



## 리눅스 명령 개요

---

순천향대학교 컴퓨터공학과  
이 상 정

리눅스 명령 개요

## 학습 내용

---

1. 파일 작업 명령어
2. 기본 관리 명령어
3. 사용자 관리



---

## 1. 파일 작업 명령어

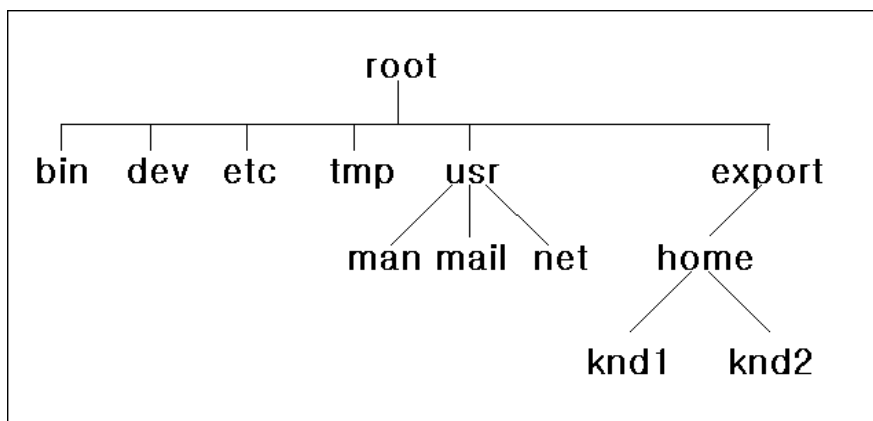
### 리눅스 명령 개요

## 리눅스 파일시스템 구조

---

### □ 트리 구조 파일 시스템

- 파일과 디렉토리의 계층적 구조
- 최상위는 항상 루트(root) 디렉토리에 기반



## 홈 디렉토리

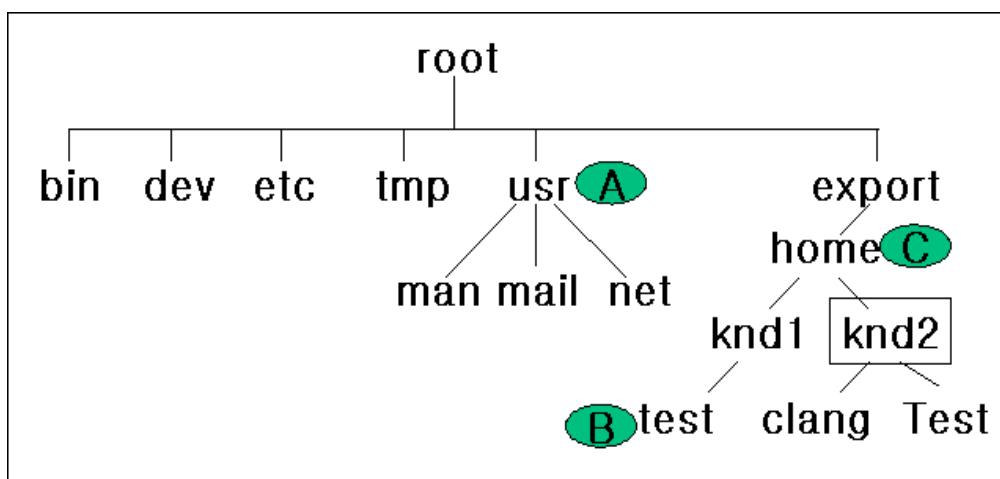
### □ 홈 디렉토리

- 시스템관리자가 사용자를 등록할 때 기본적으로 주어지는 전용 공간
- 기본적으로 `"/home/userID"` 형식으로 생김
- 사용자의 홈 디렉토리 식별하는 `$HOME`이라는 환경변수를 지정
- 홈 디렉토리 환경설정 파일
  - `".profile"` , `".login"` , `".bashrc"` , `".cshrc"`

## 디렉토리 경로

### □

- **절대경로**: 모든 경로 명을 반드시 루트 디렉토리인 `"/`로 시작되어 특정한 파일이나 디렉토리를 표시
- **상대경로**: 현재 작업 디렉토리에 대해 상대적인 경로이름 이용



