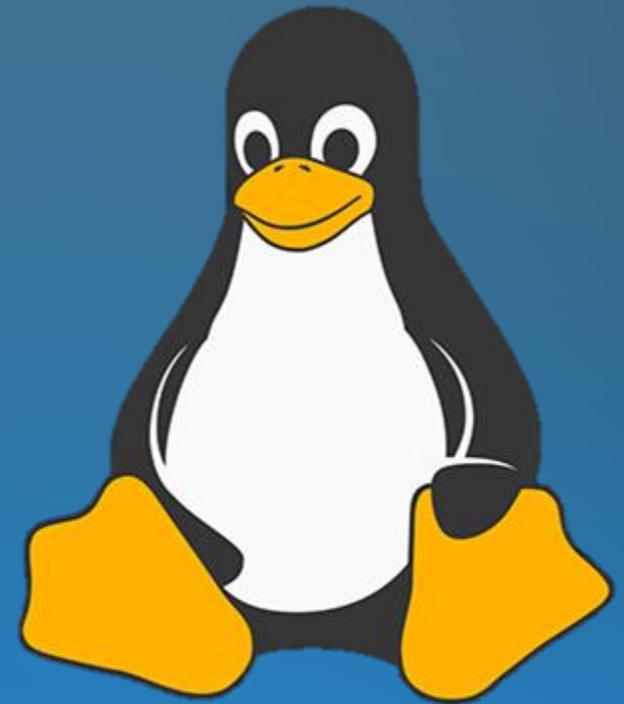


Linux

Triangle
2021 Winter Wheel Seminar

Operating
System



Linux

OS

1. 하드웨어 관리
2. 소프트웨어 간 자원 관리

자원 : CPU, Memory, I/O Device ...

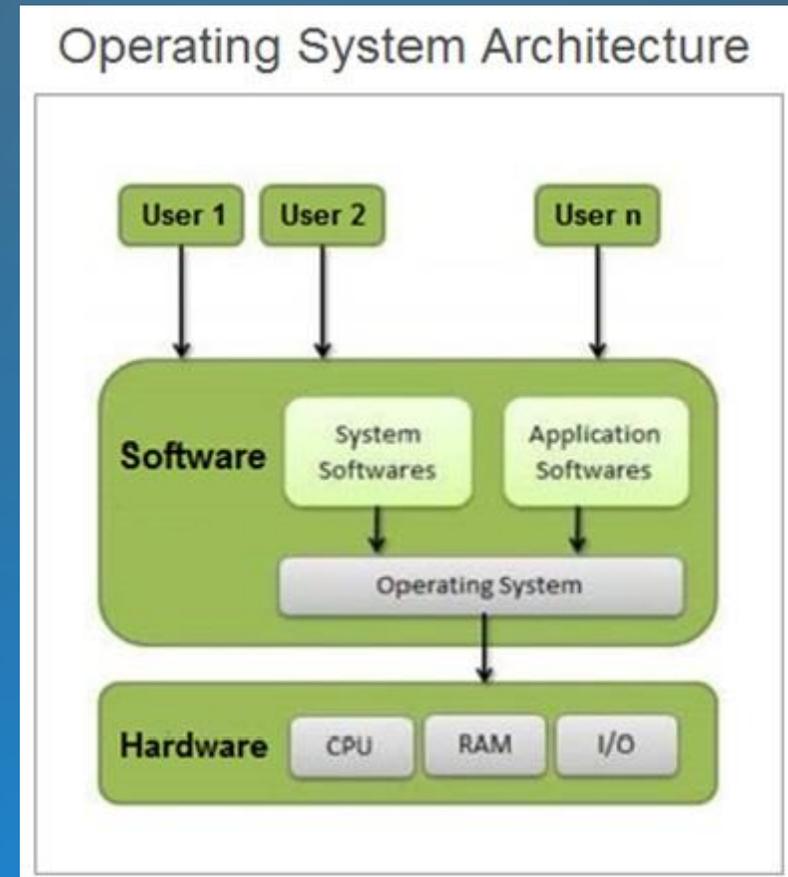
OS: 효율적으로 할당, 관리, 보호

CPU 스케줄링

I/O Device management

memory 관리

file system 제공



OS Structure

여러 껍질들로 덮혀 있는 형태

Kernel: OS에서 가장 낮은 level의 소프트웨어

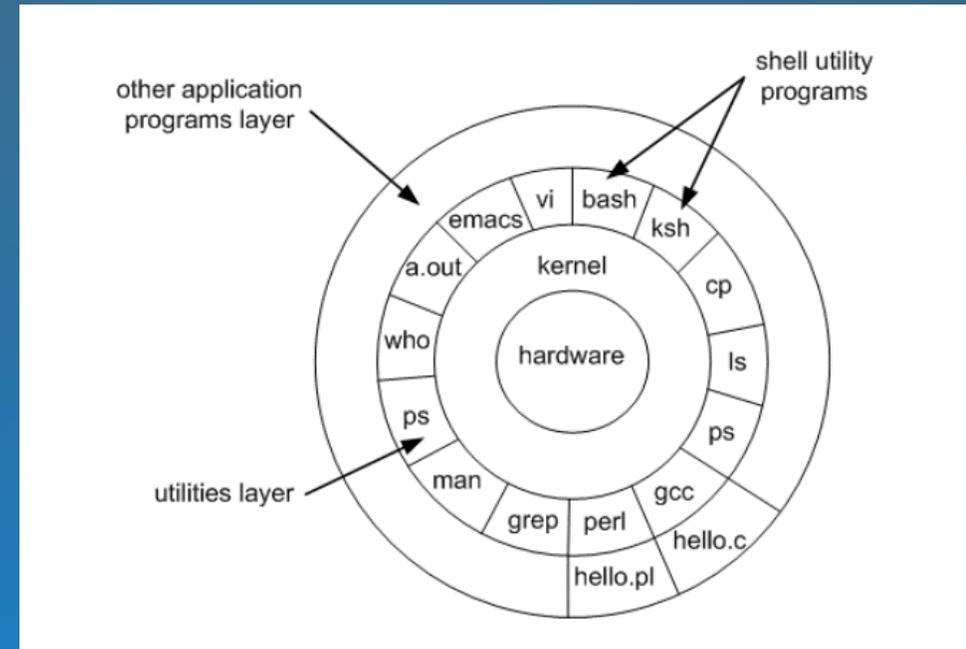
- 보안, 스케줄링, 관리, 하드웨어 추상화
- 보호된 메모리 영역

Shell: 사용자가 접근하는 사용자 interface

CLI / GUI로 조작 (bash, zsh)

