sda1 during installation
LABEL=ROOT_DISK          /                ext4             errors=remount-ro
0 1
# swap was on /dev/sda5 during installation
LABEL=SWAP_1G            none            swap            sw
0 0
LABEL=SWAP_5G            none            swap            sw
0 0
# /dev/sr0
0 0
LABEL=LFS_DISK           /mnt/lfs        ext4             defaults
1 1
```

UUID

Putty Ncurses <Home>,<End> Key

~/.profile or ~/.bashrc

```
TERM=putty-256color
export TERM
```


NFS-UTILS

NFS-Utills-2.6.4

- [libtirpc-1.3.4](#) REQ
- [libevent-2.1.12](#) REQ
- [rpcsvc-proto-1.4.4](#) REQ
- [SQLite-3.45.1](#) REQ
- [rpcbind-1.2.6](#) REQ

SSHFS (NFS)

N40L

sshfs-3.7.3

: ICU - Libxml2 - Sgml-common - UnZip - Docbook-xsl-nons - Docbook-xml - Libxslt - Docutil - Packing - PCRE2 - Glib - OpenSSH - Fuse - SSHFS

- [Fuse-3.16-1](#) REQ
- [Glib-2.80.0](#) REQ
 - [Packging \(Python module\)](#) REQ
 - [Docutils \(Python module\)](#) REC
 - [Libxslt-1.1.39](#) REC
 - [Libxml2-2.12.6](#) REQ
 - [ICU-75.1](#) REC
 - [Docbook-xml-4.5](#) Runtime REC
 - [Libxml2-2.12.6](#) REQ
 - [Sgml-common-0.6.3](#) REQ
 - [UnZip-6.0](#) REQ
 - [Docbook-xsl-nons-1.79.2](#) Runtime REC
 - [Libxml2-2.12.6](#) REQ
 - [Pcre2-10.43](#) REC
- [OpenSSH 9.6p1](#) REQ

Textinfo Dir File Rebuild

```
pushd /usr/share/info
  rm -v dir
  for f in *
```

```
do install-info $f dir 2>/dev/null
done
popd
```

Swap file

```
export LFS=/mnt/lfs
export LFS_SWAP=$LFS/Swapfile_For_LFS_1G
sudo fallocate --length 1G $LFS_SWAP
sudo chmod 0600 $LFS_SWAP
sudo mkswap $LFS_SWAP
Setting up swap space version 1, size = 1024 MiB (1073737728 bytes)
no label, UUID=890ba9a5-da48-4374-84ce-b71b91863e00
sudo swapon $LFS_SWAP
sudo echo "$LFS_SWAP swap swap defaults 0 0" >> /etc/fstab
```

Binutils

ISL

GMP

```
./configure --prefix=/usr \
            --enable-cxx \
            --disable-static \
            --docdir=/usr/share/doc/gmp-6.2.0
make
make check 2>&1 | tee gmp-check.log
awk '/# PASS:/{total+=$3} ; END{print total}' gmp-check-log
```

[ISL Homepage](#)

```
./configure --prefix=/usr --disable-static --with-gmp=system
make
make install
```

\$LFS/sources

가

```
ls -d */
```

```
alias dirfind="find -mindepth 1 -maxdepth 1 -type d | sed 's@^./@@'"
dirfind | wc -l    ##
rm -rf $(dirfind) ##
##
alias SearchAndDestroy='find -mindepth 1 -maxdepth 1 -type d -exec rm -rf {}
\;'
```

SBU

```
##          +          가
alias lfslog='tee ~/lfs-log/${(case $(basename $(pwd)) in build) echo
$(basename $(dirname $(pwd))); ;; *) echo $(basename $(pwd)); ;;
esac).$(date "+%Y%m%d_%H%M%S").log'
time { ./configure ... && make && make install; } | lfslog
```

```
<a href=""></a> ##
<a target="_blank" rel="noopener noreferrer" href=""></a> ##
```

```
./configure --prefix=/usr          \
            --bindir=/usr/bin      \
            --localstatedir=/var   \
            --disable-logger       \
            --disable-whois        \
            --disable-rcp          \
            --disable-rexec        \
            --disable-rlogin       \
            --disable-rsh          \
            --disable-servers

## Same Command

./configure --prefix=/usr --bindir=/usr/bin --localstatedir=/var \
            --disable-{logger,whois,r{cp,exec,login,sh},servers}
```

tar