## 디렉토리 경로

## □ 절대경로

디렉토리	절대경로이름
A	/usr
B	/export/home/knd1/test
C	/export/home/home

## □ 상대경로

디렉토리	상대경로이름
A	../../usr
B	../knd1/test
C	../

## ls 명령어

- 기능 ○ 파일 및 디렉토리의 목록을 출력한다.
- 형 식 ○ ls [옵션] [ {디렉토리 | 파일} ]
- 옵션 ○
  - l : 파일 크기, 저장 날짜, 허가권, 소유권 등을 포함하여 자세히 보기
  - a : 마침표로 시작하는 숨겨진 파일을 포함한 모든 파일 보여주기
  - t : 시간 순서대로 정렬해서 보기
  - R : 하위 디렉토리의 파일까지 재귀적으로 보여주기

## ls 명령어

```

embed@host: ~
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)
embed@host: ~$ ls -l
합계 16
-rw-r--r-- 1 embed embed 167 2009-12-22 11:20 examples.desktop
drwxr-xr-x 7 root root 4096 2009-10-22 16:30 vmware-tools-distrib
drwxr-xr-x 2 embed embed 4096 2009-12-22 11:40 바탕화면
drwxr-xr-x 2 embed embed 4096 2009-12-22 11:36 템플릿
embed@host: ~$ ls -a
.          .dirc      .gvfs      .update-notifier
..         .esd_auth  .local     .xsession-errors
.lCEauthority .gconf     .nautilus  .xsession-errors.old
.bash_history .gconf.d   .profile   examples.desktop
.bash_logout .gnome2    .pulse     vmware-tools-distrib
.bashrc      .gnome2-private .pulse-cookie
.cache       .gnupg     .ssh       바탕화면
.config      .gstreamer-0.10 .sudo_as_admin_successful
.dbus        .gtk-bookmarks .update-manager-core
embed@host: ~$ ls -al
합계 128
drwxr-xr-x 21 embed embed 4096 2009-12-27 15:21 .
drwxr-xr-x 3 root root 4096 2009-12-22 11:20 ..
-rw----- 1 embed embed 2790 2009-12-27 14:47 .lCEauthority
-rw----- 1 embed embed 25 2009-12-26 15:46 .bash_history
-rw-r--r-- 1 embed embed 220 2009-12-22 11:20 .bash_logout
-rw-r--r-- 1 embed embed 3180 2009-12-22 11:20 .bashrc

```

cd 명령어

- 기능 : 디렉토리를 이동한다.
- 형식 : cd [디렉토리명]
- 디렉토리명
  - ~ : 자신의 홈 디렉토리
  - ~(계정 이름) : 다른 사용자의 홈 디렉토리. 예를 들어, cd ~embed일 경우 embed의 홈 디렉토리
  - . : 현재 디렉토리
  - .. : 상위 디렉토리
  - : 바로 직전에 사용한 디렉토리

## cd 명령어

\$ pwd

=> 현재 작업 중인 디렉토리 보기

```
embed@host: ~
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)

embed@host: ~$ pwd
/home/embed
embed@host: ~$ cd /usr
embed@host: /usr$ ls
bin  games  include  lib  local /sbin  share  src
embed@host: /usr$ cd src
embed@host: /usr/src$ pwd
/usr/src
embed@host: /usr/src$ ls
linux-headers-2.6.31-14  linux-headers-2.6.31-14-generic
embed@host: /usr/src$ cd ../../
embed@host: /$ pwd
/
embed@host: /$ cd ~
embed@host: ~$ pwd
/home/embed
embed@host: ~$
```

## 파일의 생성 및 보기

### □ 문서편집기 이용

- vi, emacs, gedit

### □ cat(catenate)와 리다이렉션(redirection)

- 파일 새로 만들기/덮어쓰기
  - \$ cat > test.dat
  - This is a test file.
  - Soonchunhyang University
  - ^D
- 파일에 연속 쓰기
  - \$ cat >> test.dat
  - Computer Engineering
  - ^D
- cat, 파일 내용을 화면에 출력
  - \$ cat test.dat

```
embed@host: ~
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)

embed@host: ~$ cat > test.dat
This is a test file.
Soonchunhyang University
embed@host: ~$ ls
examples.desktop  test.dat  vmware-tools-distrib  바탕화면  템플릿
embed@host: ~$ cat >> test.dat
Computer Engineering
embed@host: ~$ cat test.dat
This is a test file.
Soonchunhyang University
Computer Engineering
embed@host: ~$
```