입력을 해석 후 커널에서 실행, 결과 출력

Application: 여러 쉘 명령어로 이루어진 프로그램



Unix

현대적 컴퓨터 OS의 원형, 일종의 표준

C언어 기반 작성

Multitask & multiuser 지원

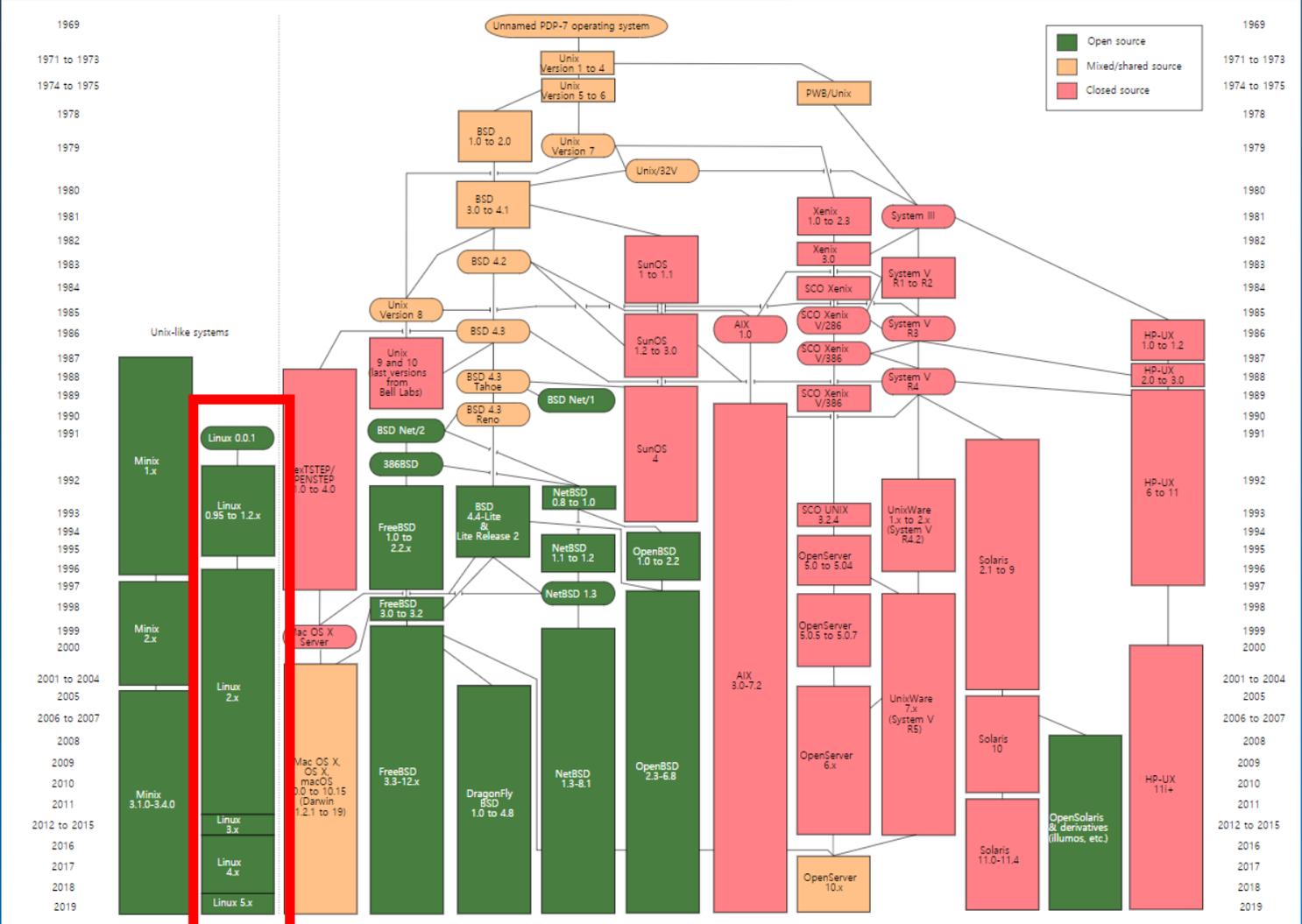
POSIX (Portable Operating System Interface + X),

TCP/IP

POSIX 표준 만족 => Unix-like OS

- macOS, iOS (유전적 Unix)
- Linux (기능적 Unix)

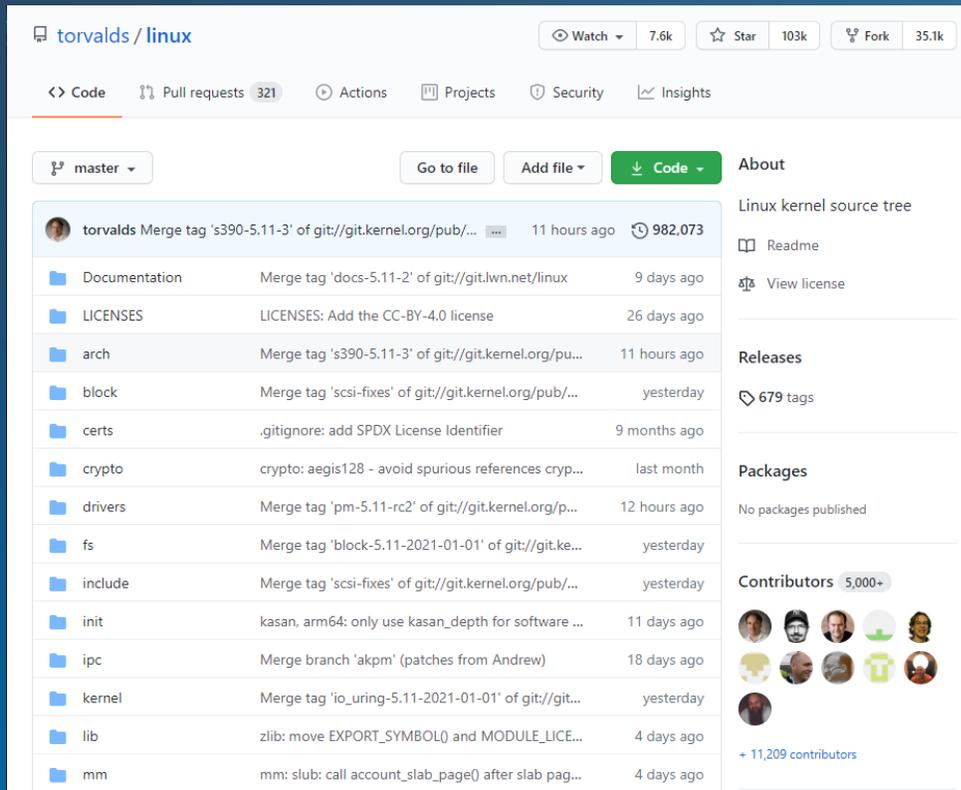
Unix-like OS



Linux

'91: Linux Torvalds의 개인 프로젝트

오픈 소스의 특성: 빠른 속도로 발전 (<https://github.com/torvalds/linux>)



torvalds / linux

Watch 7.6k Star 103k Fork 35.1k

Code Pull requests 321 Actions Projects Security Insights

master

torvalds Merge tag 's390-5.11-3' of git://git.kernel.org/pub/... 11 hours ago 982,073

