

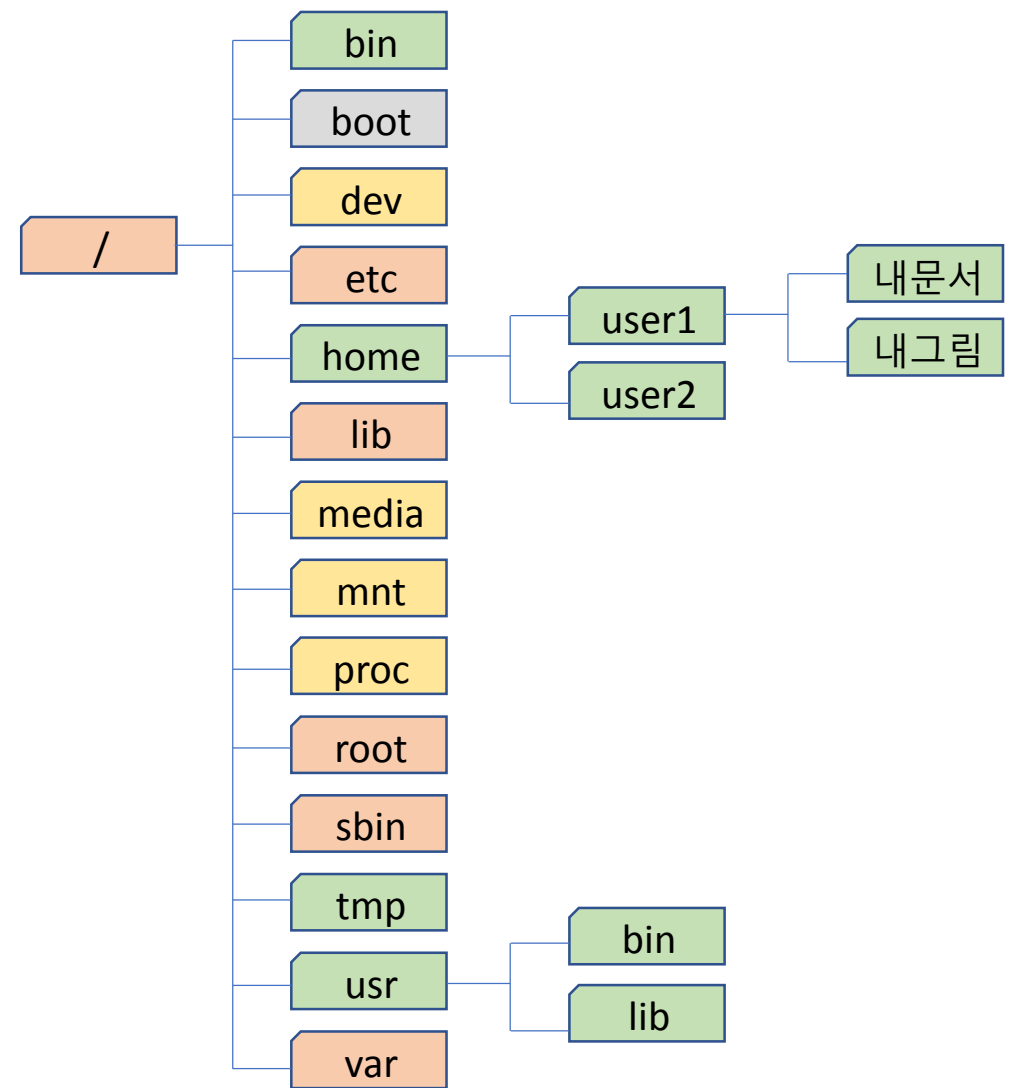
Chapter 02. 파일시스템 구조

# 파일시스템 살펴보기

## 배울 명령어 - 파일시스템 관련

- 디렉토리 구조 복습
- 파일 시스템 검색
  - find
- 파일 속성 보기
  - stat
- 파일 내용 검색 / 필터링 / 변경 등의 활용을 위한 유틸리티
  - grep
  - sort
  - awk
  - sed
  - uniq
  - wc
- 파일 시스템 용량
  - du
- 압축 등 관리
  - tar
  - gz
- 쉘 스크립트 맛보기

파일시스템 살펴보기 - 복습



FHF (Filesystem Hierarchy Standard) 표준 파일 시스템

/ : 루트 디렉토리  
/bin : 기본적인 명령어  
/boot : 부트로더 디렉토리  
/dev : 시스템 디바이스 (장치파일)  
**/etc : 각종 설정파일**  
/home : 사용자의 홈 디렉토리  
/lib : 라이브러리 (및 커널모듈)  
/media : 외부 저장소 (cdrom, usb 등)  
/mnt : 외부 저장소 또는 파일시스템 마운트포인트  
/proc : 시스템 설정들의 가상파일시스템 (pseudo-file 시스템)  
/root : 루트 사용자의 홈 디렉토리  
/sbin : 시스템 (관리자용) 명령어  
/tmp : 임시 저장소  
/usr : 일반 사용자들 공통파일  
**/var : 시스템 운용중에 생성되는 임시 데이터 저장소**

