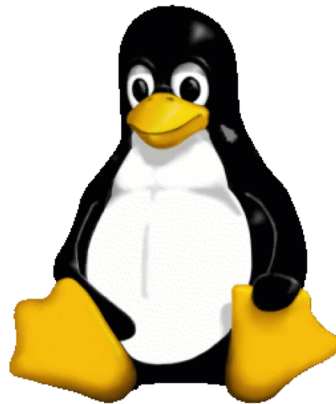


Linux Commands & Packages



appleseed

Recall OS Structure...

- Shell: 커널의 껍데기 - shell 명령어들을 통해 시스템을 조작
- Shell 명령어만 실행(execute)이 가능한가? X
- executable binary file은 모두 실행 가능
- Shell(bash, csh, zsh, ...) 또한 binary file, 입력받은 명령어를 실행하는 명령어일 뿐
- Shell은 어떻게 실행할 명령어의 위치를 찾아낼까?

Environment Variable and PATH

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

- 환경변수(environment variable): 프로세스의 동작에 영향을 끼칠 수 있는 변수들
 - echo \$[name]: 현재 bash가 사용하고 있는 환경변수 [name]의 값을 출력
 - Assignment는 뒤에서...
- PATH: 가장 대표적인 환경변수, 현재 환경에서 실행할 수 있는 binary file들을 담고 있는 디렉토리들을 : 으로 구분함
 - ex) \$ ls
 - /usr/local/bin에 ls라는 binary file이 존재할 경우 실행 ... X
 - /usr/bin 에 ls라는 binary file이 존재할 경우 실행 ... X
 - /bin에 ls라는 파일이 존재 - /bin/ls 실행

Linux Shell Commands

1. Basic commands
2. File/directory commands
3. User/privilege commands
4. Utility commands
5. Process commands
6. Network commands
7. Package & service management commands
8. Shell utilities

Basic Commands

- `man [command]` (**man**ual): [command]에 대한 설명 출력
- `which [command]`: [command]의 절대 경로를 출력
 - PATH 환경변수에 있는 디렉토리들 중 어디에 명령어가 있는지
 - 종종 명령어가 예상치 못한 위치에 있는 경우 원인 파악 시 사용
- `echo [text]`: stdout으로 [text] 출력
 - 환경변수 값 등을 확인하거나 간단한 파일 편집 시 유용
- `Ctrl + C` (command 실행 중): command 탈출하여 shell로 복귀

File/Directory Commands

- pwd (**p**rint **w**orking **d**irectory): 현재 디렉토리의 절대경로 출력
- ls [directory] (**l**ist): [directory] 내의 파일/디렉토리 출력
 - [directory] 없는 경우 현재 디렉토리 내 파일/디렉토리 출력
 - -a: 숨김 파일 포함 출력
 - -l: 상세 정보 출력 (파일/디렉토리 종류, 권한, 수정 날짜 등)
- cd [directory] (**c**hange **d**irectory): [directory]로 이동
 - cd ~: 현재 사용자의 홈 디렉토리로 이동 (/etc/passwd에 정의됨)
- touch [file]: 파일의 변경 날짜 현재로 업데이트, [file] 없을 시 빈 파일 생성

File/Directory Commands

- mv [a] [b] (**m**ove): 파일/디렉토리 옮기기 - a를 b로
 - a의 이름을 b로 바꾸고 싶을 때도 사용
- cp [a] [b] (**c**opy): 파일 복사 - a를 b로
 - -r (recursive): 재귀적으로 복사, 디렉토리 복사 시 필요
- rm [a] (**r**emove): 파일 삭제
 - -r (recursive): 재귀적으로 삭제
 - -f (force): 경고 없이
 - rm -rf /: 나는 내 리눅스 컴퓨터를 벽돌로 만들고 싶어요!