- Documentation Merge tag 'docs-5.11-2' of git://git.lwn.net/linux 9 days ago
- LICENSES LICENSES: Add the CC-BY-4.0 license 26 days ago
- arch Merge tag 's390-5.11-3' of git://git.kernel.org/pub/... 11 hours ago
- block Merge tag 'scsi-fixes' of git://git.kernel.org/pub/... yesterday
- certs .gitignore: add SPDX License Identifier 9 months ago
- crypto crypto: aegis128 - avoid spurious references cryp... last month
- drivers Merge tag 'pm-5.11-rc2' of git://git.kernel.org/p... 12 hours ago
- fs Merge tag 'block-5.11-2021-01-01' of git://git.ke... yesterday
- include Merge tag 'scsi-fixes' of git://git.kernel.org/pub/... yesterday
- init kasan, arm64: only use kasan_depth for software ... 11 days ago
- ipc Merge branch 'akpm' (patches from Andrew) 18 days ago
- kernel Merge tag 'io_uring-5.11-2021-01-01' of git://git... yesterday
- lib zlib: move EXPORT_SYMBOL() and MODULE_LICE... 4 days ago
- mm mm: slub: call account_slab_page() after slab pag... 4 days ago

About Linux kernel source tree

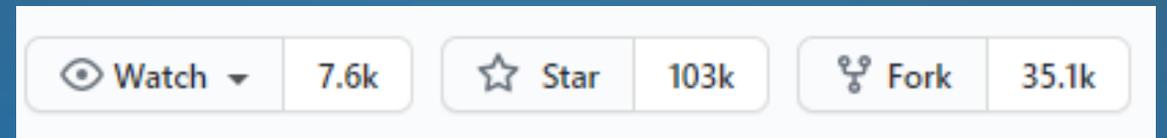
Readme View license

Releases 679 tags

Packages No packages published

Contributors 5,000+

+ 11,209 contributors



Watch 7.6k Star 103k Fork 35.1k

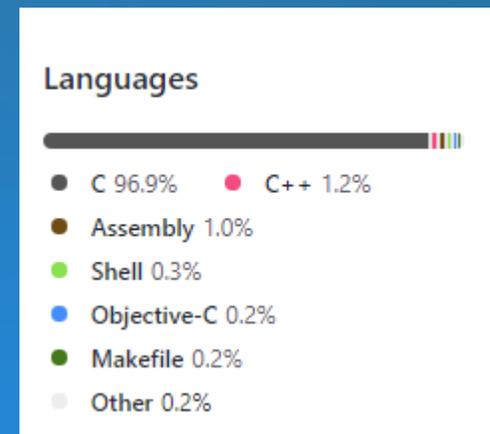


torvalds Merge tag 's390-5.11-3' of git://git.kernel.org/pub/... 11 hours ago 982,073



Contributors 5,000+

+ 11,209 contributors



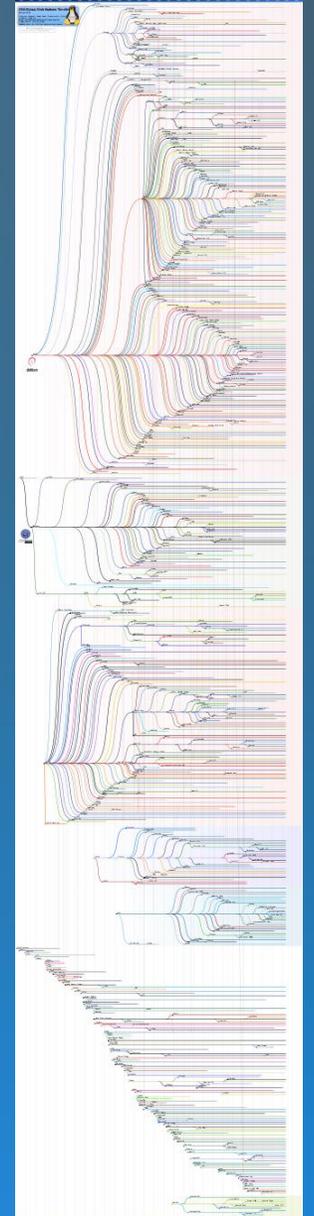
Languages

- C 96.9%
- C++ 1.2%
- Assembly 1.0%
- Shell 0.3%
- Objective-C 0.2%
- Makefile 0.2%
- Other 0.2%

Linux

리눅스 : 커널

다양한 배포판 존재



https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux_Distribution_Timeline.svg

Filesystem Hierarchy Standard



경로 표현

Root directory: `/`

현재 경로: `.`

상위 디렉토리: `..` (`/..` == `/`)