## 파일 시스템 주요 디렉토리 - /boot

grub 및 커널, initrd, 멀티 부팅 시 메모리 테스트를 위한 memtest 도구

```
user1@user1-VirtualBox:~$ ls -al /boot
합계 200796
drwxr-xr-x  3 root root    4096 5월  3 22:40 .
drwxr-xr-x 24 root root    4096 5월  3 22:40 ..
-rw-r--r--  1 root root 4048670 1월 30 2019 System.map-4.15.0-45-generic
-rw-r--r--  1 root root 4068326 4월  1 14:44 System.map-4.15.0-96-generic
-rw-r--r--  1 root root 4069610 4월 23 11:36 System.map-4.15.0-99-generic
-rw-r--r--  1 root root  217027 1월 30 2019 config-4.15.0-45-generic
-rw-r--r--  1 root root  217465 4월  1 14:44 config-4.15.0-96-generic
-rw-r--r--  1 root root  217466 4월 23 11:36 config-4.15.0-99-generic
drwxr-xr-x  5 root root    4096 5월  3 22:41 grub
-rw-r--r--  1 root root 55858415 4월  7 21:48 initrd.img-4.15.0-45-generic
-rw-r--r--  1 root root 55895985 4월  7 21:48 initrd.img-4.15.0-96-generic
-rw-r--r--  1 root root 55900177 5월  3 22:40 initrd.img-4.15.0-99-generic
-rw-r--r--  1 root root  182704 1월 28 2016 memtest86+.bin
-rw-r--r--  1 root root  184380 1월 28 2016 memtest86+.elf
-rw-r--r--  1 root root  184840 1월 28 2016 memtest86+_multiboot.bin
-rw-r--r--  1 root root 8128088 2월 27 2019 vmlinuz-4.15.0-45-generic
-rw-r--r--  1 root root 8191544 4월  7 01:08 vmlinuz-4.15.0-96-generic
-rw-r--r--  1 root root 8202744 4월 29 03:14 vmlinuz-4.15.0-99-generic
user1@user1-VirtualBox:~$
```

## 파일 시스템 주요 디렉토리 - /home

사용자 디렉토리 및 슈퍼유저(root)의 홈 디렉토리 /root

```
user1@user1-VirtualBox:~$ ls -l /home
합계 8
drwxr-xr-x 26 user1 user1 4096 5월 5 01:26 user1
drwxr-xr-x 2 user2 user2 4096 5월 5 01:04 user2
user1@user1-VirtualBox:~$ ls -l /root
ls: '/root' 디렉터를 열 수 없음: 허가 거부
user1@user1-VirtualBox:~$ sudo ls -l /root
합계 0
user1@user1-VirtualBox:~$ sudo ls -al /root
합계 32
drwx----- 4 root root 4096 5월 3 22:30 .
drwxr-xr-x 24 root root 4096 5월 3 22:40 ..
-rw----- 1 root root 31 5월 3 22:55 .bash_history
-rw-r--r-- 1 root root 3106 10월 23 2015 .bashrc
drwx----- 2 root root 4096 2월 27 2019 .cache
drwxr-xr-x 2 root root 4096 4월 13 23:02 .nano
-rw-r--r-- 1 root root 148 8월 18 2015 .profile
-rw-r----- 1 root root 5 5월 3 22:30 .vboxclient-display-svga.pid
user1@user1-VirtualBox:~$
```