## 파일 내용 보기

### □ more, 한 번에 한 화면씩 내용 보기

```
$ more /etc/bash.bashrc
```

- 다음 화면 전환 시 <스페이스바> 누름
- 종료 시 <q> 누름

### □ head, tail

- head -n 파일 : 처음 n 줄(default=10) 보기
- tail -n 파일 : 마지막 n 줄 (default=10) 보기

```
$ head /var/log/messages
```

```
$ tail -20 /var/log/messages
```

## 파일 복사

### □ cp(copy), 파일복사

- cp [옵션] 소스파일 목적파일
- 소스파일을 목적파일에 복사

cp의 옵션	설명
-f	강제적인 복사이다. 목표 파일이 이미 존재하고 있어도 확인을 요구하지 않는다.
-i	대화형 복사이다. 각 파일을 복사하기 전에 사용자에게 정말로 복사할 것인지를 확인한다.

```
$ cp test.dat copydata
```

```

embed@host: ~
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)

embed@host: ~$ cp test.dat copydata
embed@host: ~$ ls
copydata  test.dat
examples.desktop  vmware-tools-distrib
embed@host: ~$
embed@host: ~$

```

## 파일 이동 (1)

## □ mv(move), 파일 이동(이름변경)

- mv 소스파일 목적파일
  - 소스파일의 이름을 목적파일의 이름으로 변경
- mv 소스파일들 ..... 디렉토리
  - 소스파일들을 디렉토리로 이동

mv의 옵션	설명
-f	강제적인 이동이다.
-i	대화형 이동이다.

```
$ mv copydata test.txt
```

```
$ mkdir test
```

=> make directory,  
디렉토리 생성

```
$ ls
```

```
$ mv test.dat test.txt test
```

```
$ ls test
```

```
embed@host: ~$ mv copydata test.txt
embed@host: ~$ mkdir test
embed@host: ~$ ls
examples.desktop  test.dat  vmware-tools-distrib  템플릿
test              test.txt  바탕화면
embed@host: ~$ mv test.dat test.txt test
embed@host: ~$ ls test
test.dat  test.txt
embed@host: ~$
```

## 파일 이동 (2)

```
$ cd test
```

=> change directory, 디렉토리 이동

```
$ mv test.* ..
```

=> .. 는 상위 디렉토리, . 는 현재 디렉토리 표시

```
$ ls
```

```
$ cd ..
```

```
$ ls
```

```
$ rmdir test
```

=> remove directory, 디렉토리 제거

```
$ ls
```

```
embed@host: ~$ cd test
embed@host: ~/test$ mv test.* ..
embed@host: ~/test$ ls
embed@host: ~/test$ cd ..
embed@host: ~$ ls
examples.desktop  test.dat  vmware-tools-distrib  템플릿
test              test.txt  바탕화면
embed@host: ~$ rmdir test
embed@host: ~$ ls
examples.desktop  test.dat  test.txt  vmware-tools-distrib  바탕화면  템플릿
embed@host: ~$
```

## 파일 삭제

## rm 명령어

- 기능 ○ 파일을 삭제한다.
- 형식 ○ rm [옵션] 파일명
- 옵션 ○
  - f : 강제 삭제, 바로 삭제한다.
  - i : 대화형 삭제, 파일 삭제 전에 삭제 여부를 확인한다.
  - r : 재귀적으로 실행, 파일을 삭제할 때 하위 디렉토리를 포함해 모든 파일을 삭제한다.



```

embed@host: ~
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)
embed@host:~$ ls
examples.desktop test.dat test.txt vmware-tools-distrib 바탕화면 템플릿
embed@host:~$ rm test.txt
embed@host:~$ ls
examples.desktop test.dat vmware-tools-distrib 바탕화면 템플릿
embed@host:~$
  