홈 디렉토리: `~`

절대경로

현재 위치와 무관, root directory서 시작

```
/home/sparcs/my_file.txt  
/var/log
```

상대경로

현재 위치에서 시작

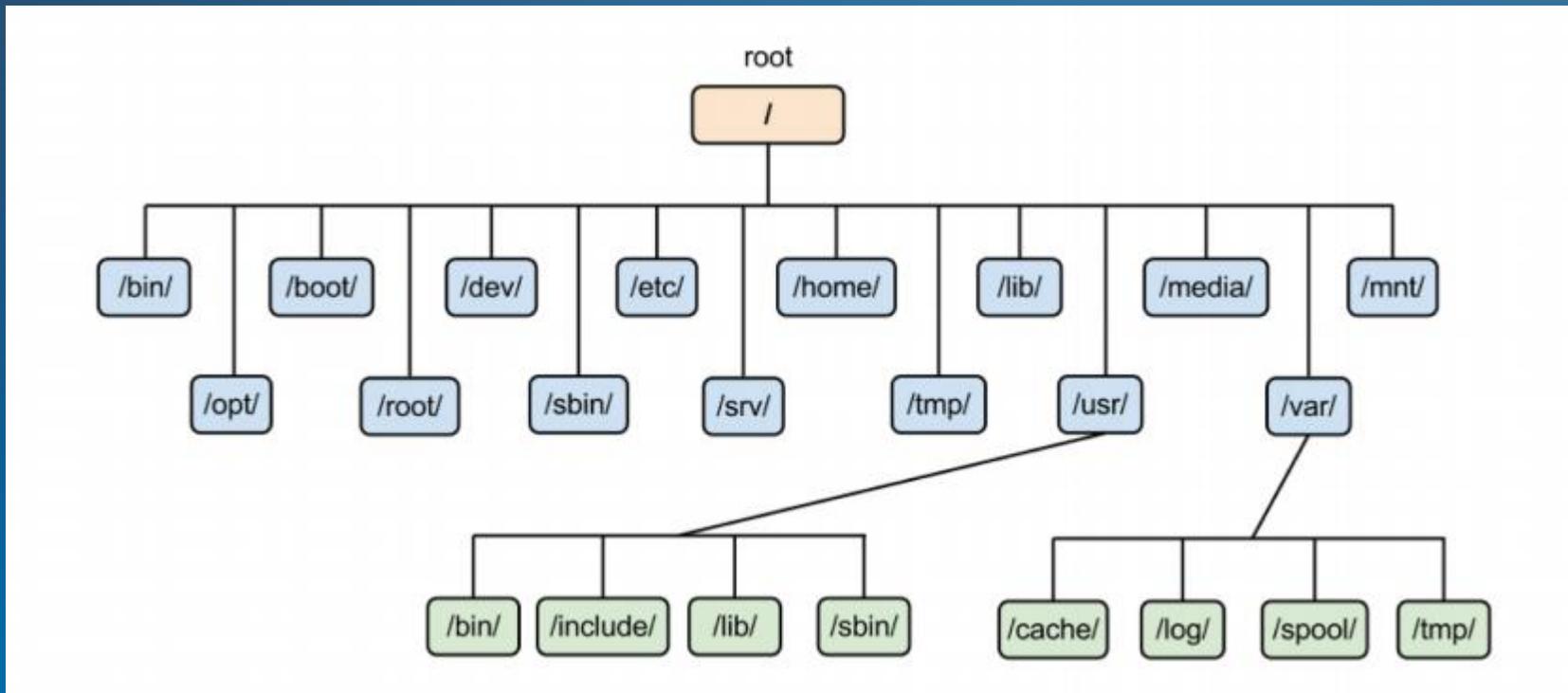
```
./my_file.txt (./ 생략 가능)  
../../var/log
```

Linux Directory Structure

Filesystem Hierarchy Standard (FHS) 준수

자료의 성격, 내용, 권한 따라 다른 디렉토리

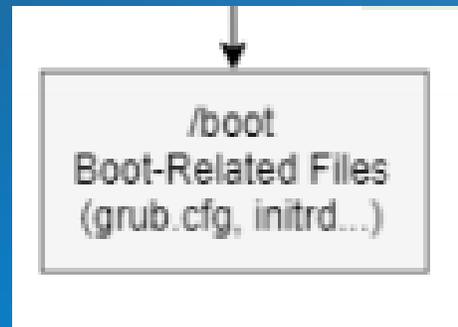
`man hier` 명령어 (<https://man7.org/linux/man-pages/man7/hier.7.html>)