# 파일 시스템 주요 디렉토리 - /etc

## 시스템 프로세스의 각종 설명파일들

```

user1@user1-VirtualBox:~$ ls -l /etc
합계 1172
drwxr-xr-x  2 root root    4096 4월  7 21:47 ImageMagick-6
drwxr-xr-x  8 root root    4096 2월 27 2019 NetworkManager
drwxr-xr-x  2 root root    4096 2월 27 2019 UPower
drwxr-xr-x 11 root root    4096 2월 27 2019 X11
drwxr-xr-x  3 root root    4096 2월 27 2019 acpi
-rw-r--r--  1 root root   3028 2월 27 2019 adduser.conf
drwxr-xr-x  2 root root    4096 5월  4 10:29 alternatives
-rw-r--r--  1 root root    401 12월 29 2014 anacrontab
-rw-r--r--  1 root root    112 1월 10 2014 apg.conf
drwxr-xr-x  6 root root    4096 2월 27 2019 apm
drwxr-xr-x  3 root root    4096 4월  7 21:47 apparmor
drwxr-xr-x  8 root root    4096 5월  3 22:40 apparmor
drwxr-xr-x  5 root root    4096 4월  7 21:47 apparmor
-rw-r--r--  1 root root    389 4월 18 2016 appstream
drwxr-xr-x  6 root root    4096 4월 18 01:20 apt
drwxr-xr-x  3 root root    4096 2월 27 2019 aptdaemon
drwxr-xr-x  2 root root    4096 2월 27 2019 at-spi
drwxr-xr-x  3 root root    4096 2월 27 2019 avahi
-rw-r--r--  1 root root   2189 5월  5 00:58 bash.bashrc
-rw-r--r--  1 root root    45 8월 13 2015 bash_completion
drwxr-xr-x  2 root root    4096 4월  7 21:48 modprobe.d
-rw-r--r--  1 root root    195 2월 27 2019 modules
drwxr-xr-x  2 root root    4096 4월  7 21:47 modules-load.d
lrwxrwxrwx  1 root root     19 4월  5 18:43 mtab -> ../proc/self/mounts
-rw-r--r--  1 root root    624 8월  8 2007 mtools.conf
-rw-r--r--  1 root root   8338 2월 15 2017 nanorc
drwxr-xr-x  7 root root    4096 4월 13 22:38 network
-rw-r--r--  1 root root     91 10월 23 2015 networks
drwxr-xr-x  2 root root    4096 2월 27 2019 newt
drwxr-xr-x  6 root root    4096 5월  4 00:01 nginx
-rw-r--r--  1 root root    529 2월 27 2019 nsswitch.conf
drwxr-xr-x  2 root root    4096 2월 27 2019 opt
lrwxrwxrwx  1 root root     21 4월  5 18:31 os-release -> ../usr/lib/os-release
-rw-r--r--  1 root root    552 3월 17 2016 pam.conf
drwxr-xr-x  2 root root    4096 5월  3 22:40 pam.d
-rw-rw-r--  1 root root     3 4월  5 18:43 papersize
-rw-r--r--  1 root root   2385 4월 25 11:03 passwd
-rw-r--r--  1 root root   2417 4월 25 11:03 passwd-
drwxr-xr-x  2 root root    4096 2월 27 2019 pcmcia
drwxr-xr-x  4 root root    4096 2월 27 2019 perl
drwxr-xr-x  4 root root    4096 2월 27 2019 pki
drwxr-xr-x  5 root root    4096 2월 27 2019 pm
-rw-r--r--  1 root root   7649 2월 27 2019 pnm2ppa.conf

```

## 파일 시스템 주요 디렉토리 - /etc/\*-release

운영체제 정보를 갖고 있는 파일들

cat /etc/\*-release

```
user1@user1-VirtualBox:~$ ls -l /etc/*-release
-rw-r--r-- 1 root root 105 2월 20 2019 /etc/lsb-release
lrwxrwxrwx 1 root root 21 4월 5 18:31 /etc/os-release -> ../usr/lib/os-release
user1@user1-VirtualBox:~$ cat /etc/*-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=16.04
DISTRIB_CODENAME=xenial
DISTRIB_DESCRIPTION="Ubuntu 16.04.6 LTS"
NAME="Ubuntu"
VERSION="16.04.6 LTS (Xenial Xerus)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 16.04.6 LTS"
VERSION_ID="16.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
VERSION_CODENAME=xenial
UBUNTU_CODENAME=xenial
user1@user1-VirtualBox:~$
```

배포판에 따라...