```
tar -cf - . | tar -xvf - -C /target_directory
```

find example

```
##      가 50M      .deb, .vmdk      가
## Operator AND = -and, -a, OR= -or = -o, NOT = !
find /media/d/ -type f -size +50M -and ! -name "*deb" -and ! -name "*vmdk"
```

grep example

```
##      grep      가      .
$ grep 'printf' fileio.c
    printf("FILE open Error\n");
    printf("chi = %c\n", chi);
    printf("cho = %c\n", cho);
    printf("FILE open Error\n");
    printf("chi = %c\n", chi);
    printf("cho = %c\n", cho);

## -n      line      .
$ grep -n 'printf' fileio.c
11:      printf("FILE open Error\n");
18:      printf("chi = %c\n", chi);
19:      printf("cho = %c\n", cho);
25:      printf("FILE open Error\n");
31:      printf("chi = %c\n", chi);
32:      printf("cho = %c\n", cho);

## -v      'chi'      print      .
$ grep -n 'printf' fileio.c | grep -v 'chi'
11:      printf("FILE open Error\n");
19:      printf("cho = %c\n", cho);
25:      printf("FILE open Error\n");
32:      printf("cho = %c\n", cho);

## grep      print      .
$ grep -n 'printf' fileio.c | grep -v 'chi' | grep -v 'cho'
11:      printf("FILE open Error\n");
25:      printf("FILE open Error\n");

## -Ev      grep      (|)      .
$ grep -n 'printf' fileio.c | grep -Ev 'chi|cho'
11:      printf("FILE open Error\n");
```

```
25:          printf("FILE open Error\n");
```

```
grep -rل '#!.*python' | xargs sed -i 'ls/python$/python3/'

//          1  #!          python          python
python3
#!/usr/bin/env python  #
#!/usr/bin/env python3  #
```

Gawk

```
df -ht ext4
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        40G   11G   27G   29% /
/dev/sde1        173M   95M   65M   60% /boot
/dev/sde2         19G   1.5G   16G    9% /mnt/debian
/dev/sda1        1.8T   9.7G   1.7T    1% /mnt/1st-bay
/dev/sdb1        1.8T   1.1G   1.7T    1% /mnt/2nd-bay
/dev/sdc1        1.8T    28K   1.7T    1% /mnt/3rd-bay
/dev/sdd1        1.8T    32K   1.7T    1% /mnt/4th-bay
```

```
#          NR          (          Filesystem  Size)          가 2          awk

#          sort
df -ht ext4 | awk 'NR<2{print $0;next}{print $0| "sort"}'
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        40G   11G   27G   29% /
/dev/sda1        1.8T   9.7G   1.7T    1% /mnt/1st-bay
/dev/sdb1        1.8T   1.1G   1.7T    1% /mnt/2nd-bay
/dev/sdc1        1.8T    28K   1.7T    1% /mnt/3rd-bay
/dev/sdd1        1.8T    32K   1.7T    1% /mnt/4th-bay
/dev/sde1        173M   95M   65M   60% /boot
/dev/sde2         19G   1.5G   16G    9% /mnt/debian
```

```
# lq          sed          sort가
df -ht ext4 | { sed -u lq; sort; }

Filesystem      Size  Used Avail Use% Mounted on
/dev/root        40G   11G   27G   29% /
```

```
/dev/sda1      1.8T  9.7G  1.7T   1% /mnt/1st-bay
/dev/sdb1      1.8T  1.1G  1.7T   1% /mnt/2nd-bay
/dev/sdc1      1.8T   28K  1.7T   1% /mnt/3rd-bay
/dev/sdd1      1.8T   32K  1.7T   1% /mnt/4th-bay
/dev/sde1      173M  95M   65M  60% /boot
/dev/sde2       19G  1.5G   16G   9% /mnt/debian
```

wget

```
# -r / --recursive :
# -np / --no-parent :
# -R / --reject :
# -P / --directory-prefix=PREFIX : PREFIX/..
wget -r -np -R "index.html*" -P <DIR> lfs/
https://ftp.osuosl.org/pub/lfs/lfs-packages/12.1/
```

Firmware Blob Kernel config

```
echo CONFIG_EXTRA_FIRMWARE="'${ cd /lib/firmware; echo amd-ucode/*; echo
amdgpu/re*; echo rtl_nic/*; echo regulatory*; }'" >> .config
make oldconfig
```

Man page count

```
find /usr/share/man/ -type f -name '*. [0-9]' | wc -l
```

From:

<https://www.gamu.kr/dokuwiki/> -

Permanent link:

<https://www.gamu.kr/dokuwiki/linuxfromscratch?rev=1773413908>

Last update: **2026/03/13 14:58**