```

## 링크 만들기 (1)

## □ ln, 링크 만들기

- ln [-s] 원본파일 링크파일
- 원본파일을 가리키는 링크 파일 생성
- -s 옵션 시 심볼릭(소프트) 링크
- 링크의 종류
  - 하드 링크(hard link)
    - 원래의 파일과 동일한 i-node를 가리킴
    - i-node 는 파일의 속성과 디스크 상의 위치에 대한 정보를 갖고 있는 커널 구조체
    - 원본 파일 이동 시에도 원본 파일에 접근 가능
  - 심볼릭 링크(symbolic link)
    - 새로운 i-node 생성되어 원본 파일과 연결
    - 다른 파티션 뿐만 아니라 다른 네트워크 장치 상에 있는 파일도 링크
    - 원본 파일 이동 시 접근 못함, 윈도우 바로 가기 유사



## 링크 만들기 (2)

```
$ ln test.dat testlink
$ ln -s test.dat softlink
$ ls
$ cat testlink
$ cat softlink
```

```
embed@host:~$ ln test.dat testlink
embed@host:~$ ln -s test.dat softlink
embed@host:~$ ls
examples.desktop  test.dat  vmware-tools-distrib  템플릿
softlink          testlink  바탕화면
embed@host:~$ cat testlink
This is a test file.
Soonchunhyang University
Computer Engineering
embed@host:~$ cat softlink
This is a test file.
Soonchunhyang University
Computer Engineering
embed@host:~$
```

## 파일 권한 변경하기 (1)

### ❑ chmod , 파일 권한 변경하기

- 파일의 허가권은 네 개의 부분으로 나누어져 있다.

- 파일 타입
- 소유자(user)
- 그룹(group)
- 기타 사용자(others)

문자	허가권	값(8진수)
R	읽기(Read)	4
W	쓰기(Write)	2
X	실행하기(Execute)	1

- 기호로 표시한 허가권 예:

문자	허가권	값
---	허가권이 없음	0
r--	읽기만 가능	4
rw-	읽기/쓰기 가능	6
rwX	읽기/쓰기/실행 가능	7
r-x	읽기/실행 가능	5
--x	실행만 가능	1

## 파일 권한 변경하기 (2)

허가권	숫자 값	설명
-rw-----	600	소유자에게만 읽기/쓰기 허가권이 있음. 대부분 파일은 이렇게 설정된다.
-rw-r--r--	644	소유자에게 읽기/쓰기 허가권이 있고, 그룹과 기타 사용자에게는 읽기 허가권만 있음. 소유자 외의 다른 사용자들은 이 파일을 읽기만 하도록 하고 싶을 때 많이 사용하는 권한이다.
-rw-rw-rw-	666	모든 사용자에게 읽기/쓰기 권한을 부여한다. 이 조합은 시스템의 모든 사람이 파일을 접근하여 수정할 수 있으므로 보안 상 권하고 싶지 않은 허가권이다.
-rwx-----	700	소유자가 읽기/쓰기/실행 허가권이 있음. 소유자가 실행하려는 프로그램 파일에 사용함. (보통 C 또는 C++ 프로그램의 실행 파일에 사용한다.)
-rwxr-xr-x	755	소유자가 읽기/쓰기/실행 허가권이 있음. 다른 모든 사용자는 읽기/실행 허가권이 있음.
-rwxrwxrwx	777	모든 사람이 읽기/쓰기/실행 허가권이 있음. 666 설정과 마찬가지로 피하는 것이 좋다.
-rwx--x--x	711	소유자가 읽기/쓰기/실행 허가권이 있음. 다른 사람은 실행 권한만 있음. 다른 사람이 실행만 하고 복사는 못하게 하고 싶을 때 유용하다.
drwx-----	700	이것은 <b>mkdir</b> 명령어를 사용하여 만든 디렉터리를 나타낸다. 오직 소유자만이 이 디렉터리를 읽기/쓰기 할 수 있고 진입할 수 있다. 모든 디렉터리에는 실행 권한이 적어도 하나는 세팅되어 있어야 진입할 수 있다.
drwxr-xr-x	755	이 디렉터리는 소유자에 의해서만 변경될 수 있다.(즉, 디렉터리 안에 파일이나 서브 디렉터리를 만들 수 있다.) 다른 사용자들은 이 디렉터리로 진입할 수 있고 디렉터리 엔트리들을 읽어 볼 수도 있다.
drwx--x--x	711	모든 사용자들이 디렉터리로 진입할 수는 있지만 기타 사용자들은 디렉터리 엔트리들을 읽어 볼 수는 없다. 따라서, 소유자를 제외한 기타 사용자들은 이 디렉터리 내에서 <b>ls</b> 명령을 수행할 수 없다. 이 디렉터리 내의 파일은 파일 이름을 정확히 아는 사용자가만이 읽을 수 있을 것이다.