- os-release
- centos-release
- redhat-release

# 파일 시스템 주요 디렉토리 - /var 및 /var/log

## 시스템 프로세스의 각종 임시 파일들 및 로그 파일들

```
user1@user1-VirtualBox:~$ ls -l /var
합계 52
drwxr-xr-x 2 root root 4096 5월 4 07:35 backups
drwxr-xr-x 16 root root 4096 4월 5 18:44 cache
drwxrwsrwt 2 root whoopsie 4096 4월 26 07:35 crash
drwxr-xr-x 71 root root 4096 5월 4 00:01 lib
drwxrwsr-x 2 root staff 4096 4월 13 2016 local
lrwxrwxrwx 1 root root 9 4월 5 18:34 lock -> /run/lock
drwxrwxr-x 13 root syslog 4096 5월 4 07:35 log
drwxrwsr-x 2 root mail 4096 2월 27 2019 mail
drwxrwsrwt 2 root whoopsie 4096 2월 27 2019 metrics
drwxr-xr-x 2 root root 4096 2월 27 2019 opt
lrwxrwxrwx 1 root root 4 4월 5 18:34 run -> /run
drwxr-xr-x 2 root root 4096 1월 30 2019 snap
drwxr-xr-x 7 root root 4096 2월 27 2019 spool
drwxrwsrwt 5 root root 4096 5월 5 00:58 tmp
drwxr-xr-x 3 root root 4096 5월 4 00:01 www
```

```
user1@user1-VirtualBox:~$ ls -l /var/log
합계 3912
-rw-r--r-- 1 root root 23559 5월 3 22:30 Xorg.0.log
-rw-r--r-- 1 root root 24399 4월 27 23:26 Xorg.0.log.old
-rw-r--r-- 1 root root 257 5월 4 10:29 alternatives.log
-rw-r--r-- 1 root root 44185 4월 23 00:25 alternatives.log.1
-rw-r----- 1 root adm 0 4월 19 21:04 apport.log
-rw-r----- 1 root adm 518 4월 18 01:23 apport.log.1
-rw-r----- 1 root adm 393 4월 8 00:32 apport.log.2.gz
drwxr-xr-x 2 root 4096 5월 3 22:34 apt
-rw-r----- 1 syslog adm 28797 5월 5 01:47 auth.log
-rw-r----- 1 syslog adm 17966 5월 3 22:34 auth.log.1
-rw-r----- 1 syslog adm 8163 4월 27 07:30 auth.log.2.gz
-rw-r----- 1 syslog adm 5176 4월 19 21:04 auth.log.3.gz
-rw-r----- 1 syslog adm 218130 5월 3 22:33 kern.log.1
-rw-r----- 1 syslog adm 110462 4월 27 07:34 kern.log.2.gz
-rw-r----- 1 syslog adm 93577 4월 19 21:04 kern.log.3.gz
-rw-r----- 1 syslog adm 218751 4월 13 07:31 kern.log.4.gz
-rw-rw-r-- 1 root utmp 292876 4월 25 11:03 lastlog
drwxr-xr-x 2 root root 4096 5월 4 07:35 lightdm
drwxr-xr-x 2 root adm 4096 5월 4 00:01 nginx
drwx----- 2 speech-dispatcher root 4096 2월 19 2016 speech-dispatcher
-rw-r----- 1 syslog adm 396339 5월 5 01:49 syslog
-rw-r----- 1 syslog adm 209479 5월 4 07:35 syslog.1
-rw-r----- 1 syslog adm 53745 5월 3 22:34 syslog.2.gz
-rw-r----- 1 syslog adm 28147 4월 27 07:35 syslog.3.gz
-rw-r----- 1 syslog adm 28044 4월 26 07:35 syslog.4.gz
-rw-r----- 1 syslog adm 28733 4월 25 07:35 syslog.5.gz
-rw-r----- 1 syslog adm 28075 4월 24 07:35 syslog.6.gz
-rw-r----- 1 syslog adm 76937 4월 23 07:35 syslog.7.gz
drwxr-xr-x 2 root adm 4096 5월 3 22:34 unattended-upgrades
drwxr-xr-x 2 root root 4096 5월 19 2016 upstart
-rw-r--r-- 1 root root 631 4월 5 23:49 vboxadd-install.log
-rw-r--r-- 1 root root 157 5월 3 22:41 vboxadd-setup.log
-rw-r--r-- 1 root root 63 5월 3 22:30 vboxadd-setup.log.1
-rw-r--r-- 1 root root 63 4월 22 23:41 vboxadd-setup.log.2
-rw-r--r-- 1 root root 63 4월 22 23:37 vboxadd-setup.log.3
-rw-r--r-- 1 root root 63 4월 19 21:00 vboxadd-setup.log.4
```



## 파일 시스템 명령어 - 검색 (find)

find [OPTIONS] [PATH] [EXPRESSION]

원하는 파일의 검색

내 현재 디렉토리에서 확장자가 .txt 인 파일을 찾는 법

- find \*.txt

내 현재 디렉토리에서부터 확장자가 .txt 인 파일을 찾는 법

- find . -name "\*.txt"

내 현재 디렉토리에서 파일명이 hello로 시작하는 "파일" 만 찾는 법

- find . -name "hello\*" -type f

내 현재 디렉토리에서 디렉토리명이 dir 로 시작하는 "디렉토리" 만 찾는 법

- find . -name "dir\*" -type d

내 현재 디렉토리에서 용량이 100,000,000바이트 (100M) 보다 큰 파일 찾는 법

- find . -size +1000000000c <-- 여기 1000000000c 대신 100000k 또는 100M 사용 가능

최근 생성된 파일만 찾아보기 (2020년 5월 15일 이후 변경된 파일)

- find -newerct "15 May 2020" -ls <-- 여기 newerct 대신 newermt 로 할 경우 최근 변경된

최근 2일에서 5일 사이에 변경된 파일 찾기

- find . -mtime +2 -a -mtime -5 -ls

```
user1@user1-VirtualBox:~$ find *.txt
file.txt
hello.txt
readonly.txt
result.txt
user1@user1-VirtualBox:~$ find . -name "dir*" -type d
./dir1
./dir3
./dir2
user1@user1-VirtualBox:~$ find . -size +100M
user1@user1-VirtualBox:~$ find . -size +50M
./다운로드/google-chrome-stable_current_amd64.deb
user1@user1-VirtualBox:~$
```

## 파일 시스템 명령어 - 속성 (stat)

stat [OPTIONS] [FILE]

원하는 파일의 속성 (주로 시간) 확인

man 8 mount

- relatime (우분투의 기본값)

시간의 유형 (atime, mtime, ctime)