FHS: `/boot`

부팅에 필수적인 bootloader, 커널 위치

건드리지 않는 것이 좋다



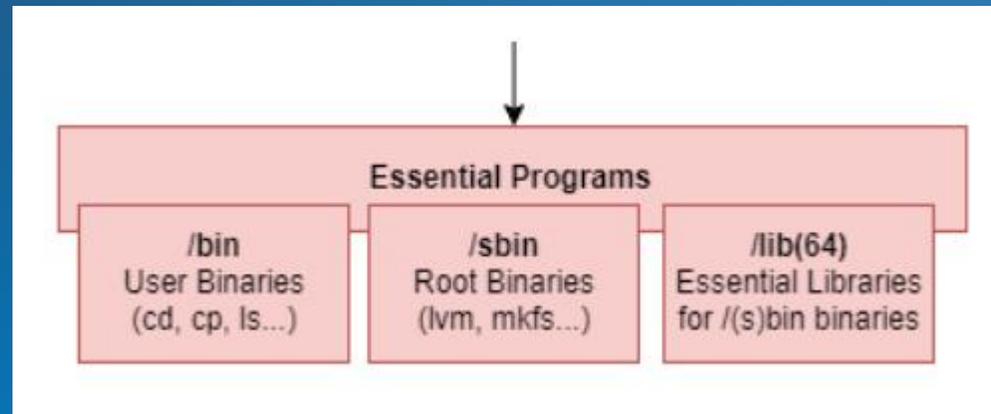
FHS: `/bin` `/sbin`

`bin` : user binaries

기본적인 Linux 명령어, 셸 위치

`sbin` : system binaries

root user 전용 명령어 위치

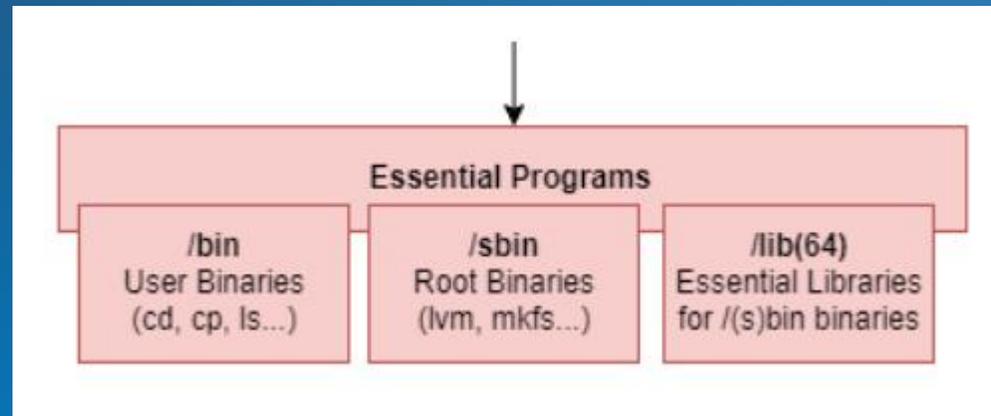


FHS: **/lib**

lib : system **lib**raries

커널이 필요로 하는 라이브러리 위치

대부분 symbolic link로 연결



FHS: `/etc` `/opt`

`etc` : config files

시스템 설정 파일 위치

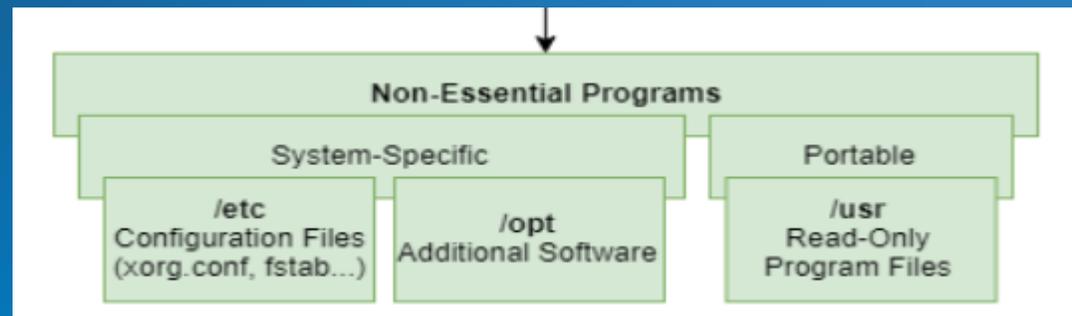
자료 파일 관리자용 명령어도 위치

`/etc/skel` : 새 계정 생성 시 기본 파일들의 뼈대가 되는 폴더 (`.bash_logout` `.bashrc` 등)

`/etc/passwd` : 리눅스 사용자 정보 저장, 비밀번호 x

`/etc/group` : 리눅스 그룹 정보 저장

`opt`: optional add-ons



FHS: **/usr**

usr : **user** programs

일반 사용자들을 위한 프로그램 위치 (C:\Program Files\)

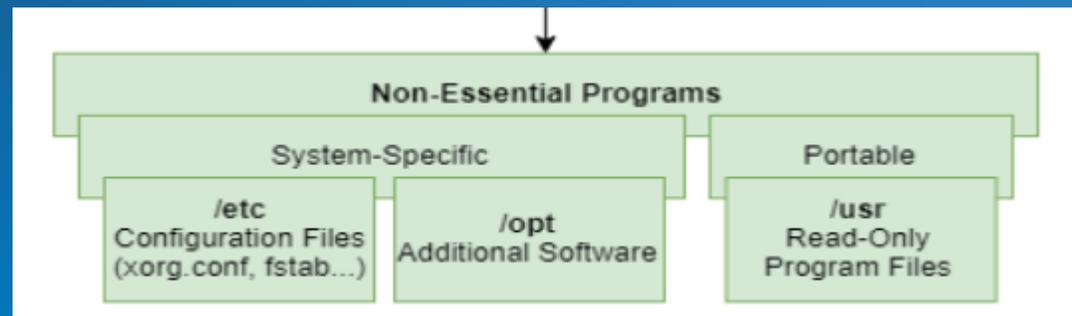
/usr/src : 시스템에 빌드하는 프로그램, 커널 소스 보관

/usr/share : 매뉴얼 등의 자료

/usr/local : 애플리케이션을 소스로 컴파일 시 사용

/usr/include : C 컴파일러를 위한 파일 포함

/usr/lib : /lib에 들어가지 않은 라이브러리 디렉토리



FHS: `/usr/bin` `/usr/local/bin`

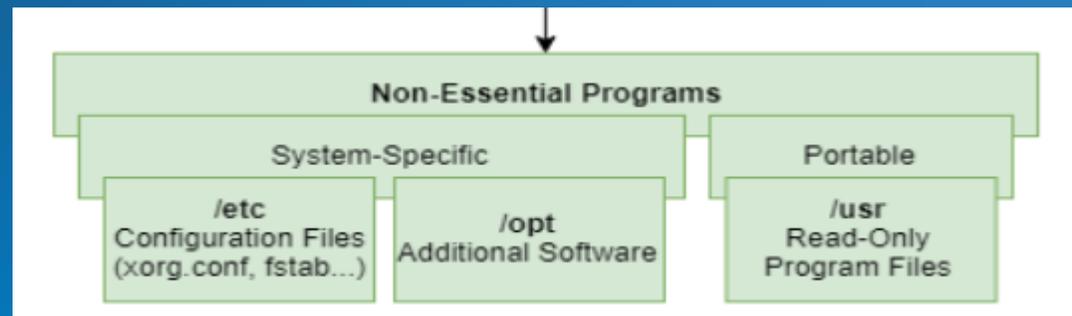
Binary directory 차이

`/bin` : 아주 기본적인 프로그램 위치 (리눅스)

`/usr/bin` : 확장된 프로그램, 각 리눅스 배포판의 패키지 관리자에 의해 관리

`/usr/local/bin` : 로컬에서 컴파일된 패키지들

`sbin` 은 위와 큰 차이가 없으나, 실행하기 위해선 root권한 필요



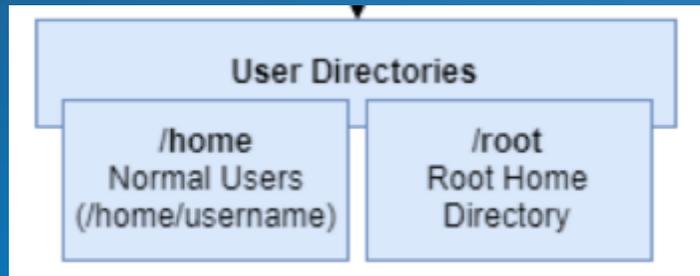
FHS: /home /root

home

각 리눅스 유저에게 기본 생성되는 홈 디렉토리 위치 (home/sparcs/)

root : **root** home

root 유저를 위한 홈 디렉토리



FHS: /proc /sys /run

proc : process

프로세스, 커널 정보 가진 파일들 위치

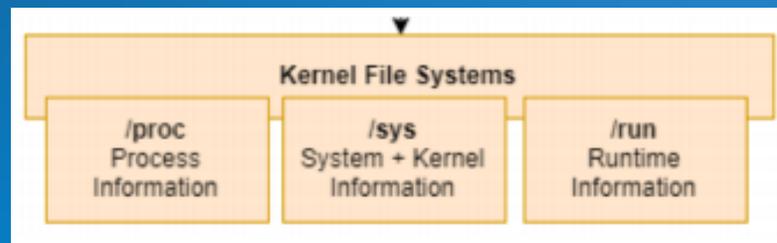
cpuinfo, meminfo, stat, uptime, version 등

sys: system

디바이스, 드라이버, 일부 커널 정보 가진 파일들 위치

run: runtime

런타임 정보 가진 파일들 위치 (로그인된 유저, 실행 중 daemon 조회)



FHS: /media /mnt /tmp

media, mnt (mount)