File/Directory Commands

- mkdir [a] (**make directory**): [a] 이름의 디렉토리 생성
 - -p (parents): 부모 디렉토리가 존재하지 않을 경우 같이 생성
- rmdir [a] (**remove directory**): [a] 이름의 디렉토리 삭제
 - 디렉토리가 비어있어야 사용가능하므로 rm -r이 더 유용
- du [file] (**disk usage**): 파일/디렉토리의 용량 출력
 - -s (summarize): 해당 디렉토리만, 없으면 recursive
 - -h (human-readable): 사람이 읽기 편하게 써줌

File/Directory Commands

- 리눅스 파일/디렉토리의 접근 권한 구조 - ls -al 로 확인
- [type]rwxrwxrwx [owner] [group] [size]
 - [type]: d (directory), l (link), - (normal file), ...
 - r (read), w (write), x (execute) 세 개의 비트로 권한을 표시
 - 첫 세 글자는 소유자, 두 번째는 같은 그룹의 사용자, 세 번째는 모두

```
wheelseminar@tong:/home$ ls -al
total 44
drwxr-xr-x  8 root          root          4096 Sep 12  2018 .
drwxr-xr-x 23 root          root          4096 Jul 29  2018 ..
-rw-r--r--  1 root          root         12288 Aug 23  2018 .paper-docker-compose.yml.swo
drwxr-xr-x  2 ara           ara           4096 Sep 11  2018 ara
drwxr-xr-x  3 root          root          4096 Aug 25  2018 jara
drwxr-xr-x  6 wheel         root          4096 Aug 30  2018 paper
drwxr-xr-x  7 wheel         wheel         4096 Jun  9 05:02 wheel
drwxr-xr-x  8 wheelseminar wheelseminar  4096 Jun 29 01:25 wheelseminar
```

File/Directory Commands

- `chmod [privilege] [file]` (**change file mode**): [file] 접근 권한 변경
 - [privilege]: 세 자리 8진수 숫자로 세 종류의 사용자에게 대한 권한 표시
 - ex) `rwxr-xr-- = 754, chmod 754 test.c`
- `chown [user].[group] [file]` (**change owner**): [file] 소유자 변경
 - ex) `chown wheel.root test.c`
- 두 명령어 모두 재귀적으로 수행하는 `-R` 옵션 자주 사용됨