- **Access** : 파일에 최근 **접근 시간** (고전적으로는 read 시에도 올라갔으나 지금은 다소 변경됨)
- **Modify** : 파일의 내용 **변경 시간**
- **Change** : 파일의 **수정 시간** (inode 관점에서의 변화 시간 - 생성, 변경, 속성수정 등)

```
user1@user1-VirtualBox:~$ stat hello.txt
  File: 'hello.txt'
  Size: 12          Blocks: 8          IO Block: 4096   일반 파일
Device: 801h/2049d Inode: 521713       Links: 2
Access: (0664/-rw-rw-r--)  Uid: ( 1000/   user1)   Gid: ( 1001/   user2)
Access: 2020-05-13 22:23:42.601442674 +0900
Modify: 2020-05-04 12:03:18.362285911 +0900
Change: 2020-05-04 12:03:18.362285911 +0900
 Birth: -
```

ls 명령어에서의 시간 확인

- ls -l : 기본값 (mtime)
- ls -l -u : atime
- ls -l -c : ctime

## 파일 시스템 명령어 - 검색 (find) advanced

루트 디렉토리로부터 파일 사이즈가 100M 이상인 파일을 찾아서 ls 로 상세 표시하기

- `find / -size 100M -exec ls -l {} \; 2>/dev/null`

```
user1@user1-VirtualBox:~$ find / -size +100M -exec ls -l {} \; 2>/dev/null
-rw----- 1 root root 123150336 4월  8 00:32 /core
-r----- 1 root root 140737477881856 5월 13 22:20 /proc/kcore
-rw-r--r-- 1 root root 125842592 5월  9 01:22 /usr/lib/firefox/libxul.so
-rw-r--r-- 1 root root 122208776 4월 18 06:32 /usr/lib/thunderbird/libxul.so
-rwxr-xr-x 1 root root 162394240 4월 15 08:27 /opt/google/chrome/chrome
user1@user1-VirtualBox:~$
```

루트 디렉토리로부터 txt 파일을 찾아서 그 안에 “HELP” 라는 글자가 포함된 파일과 내용 표시

- `find / -name "*.txt" -exec grep "HELP" {} \; -print 2>/dev/null`

```
user1@user1-VirtualBox:~$ find / -name "*.txt" -exec grep "HELP" {} \; -print 2>/dev/null
<HELP>          <Help>          0x9c    156    0xcc    204
/usr/share/vim/vim74/doc/version4.txt
- Use "datadir" in configure, instead of our own check for HELPDIR.
Solution:  When the BCO_NOHELP flag is given to buf_copy_options(), still
Solution:  Only return the help line when called with the GCS_HELPTEXT flag.
/usr/share/vim/vim74/doc/version5.txt
QUICK HELP                                           *netrw-quickhelp* {{{2
/usr/share/vim/vim74/doc/pi_netrw.txt
:syn keyword myword HELP containedin=cComment contained
/usr/share/vim/vim74/doc/syntax.txt
user1@user1-VirtualBox:~$
```

루트 디렉토리로부터 특정 [조건]의 파일 찾아서 특정 [디렉토리] 로 복사하기

- `find / [조건문] -print -exec cp {} [경로] \; 2>/dev/null`

```
user1@user1-VirtualBox:~$ find . -size +50M -exec cp {} /tmp \;
```

## 검색 명령어 - 필터링 (grep)

grep [OPTION] PATTERN [FILE]

특정 패턴 검색 (또는 정규표현식 패턴 검색)

파일 내에서 usage 라는 단어 검색

- grep "usage" [FILE]

파일 내에서 "vim" 또는 "Vim" 이라는 단어 각각 찾는 법 (및 대소문자 구분 없이)

- grep "vim" [FILE] (대소문자 구분)
- grep "Vim" [FILE]
- grep -i "vim" [FILE] (대소문자 무시)

하위 디렉토리 모두 검색

- grep -r "vim" [PATH]

```
user1@user1-VirtualBox:~$ grep "usage" /usr/share/doc/vim/copyright
that you made from the unmodified Vim sources, plus your own usage
user1@user1-VirtualBox:~$ grep "vim" /usr/share/doc/vim/copyright
This is the Debian pre-packaged version of vim. Vim is an almost compatible
sources obtained from: ftp://ftp.vim.org/pub/vim/unix/. The current maintainer
is Debian VIM Maintainers <pkg-vim-maintainers@lists.alioth.debian.org>.
The current maintainer is Bram Moolenaar <Bram@vim.org>. If this
vim.sf.net, www.vim.org and/or comp.editors). When it is completely
<maintainer@vim.org>
```

```
user1@user1-VirtualBox:~$ grep -r "vim" /usr/share/doc/
/usr/share/doc/libgstreamer-plugins-good1.0-0/copyright:Files: gst/avi/gstavimux.h
/usr/share/doc/libgstreamer-plugins-good1.0-0/copyright:Files: tests/check/elements/avimux.c
/usr/share/doc/libgstreamer-plugins-good1.0-0/copyright:Files: gst/avi/gstavimux.c
/usr/share/doc/gstreamer1.0-plugins-base-apps/copyright: sys/xvimage/xvimagepool.c
/usr/share/doc/gstreamer1.0-plugins-base-apps/copyright: sys/xvimage/xvimagepool.h
/usr/share/doc/gstreamer1.0-plugins-base-apps/copyright: sys/xvimage/xvimagesink.c
/usr/share/doc/gstreamer1.0-plugins-base-apps/copyright: sys/xvimage/xvimagesink.h
/usr/share/doc/gstreamer1.0-plugins-base-apps/copyright: sys/xvimage/xvimage.c
/usr/share/doc/oxideqt-codecs/copyright:Files: chromium/src/native_client/tests/vim/{vim_tool.sh}
/usr/share/doc/liboxideqtquick0/copyright:Files: chromium/src/native_client/tests/vim/{vim_tool.sh}
/usr/share/doc/git/RelNotes/2.7.2.txt: * The vimdiff backend for "git mergetool" has been tweaked to
```