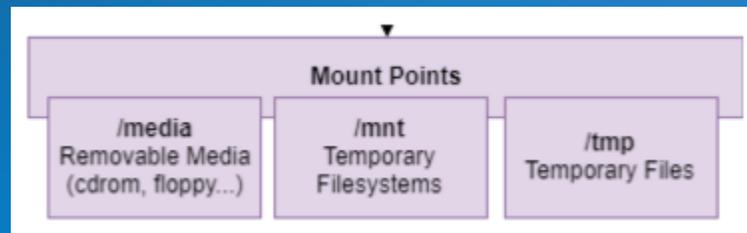
mount : 하드웨어를 운영체제에 인식 가능하게 해주는 것

mnt는 사용자가 mount 할 때, media는 시스템이 mount 할 때 사용

mnt directory는 변경 가능

tmp: temporary

임시 파일 저장, 공동 사용, 시스템 종료시 삭제



FHS: /var

var : variable files

내용 자주 변경되는 파일 저장

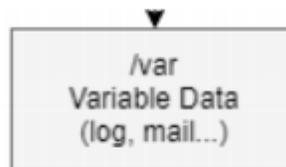
/var/log : 시스템 로그 파일

/var/spool : 메일, 프린트 등 위한 스푼 파일

/var/lib : 시스템 운용 시 계속 바뀌는 파일

/var/tmp : 임시 디렉토리, /tmp보단 오래 지속

/var/run : 시스템 정보 저장, 부팅 시 초기화



FHS: /dev

dev : **device**

HDD, ODD등의 시스템 디바이스 파일 위치

자원 접근하는데 사용

/dev/null : 데이터 블랙홀

/dev/zero : 무수히 많은 0 출력

/dev/urandom : 랜덤 비트 출력



FHS: /srv

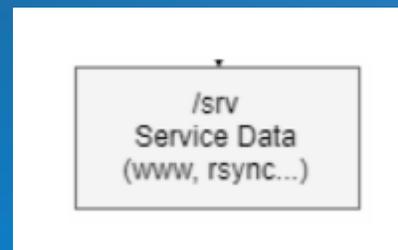
srv : site-specific data **served** by this system

시스템에서 제공되는 사이트 특정 데이터

웹 서버를 위한 데이터/스크립트

FTP 서버를 위해 권장되는 데이터

VCS를 위한 repository



Shell Command



환경변수 및 PATH

환경변수 (environment variable) : 프로세스의 동작에 영향 미치는 변수들

echo \$[name]: 현재 bash가 사용중인 환경변수 [name] 출력

PATH: 대표적인 환경변수, 실행 가능한 binary file를 담는 디렉토리를 :로 구별

```
wheelseminar@tong:~$ echo $PATH  
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
```

Shell Commands

Basic Commands

File/directory
Commands

User/privilege
Commands

Utility
Commands

Process
Commands

Network
Commands

Package &
service
management
Commands

Shell utilities

<https://bellard.org/jslinux/>

1. Basic Commands

`man [command]`: manual, command에 대한 설명 출력

`echo [text]`: stdout으로 text 출력

`which [command]`: command의 절대 경로를 출력

PATH 환경변수내 디렉토리 중 어디에 명령어 있는지 확인

Ctrl + C : command 실행 중 프로그램 종료 후 shell로 복귀 할 때

방향키 (history), Ctrl + L(clear) ...

기존의 복사 붙여넣기: Ctrl + Insert, Shift + Insert

`--help`

```
report bugs to cjwatson@debian.org.
sparcs@43b5c5dc7680:~/test_1$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
-a, --all                do not ignore entries starting with .
-A, --almost-all       do not list implied . and ..
--author                with -l, print the author of each file
```

2. File/directory Commands

pwd: **p**rint **w**orking **d**irectory, 현재 디렉토리 절대경로 출력

ls [**dir**]: **l**ist, [**dir**] 내 파일/디렉토리 출력 ([**dir**] 없을경우 현재 디렉토리)

-a: **a**ll, 숨김 파일 포함 출력

-l: **l**ist, 상세 정보 출력 (종류, 권한, 수정 날짜 등)

-h: **h**uman-readable, k/M/GB 등 사람이 읽기 쉽게

-i: **i**node, inode 확인

cd [**dir**]: **c**hange **d**irectory, [**dir**]로 이동 (~: /etc/passwd 정의)

touch [**file**]: 파일 존재 ? 현재 날짜로 업데이트 : 빈 파일 생성

2. File/directory Commands

`mv [a] [b]`: **move**, 파일, 디렉토리 이동 (a to b) - 이름 변경 할 때도 사용

`cp [a] [b]`: **copy**, 파일, 디렉토리 복사 (a to b)

`-r`: **recursive**, 디렉토리 복사 시 필수

`rm [a]`: **remove**, 파일 삭제

`-r`: **recursive**, 디렉토리 삭제 시 필수

`-f`: **force**, 경고 없이 삭제

~~`rm -rf /`~~

2. File/directory Commands

mkdir [a]: **make directory**, [a] 이름 디렉토리 생성 / (rmdir)

-p: **parents**, 부모 디렉토리 존재하지 않을때 같이 생성

du [file]: **disk usage**, 파일, 디렉토리 용량 출력

-s: **summarize**, 해당 디렉토리만

-h: **human-readable**, k/M/GB 등 사람이 읽기 쉽게

2-1. Link

inode: 해당 파일에 대한 부분 정보 포함, 고유한 inode 번호에 매핑

hard link: 원본 파일의 inode에 대한 직접적인 포인터 (same inode)

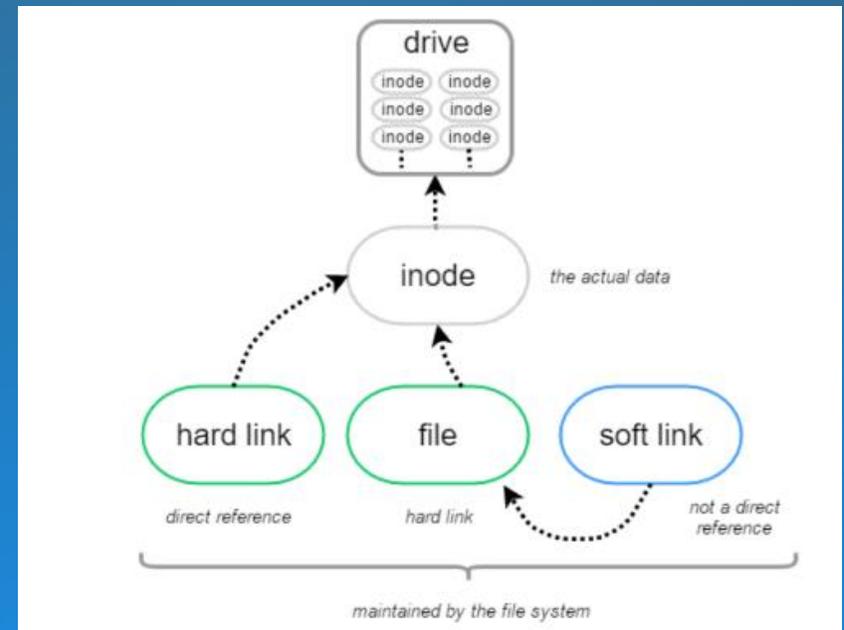
성능, 저장 공간 유리, 디렉토리 생성 X

soft(symbolic) link: 원본 파일의 inode를 가리키는 정보가 있는 파일 (diff inode)