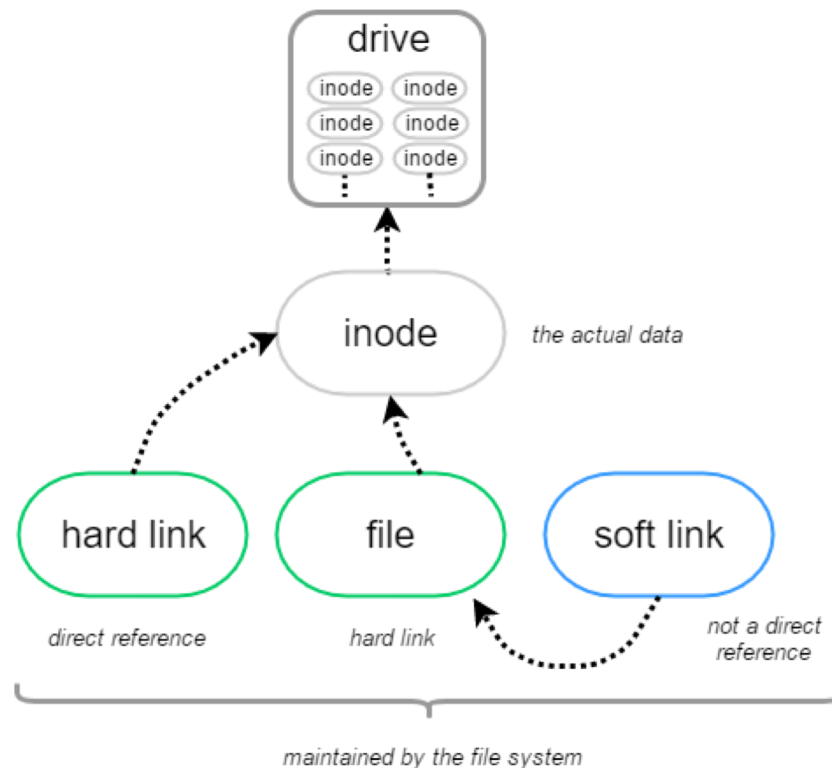
File/Directory Commands

- In [target] [name] (**link**): [target] 파일로 [name] 이름의 바로가기 생성
 - -s (symbolic): 심볼릭 링크/소프트 링크 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
```

File/Directory Commands

- hard link로는 원본 파일의 목적을 확인하기 힘들기 때문에 soft link, symbolic link가 많이 사용됨
- 명령어를 버전별로 관리 or 목적 지정하기 용이



File/Directory Commands

- hard link로는 원본 파일의 목적을 확인하기 힘들기 때문에 soft link, symbolic link가 많이 사용됨
- 명령어를 버전별로 관리 or 목적 지정하기 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      9 Jan 24  2017 python -> python2.7
lrwxrwxrwx 1 root root      16 Jan 24  2017 python-config -> python2.7-config
lrwxrwxrwx 1 root root      9 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      9 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
```

User/Privilege Commands

- whoami: 나의 유저 이름 출력
- su [user] (substitute user): 다른 사용자로 전환
 - [user] 없을 경우 root로 전환
- sudo [command] (substitute user do): 다른 사용자 권한으로 명령어 실행
 - 보통 특정 사용자를 입력하지 않고 root의 권한을 얻기 위해 사용
 - sudoers 되기 (sudo 사용할 수 있게 되기)
 - /etc/sudoers 파일을 수정하여 유저 추가
 - sudo 그룹에 유저 추가
- passwd (password): 사용자의 비밀번호 변경
 - 보통 특정 사용자를 입력하지 않고 본인의 비밀번호를 변경

User/Privilege Commands

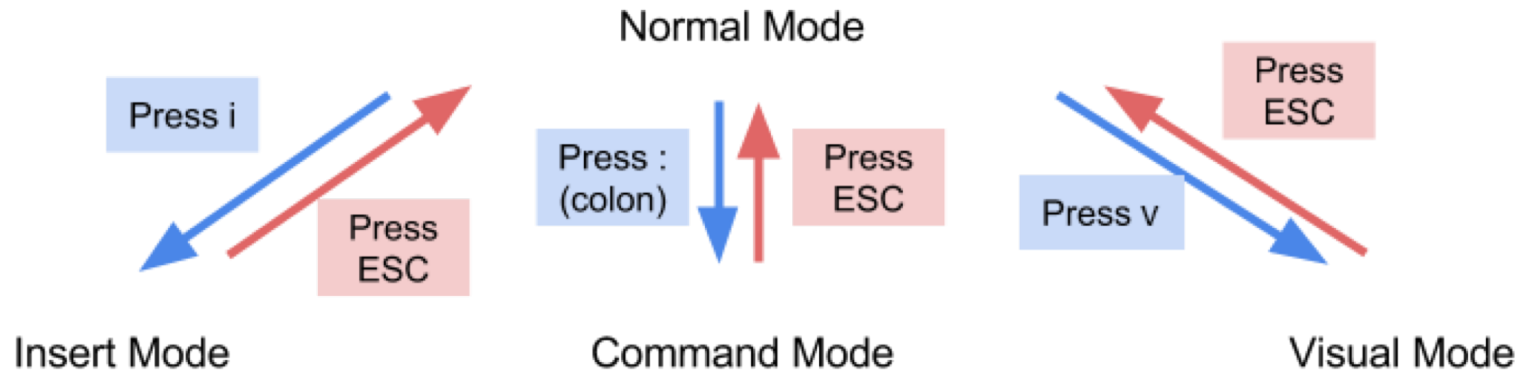
- adduser [name]: [name] 이름의 사용자 추가
 - adduser [name] [group]: [name] 이름의 사용자 [group]에 추가
- deluser [name] (**delete** ~): [name] 이름의 사용자 삭제
- usermod [name] (**modify** ~): [name] 이름의 사용자 정보 편집
 - 여러 옵션과 조합하여 사용
- 종종 위 세 command는 PATH에 존재하지 않는 경우가 있음
 - /usr/sbin 등에 존재