## 검색 명령어 - 필터링 (grep) advanced

정규표현식 (RegEx = Regular Expression)

- 기본 정규 표현식 (Basic Regular Expression)
  - 실습 : `grep -n "PATTERN" /usr/share/doc/vim/copyright`
    - `^s` (문장의 시작이 s 로 시작하는 줄)
    - `e$` (문장의 끝이 e 로 끝나는 줄)
    - `..e$` (문장의 끝이 3글자 중 e로 끝나는 줄)
    - `app*` (문장의 시작/중간/끝 이 ap와 p의 "0개혹은 그 이상" 의 개수를 갖고 있는 줄)
    - `^[at]` (문장의 시작 첫 단어가 a 또는 t 로 시작하는 줄)
    - `[0-9]` (문장의 중간에 숫자 0~9 까지를 포함하고 있는 줄)
- 확장 정규 표현식 (Extended Regular Expression)
  - 실습 : `grep -E "PATTERN" /usr/share/doc/vim/copyright` (`grep -E` 는 `egrep` 과 동일)
    - `[p]{2}` (문장 내 p라는 글자가 연속 두번 나오는 경우)
    - `^[a-zA-Z0-9]{3,9}` (문장의 시작이 소문자/대문자/숫자 로 시작하는 3~9길이)

정규표현식 테스트 사이트

- <https://regexr.com/>
- <https://regex101.com/>

```
user1@user1-VirtualBox:~$ grep -n -E "^[a-zA-Z0-9]{3,9}" /usr/share/doc/vim/copyright
2:version of the UNIX editor Vi. Many new features have been added: multi level
3:undo, syntax highlighting, command line history, on-line help, filename
4:completion, block operations, etc.
7:sources obtained from: ftp://ftp.vim.org/pub/vim/unix/. The current maintainer
11:from the online documentation.
92:the filesystem at /usr/share/common-licenses/GPL-2.
106:part, in any medium physical or electronic, provided that the terms of this
107:license are adhered to, and that this license or an incorporation of it by
108:reference (with any options elected by the author(s) and/or publisher) is
```

## 검색 명령어 - 필터링 (grep) 응용

더 일반적인 활용 사례

- 파이프로 연결한 grep 의 응용

파일 목록에서 특정 단어 검색

- `ls -al | grep txt`

로그 파일에서 경고만 검색

- `cat /var/log/syslog | grep -i "warn"`

프로세스 목록에서 특정 단어 검색 (및 특정 단어 예외)

- `ps x | grep "/bin"`
- `ps x | grep "/bin" | grep -v "grep"`

특정 포트가 열려 있는지 확인

- `netstat -a | grep 80`
- `netstat -a | grep ":80"`

## 정렬 명령어 - 소팅 (sort)

sort [OPTION] [FILE]

파일의 내용을 특정 순서(옵션)로 정렬 (그러나 현실적으로는 FILE 보다는 PIPE 와 더 많이 연동)

디렉토리 목록을 소팅 (기본값 : 첫번째 컬럼)

- `ls -l | sort`

디렉토리 목록을 두번째 컬럼으로 소팅 (기본값 : 캐릭터 소팅)

- `ls -l | sort -k 2`

디렉토리 목록을 두번째 컬럼으로 숫자로 소팅

- `ls -l | sort -k 2 -n` (또는 `-k2n` 붙여써도 무방)

디렉토리 목록을 파일 사이즈별로 소팅 (기본값: 오름차순)

- `ls -l | sort -k 5 -n`

디렉토리 목록을 파일 사이즈별로 역순 소팅 (내림차순)

- `ls -l | sort -k 5 -n -r`

두개 이상의 키로 소팅 (두번째 컬럼(숫자) & 다섯번째 컬럼)