윈도우 바로가기와 비슷

명령어를 버전별 관리 or 목적 지정 용이

```
wheelseminar@tong:~/usr/bin$ ls -l | grep python
lrwxrwxrwx 1 root root      26 Jan 25  2017 dh_pypy -> ../share/dh-python/dh_pypy
-rwxr-xr-x 1 root root    1056 Jan 24  2017 dh_python2
lrwxrwxrwx 1 root root      29 Jan 25  2017 dh_python3 -> ../share/dh-python/dh_python3
lrwxrwxrwx 1 root root      23 Sep 27  2018 pdb2.7 -> ../lib/python2.7/pdb.py
lrwxrwxrwx 1 root root      23 Sep 28  2018 pdb3.5 -> ../lib/python3.5/pdb.py
lrwxrwxrwx 1 root root      31 Jan 20  2017 py3versions -> ../share/python3/py3versions.py
lrwxrwxrwx 1 root root      26 Jan 25  2017 pybuild -> ../share/dh-python/pybuild
lrwxrwxrwx 1 root root      16 Jan 24  2017 python -> python2.7
lrwxrwxrwx 1 root root      16 Jan 24  2017 python-config -> python2.7-config
lrwxrwxrwx 1 root root      16 Jan 24  2017 python2 -> python2.7
lrwxrwxrwx 1 root root      16 Jan 24  2017 python2-config -> python2.7-config
-rwxr-xr-x 1 root root      154 Sep 21  2016 python2-pbr
-rwxr-xr-x 1 root root    3779512 Sep 27  2018 python2.7
lrwxrwxrwx 1 root root      33 Sep 27  2018 python2.7-config -> x86_64-linux-gnu-python2.7-config
lrwxrwxrwx 1 root root      10 Jan 20  2017 python3 -> python3.5
-rwxr-xr-x 2 root root    4751184 Sep 28  2018 python3.5
-rwxr-xr-x 2 root root    4751184 Sep 28  2018 python3.5m
lrwxrwxrwx 1 root root      10 Jan 20  2017 python3m -> python3.5m
lrwxrwxrwx 1 root root      29 Jan 24  2017 pyversions -> ../share/python/pyversions.py
```



2-1. Link Command

`ln [target] [name]: link`, [target] 파일로 [name] 이름의 Link 생성

`-s`: symbolic, Symbolic link 생성

```
wheelseminar@tong:~/appleseed$ ln -s b c
wheelseminar@tong:~/appleseed$ ls
a b c
wheelseminar@tong:~/appleseed$ ls -al
total 12
drwxr-xr-x 3 wheelseminar wheelseminar 4096 Jun 29 01:37 .
drwxr-xr-x 8 wheelseminar wheelseminar 4096 Jun 29 01:25 ..
drwxr-xr-x 2 wheelseminar wheelseminar 4096 Jun 29 01:36 a
-rw-r--r-- 1 wheelseminar wheelseminar    0 Jun 29 01:37 b
lrwxrwxrwx 1 wheelseminar wheelseminar    1 Jun 29 01:37 c -> b
```

2-2. Permission

`ls -al` or `ll`: 현재 디렉토리의 접근 권한 구조 확인 유용

Directory의 경우에는?

```
shum@sol:~$ ls -l
total 20
drwx----- 2 shum  staff  4096 Jan 16 22:04 Mail
drwx----- 3 shum  staff  4096 Jan 16 14:15 csc128
drwxr-xr-x  2 shum  staff  4096 Jan 13 16:42 public
drwxr-xr-x  2 shum  staff  4096 Jan 16 14:07 public_html
-rw-r--r--  1 shum  staff   628 Jan 15 20:04 verse
```

file type

number of hard links

user (owner) name

group name

size

date/time last modified

filename

executable

writeable

readable

other (everyone) permissions

group permissions

user permissions

2-2. Permission Commands

chmod [privilege] [file]: change file mode, [file] 접근 권한 변경

[privilege]: 3자리 8진수 숫자: 사용자 권한 표시

문자열 모드: (u/g/o)(+/-/=)(r w x -)

chown [user].[group] [file]: change owner, [file] 소유자 변경

-R: recursive 옵션

drwxrwxrwx	7	rwX	111
	6	rw-	110
	5	r-x	101
	4	r--	100
	3	-wx	011
	2	-w-	010
	1	--x	001
	0	---	000

d = Directory
r = Read
w = Write
x = Execute

chmod 777

rwX | rwX | rwX
Owner | Group | Others

https://www.reddit.com/r/linux/comments/ayditr/chmod_cheatsheet/

3. User/Privilege Commands

whoami: 내 이름 출력

su [user]: substitute user 다른 사용자로 전환

[user] 없을 때는 root로 전환

sudo [command]: substitute user do 다른 사용자 권한으로 명령어 실행

사용자 전환 없이 root 권한을 얻기 위해 사용

sudo를 사용하기 위해서

/etc/sudoers 파일 수정

sudo 그룹 유저 추가

passwd: password: 사용자의 비밀번호 변경

본인 비밀번호 변경시 주로 사용

3. User/Privilege Commands

`adduser [name] ([group]):` [name] 이름의 사용자 추가

`deluser [name]:` **delete** ~ [name] 이름의 사용자 삭제

[user] 없을 때는 root로 전환

`usermod [name]:` **modify** ~ [name] 이름의 사용자 정보 편집 (option 참고)