Utility Commands

- cat [file] (con**cat**enate): [file] 내용 전부 출력
 - head [file]: 파일 시작부터 10줄 출력
 - tail [file]: 파일 끝부터 10줄 출력
 - 둘 다 -n [number] 옵션으로 출력 줄 수 조절 가능
- wc [file] (**w**ord **c**ount): [file]의 줄 수, 단어 수, 바이트 수 출력
- less [file]: [file] 내용을 좀 더 깔끔하게 출력
 - 한 번에 전부 출력하지 않음
 - 검색, 줄 번호 찾아가기 등 용이
 - vim과 조작법 유사

Utility Commands

- vim [file]: [file] 편집하는 CLI 에디터 실행
 - less의 검색, 인덱싱 기능에 강력한 편집 기능 제공
 - i: insert mode, esc로 탈출
 - u: undo
 - :[number]: [number] 줄 찾아가기
 - /[regex]: [regex] 내용 검색
 - regex는 뒤에서 다시 나옵니다



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
- 압축 시: tar -cvzf [source] [target]
- 해제 시: tar -xvzf [target]

Utility Commands

- find [dir]: [dir]에서 원하는 파일 검색
 - [dir] 없을 경우 현재 디렉토리가 기준
 - -name [name]: 파일명이 [name]인 파일 검색
- grep [regex] [file] (global regular expression print): regex에 해당하는 줄을 파일에서 찾아 출력
 - -i: 대소문자 구분 안함
 - -n: 파일에서의 줄 번호도 출력
 - -r: 재귀적으로 하위 파일 탐색

Utility Commands

- Regular expression: 정규 표현식, 하나의 식으로 여러 문장을 나타낼 수 있음
 - $[w]^*$ 는 문자 0개 이상, $[w]^+$ 는 문자 1개 이상, $.$ 은 임의의 문자 1개, ...
 - 자세한건 CS322를 수강하자!
- grep의 가장 보편적인 사용법은 단순히 특정 단어가 들어간 파일의 열을 찾는 것
 - 후술할 파이프라인과 조합하여 유용하게 사용됨

Process Commands

- ps (**p**rocess): 현재 실행되고 있는 사용자의 프로세스 출력
 - -e: 모든 프로세스 출력
 - -f: 완전한 출력
 - PID: 프로세스가 할당받는 고유 번호
 - PPID: 프로세스를 실행한 부모 프로세스의 PID

```
wheelseminar@tong:~$ ps
  PID TTY          TIME CMD
 27288 pts/0        00:00:00 bash
 27618 pts/0        00:00:00 ps
wheelseminar@tong:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
wheelse+    27288  27287  0 00:46 pts/0        00:00:00 -bash
wheelse+    27619  27288  0 02:37 pts/0        00:00:00 ps -f
```

Process Commands

- fg [process] (foregound), bg [process] (backgound): 프로세스를 foreground, background로 옮김
- [command] &: background에서 [command] 실행
- nohup [command]: [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

Network commands

- ssh [user]@[address] (secure shell): 원격 서버에 안전한 연결로 접속
 - [user]@ 생략 시 현재 장치의 user 명으로 접속
 - -p [port]: 지정한 [port]로 ssh 접속
- scp [file] [user]@[address]:[path] (secure copy): [file]을 원격 서버 [address]의 [path]에 [user] 권한으로 복사 (업로드)
- scp [user]@[address]:[path-to-file] [path]: 현재 장치의 [path]에 [user] 권한으로 [address]의 [path-to-file]을 복사 (다운로드)
 - [user]@ 생략 시 현재 장치의 user 명으로 접속
 - -P [port]: 지정한 [port] 사용
 - -R (recursive): 재귀적으로 복사

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, Ctrl + C로 탈출
 - 보통 DNS 설정이 맞지 않을 경우 ping 8.8.8.8 등으로 테스트
- ifconfig: 장치의 네트워크 연결 정보 출력

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

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 service인지는 배포판마다 다름
 - start [daemon]: daemon 시작
 - stop [daemon]: daemon 중지
 - restart [daemon]: daemon 재시작

Shell Utilities

- [process-a] | [process-b] (pipe): [process-a]의 stdout을 [process-b]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