- `ls -l | sort -k 2n -k5`

파이프 외에 인풋 리디렉션도 가능

- `sort -k 2 -k 5 < dir.txt`

```
user1@user1-VirtualBox:~$ ls -l | sort -k 5
합계 156
lrwxrwxrwx 1 user1 user1 9 5월 3 23:41 hellosymlink -> hello.txt
-rw-r----- 1 user1 user1 9 5월 4 23:00 readonly.txt
-rw-rw-r-- 2 user1 user2 12 5월 4 12:03 hello.txt
-rw-rw-r-- 2 user1 user2 12 5월 4 12:03 hellolink
-rwxrwxr-x 1 user1 user1 24 5월 5 01:26 test2.sh
-rwxrwxr-x 1 user1 user1 30 5월 5 01:25 test1.sh
-rw-rw-r-- 1 user1 user1 356 5월 4 11:38 file.txt
-rw-rw-r-- 1 user1 user1 586 5월 4 12:48 result.txt
-rw-rw-r-- 1 user1 user1 1350 5월 13 23:38 dir.txt
drwxr-xr-x 2 user1 user1 4096 4월 5 18:54 공개
drwxr-xr-x 2 user1 user1 4096 4월 5 18:54 문서
drwxr-xr-x 2 user1 user1 4096 4월 5 18:54 바탕화면
drwxr-xr-x 2 user1 user1 4096 4월 5 18:54 비디오
drwxr-xr-x 2 user1 user1 4096 4월 5 18:54 사진
drwxr-xr-x 2 user1 user1 4096 4월 5 18:54 음악
drwxr-xr-x 2 user1 user1 4096 4월 5 18:54 템플릿
drwxr-xr-x 2 user1 user1 4096 4월 18 01:18 다운로드
drwxrwxr-x 4 user1 user1 4096 4월 25 00:39 dir1
drwxrwxr-x 3 user1 user1 4096 4월 25 00:40 dir2
drwxrwxr-x 2 user1 user1 4096 5월 3 22:46 testdir
drwxrwxrwt 2 user1 user1 4096 5월 4 00:10 dir3
-rw-r--r-- 1 user1 user1 8980 4월 5 18:38 examples.desktop
```



## 내용 검색/편집 명령어 (awk)

### 패턴 검색 및 텍스트 프로세싱

디렉토리 목록 중 첫번째 컬럼만 출력

- `ls -l | awk '{print $1}'`

디렉토리 목록 중 파일명과 사이즈만 출력 (아홉번째 컬럼, 다섯번째 컬럼)

- `ls -l | awk '{print $9, $5}'`
- `ls -l | awk '{print "FILENAME:"$9, "SIZE:"$5}'`

디렉토리 목록 중 사이즈를 모두 더해서 결과만 출력

- `ls -l | awk '{sum += $5} END {print sum}'`

디렉토리 목록 중 파일 사이즈별로 소팅해서 10000 바이트보다 큰 것만 출력

- `ls -l | sort -k 5 | awk '$5 >= 10000 {print}'`

암호 파일에서 콜론(:) 을 구분자로 잘라서 첫번째 컬럼만 출력

- `cat /etc/passwd | awk -F":" '{print $1}'`



## 내용 검색/편집 명령어 (sed)

sed 's/패턴/변환/g'

스트림라인 편집기 (search and replace)

파일 내의 모든 book 을 books 로 변경

- `cat /usr/share/doc/vim/copyright | sed 's/book/books/g'`

(글자가 있는) 모든 줄의 맨 끝을 ! 표로 끝나도록 변경

- `cat /usr/share/doc/vim/copyright | sed 's./!/g'`

출력 결과를 소문자에서 대문자로 변경

- `ls -l | sed 's/[a-z]/\U&/g'`

출력 결과를 대문자에서 소문자로 변경

- `cat /etc/passwd | sed 's/[A-Z]/\L&/g'`

## 기타 명령어 - 분석(uniq / wc)

중복제거 유틸 (uniq) 및 단어분석 (word-count)

파일 내에 중복되는 줄 제거

- `cat hello.txt | uniq`

파일 내의 “라인수/단어수/문자수” 출력

- `wc hello.txt` (또는 `cat hello.txt | wc`)
- `wc -l hello.txt` (라인수만 출력)

```
user1@user1-VirtualBox:~$ wc dir.txt
25  220 1350 dir.txt
user1@user1-VirtualBox:~$ wc hello.txt
2  2 12 hello.txt
user1@user1-VirtualBox:~$ cat hello.txt
hello
world
user1@user1-VirtualBox:~$
```

### 종합 응용:

인증 로그 내에서 시간값 (처음 3컬럼) 을 제거하고 나머지 값에서 uniq 한 메시지만 출력

- `cat /var/log/auth.log | awk '$1=$2=$3=""; {print $0}' | sort | uniq`
- `cat /var/log/auth.log | awk '$1=$2=$3=""; {print substr($0,4)}' | sort | uniq`
- `cat /var/log/auth.log | awk '{out=""; for(i=4;i<=NF;i++){out=out" "$i}; print out}' | sort | uniq`

## 파일시스템 주요 명령어 (디스크 용량) - du (disk usage)

du [OPTION] [FILE]

파일 용량 출력

현재 디렉토리부터 사용된 용량 확인

- du

각종 옵션:

- k : 결과 값의 KB 단위 출력 (기본값)
- m : 결과 값의 MB 단위 출력
- h : 사용자 편의 용량 (KB/MB/GB, 1M=2<sup>20</sup>=1,048,576)
- H : 사용자 편의 용량 (KB/MB/GB, 1M=1,000,000)
- s : 합계만 출력
- S : 서브 디렉토리 용량 합치지 않고, 각각 계산

사용 예시:

- du -S | sort -n (디렉토리별 용량을 오름차순 소팅해서 출력)
- du / -h 2>/dev/null | grep [0-9]G (디렉토리별 누적 용량을 출력하여 GB 이상의 디렉토리 출력)
- du --max-depth=1 (디렉토리 용량을 최대 1 디렉토리 depth 까지만 출력)

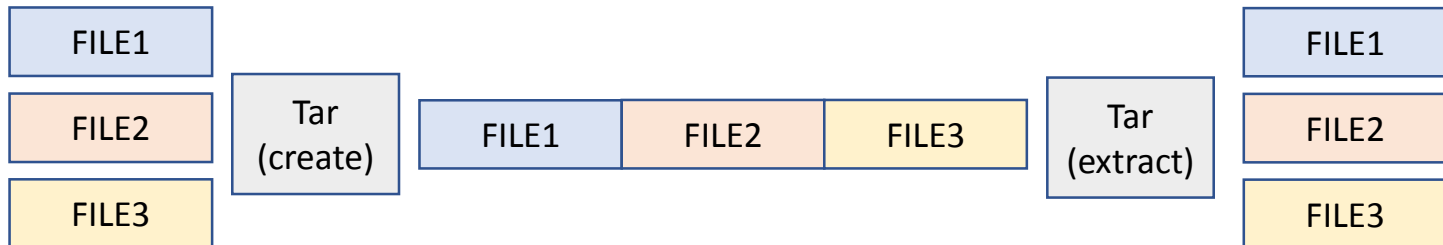
## 파일시스템 주요 명령어 (묶음/압축) – tar (tape archive)

tar [OPTION] [FILE] [PATH]

파일 묶음

각종 옵션:

- c : create (생성)
- x : extract (해지)
- v : verbose (디테일한 상황 보고 - 실행 중 파일 목록 출력)
- f : file (저장될 파일명 지정하기 위해)
- t : list (목록 확인)
- z : zip (압축)
- j : bzip2 (압축)



활용예시:

- tar 아카이브 만들기
  - tar cvf myzip.tar dir1
- tar 아카이브 내용 확인
  - tar tf myzip.tar
- tar 아카이브 풀기
  - tar xvf myzip.tar

활용예시 (압축버전):

- tar.gz 아카이브 만들기
  - tar cvfz myzip.tgz dir1
- tar 아카이브 내용 확인
  - tar tf myzip.tgz
- tar 아카이브 풀기
  - tar xvfz myzip.tgz

활용예시 (압축버전2):

- tar.bz2 아카이브 만들기
  - tar cvfj myzip.tar.bz2 dir1
- tar 아카이브 내용 확인
  - tar tf myzip.tar.bz2
- tar 아카이브 풀기
  - tar xvfj myzip.tar.bz2

## 파일시스템 주요 명령어 (묶음/압축) – gz, bz2, xz

gzip [OPTION] [FILE]

파일 압축

다양한 압축 유틸리티들 (시대의 변화에 따라, 압축 알고리즘의 발전에 따라...)

- gzip
- bzip2
- xz

알고리즘 비교

(리눅스 커널로 테스트 결과)

- 압축 용량 (작은게 좋음)
  - gzip > bzip2 > xz
- 압축 시간 (작은게 좋음)
  - gzip < bzip2 << xz
- 압축 해지 시간 (작은게 좋음)
  - gzip < xz < bzip2

사용 방법:

- 압축 하기
  - gzip filename
  - bzip2 filename
  - xz filename
    - 각각 filename.gz, filename.bz2, filename.xz 형태로 압축됨
- 압축 풀기
  - gzip -d filename
  - bzip2 -d filename
  - xz -d filename

## 셸 스크립트의 활용

내가 자주 사용하는 명령어를 스크립트로...

용량이 100M 보다 큰 파일 찾기

```
bash$ cat find_large.sh
#!/bin/bash
find . -size +100000000c
```

용량이 \$1 입력 인자보다 큰 파일 찾기

```
bash$ cat find_large2.sh
#!/bin/bash
find . -size +$1
```

오래된 파일 찾기 (1년 이상 된 파일 찾기)

```
bash$ cat find_old.sh
#!/bin/bash
find . -mtime +365 -type f
```

입력 조건문에 따른 분기문 처리

```
bash$ cat find_opt.sh
#!/bin/bash
if [ $# -eq 0 ]; then
    echo "usage: $0 [option]"
    exit 1
fi
```

case \$1 in

1)

ls

;;

2)

pwd

;;

\*)

echo "Error"

esac