/usr/sbin 등에 존재

`addgroup [name]`

`delgroup [name]`

`gpasswd [name]`

`groupmod [name]`

4. Utility Commands

`cat [file]`: concatenate, [file] 내용 전부 출력

`head [file]`: 파일 시작부터 10줄 출력

`tail [file]`: 파일 끝부터 10줄 출력

`-n`: number, 출력 줄 수 조절 가능

`wc [file]`: word count, [file]의 줄 수, 단어 수, 바이트 수 출력

`less [file]`: 좀 더 깔끔하게 출력, vim과 유사

4. Utility Commands

`vim [file]`: CLI 에디터 실행

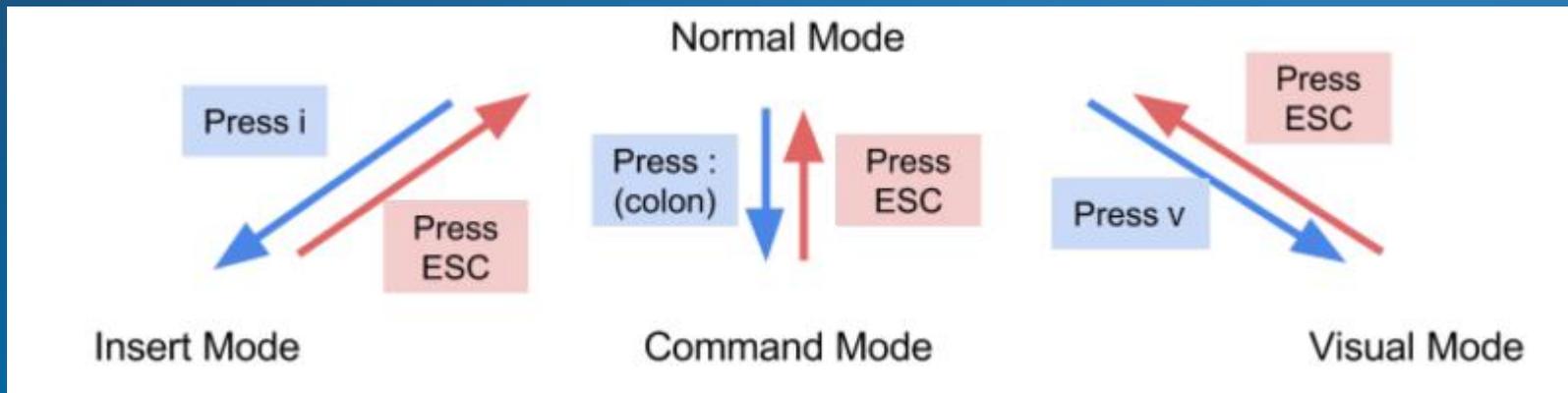
less의 강력한 검색, 인덱싱 기능 + 편집 기능

i: insert mode, esc로 탈출

u: undo

:[number]: [number] 줄 찾기 (:wq, :q!, :w [name])

/[regex]: [regex] 내용 검색



4. Utility Commands

`tar [target]`: 압축/압축해제

`-c` (create): 압축하기

`-x` (extract): 압축 풀기

`-v` (verbose): 진행상황 출력

`-z` (gzip): gzip으로 압축/해제

`-f` (file): 압축/해제 결과를 파일로 출력

```
tar -cvzf [source] [target]
```

```
tar -xvzf [target]
```

4. Utility Commands

find ([dir]): [dir]에서 원하는 파일 검색

-name [name]: 파일명이 [name]인 파일 검색

grep [regex] [file]: global regular expression print

정규식에 해당하는 줄 찾아 출력 (<https://regex101.com/>)

특정 단어가 들어갈 파일의 열 찾는데 많이 사용 => 파이프라인 (|) 과 조합

-i: 대소문자 구분 안함

-n: 파일에서의 줄 번호도 출력

-r: 재귀적으로 하위 파일 탐색

5. Process Commands

ps: **p**roces**s**, 현재 실행되고 있는 사용자의 프로세스 출력

-e: 모든 프로세스 출력

-f: 완전한 출력

PID: 프로세스가 할당받는 고유 번호

PPID: 프로세스를 실행한 부모 프로세스의 PID

5. Process Commands

`fg [process]`(foreground), `bg [process]`(background): 프로세스를 foreground, background로 옮김

`[command] &`: background에서 [command] 실행

`nohup [command]`: no hangups, [command]의 stdout을 ./nohup.out 파일로 리다이렉션 후 shell과 독립적으로 실행

`nohup [command] &`: [command]를 background에서 shell과 독립적으로 실행

Ctrl + Z: 현재 shell을 점유하고 있는 프로세스를 background로 보냄

`jobs`: background process 모두 출력

6. Network Commands

wget [http-address] (**w**eb **g**et): [http-address] 파일을 현재 디렉토리에 같은 이름으로 다운로드

ping [address]: [address]로 ping을 전송하여 되돌아오는 시간을 반복적으로 측정
보통 DNS 설정이 맞지 않을 경우 ping 8.8.8.8 등으로 테스트

ifconfig: 장치의 네트워크 연결 정보 출력

```
ubuntu@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:3c:73:32
          inet6 addr: fe80::20c:29ff:fe3c:7332/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:94 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:20948 (20.9 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1312 (1.3 KB)  TX bytes:1312 (1.3 KB)
```

7. Package & Service Management Commands

리눅스 배포판들은 응용 프로그램들을 관리하기 위한 package manager 명령어들을 가지고 있음

Debian 계열 (Ubuntu 등): dpkg, apt, apt-get

Redhat 계열 (CentOS 등): rpm, yum

apt (advanced package tool) (Ubuntu)

apt update: package 목록 업데이트

apt install [package]: [package]와 dependency들 설치

apt remove [package]: [package]와 dependency들 삭제

apt purge [package]: dependency, 환경설정 파일까지 삭제

apt search [regex]: [regex]와 매치되는 package 검색

7. Package & Service Management Commands

Daemon: 시스템에 상주하며 특정 상태(주로 부팅)가 되면 자동으로 동작하는 프로세스

지속적인 서비스 제공을 위하여 OS에서 직접 관리

백그라운드에서 동작하며 치명적인 오류가 없을 경우 항상 작동

부팅 시 작동해야 하는 daemon들은 /etc/init.d에 정의됨

sshd, httpd, apache2, mysqld, ...

systemctl (**system control**) / **service**: daemon 관리 명령어

systemctl인지 **service**인지는 배포판마다 다름

start [daemon]: daemon 시작

stop [daemon]: daemon 중지

restart [daemon]: daemon 재시작

8. Shell Utilities

`[process-a] | [process-b] (pipe)` [process-a]의 stdout을 [processb]의 stdin으로 연결

ex) `ls -al | grep [regex]`: `ls -al`의 결과 중 [regex]와 일치하는 줄 출력

Redirection: stdin/stdout/stderr을 특정 파일로 redirect

`[command] > [file]`: [command]의 stdout이 [file]로

`[command] < [file]`: [file]이 [command]의 stdin으로

`[command] 2> [file]`: [command]의 stderr이 [file]로

Q&A

참고 자료

Appleseed 님의 2020 겨울 힐 linux 세미나 자료
및 sparcs 내 linux 세미나 자료