표. 파일 허가권 조합

## 권한 변경하기 (3)

❑ **chmod [-R] change ,{change}\* {fileName}+**

- 허가권 변경
- clusterSelection+newPermission (허가권 추가)
- clusterSelection-newPermission (허가권 제거)
- clusterSelection=newPermission (절대적 허가권 할당)
  - clusterSelection: u(user/owner) g(group) o(others) a(all)
  - newPermission: r(read) w(write) x(execute)

## • 예

```
-rw-rw-r-- 2 lee lee 21 3월 22 14:34 test.dat
```

```
$ chmod u+x,g-w,o-r test.dat
```

```
=> -rwxr----- 2 lee lee 21 3월 22 14:34 test.dat
```

## 권한 변경하기 (4)

### □ chmod [-R] octal {fileName}+

- 8진수로 권한 설정
- 예
 

```
-rw-rw-r-- 2 lee lee 21 3월 22 14:34 test.dat
$ chmod 740 test.dat
=> -rwxr----- 2 lee lee 21 3월 22 14:34 test.dat
```

```
embed@host: ~$ ls -l
합계 20
-rw-r--r-- 1 embed embed 167 2009-12-22 11:20 examples.desktop
lrwxrwxrwx 1 embed embed 8 2009-12-27 16:24 softlink -> test.dat
-rw-r--r-- 1 embed embed 67 2009-12-27 15:51 test.dat
drwxr-xr-x 7 root root 4096 2009-10-22 16:30 vmware-tools-distrib
drwxr-xr-x 2 embed embed 4096 2009-12-22 11:40 바탕화면
drwxr-xr-x 2 embed embed 4096 2009-12-22 11:36 템플릿
embed@host: ~$ chmod g+wx,o-r test.dat
embed@host: ~$ ls -l test.dat
-rwxrwx--- 1 embed embed 67 2009-12-27 15:51 test.dat
embed@host: ~$ chmod 755 test.dat
embed@host: ~$ ls -l
합계 20
-rw-r--r-- 1 embed embed 167 2009-12-22 11:20 examples.desktop
lrwxrwxrwx 1 embed embed 8 2009-12-27 16:24 softlink -> test.dat
-rwxr-xr-x 1 embed embed 67 2009-12-27 15:51 test.dat
```

21

## 파일 찾기

### □ find

- 디렉토리 트리에서 파일을 찾고자 할 때 사용
- 검색을 위한 다양한 조건을 명식 가능
- 일반적인 사용법
  - \$ find start\_dir [options], start\_dir: 탐색을 시작할 디렉토리
- 예1: /etc 이하 디렉토리에서 "pass"로 시작하는 파일의 경로 찾기
 

```
$ find /etc -name pass* -print
/etc/passwd
/etc/pam.d/passwd
.....
```
- 예2: /usr/src 이하 디렉토리에서 이름이 core인 모든 파일을 찾아 삭제
 

```
$ find /usr/src -name core -exec rm
```

### □ locate

- 파일이름으로 간단하게 파일을 찾는 명령

```
embed@host: ~$ locate passwd
/etc/passwd
/etc/passwd-
/etc/pam.d/chpasswd
```

## 파일/명령어 위치 보기

### □ which, 파일의 위치 보기

- 인자로 준 파일을 찾기 위해 경로를 검색
- 찾고자 하는 파일의 전체 경로(full path)를 알아 볼 때 유용함
- find 명령어와의 차이점
  - which는 환경 변수에 설정되어 있는 실행 경로(path)만 검색
- 예
 

```
$ which passwd
/usr/bin/passwd
```

### □ whereis , 명령어 위치 보기

- 환경 변수에 등록되어 있는 각종 경로를 검색하여 명령어 및 매뉴얼 위치 검색
- 예
 

```
$ whereis useradd
useradd: /usr/sbin/useradd
/usr/share/man/man8/useradd.8.gz
```

## 리다이렉션(Redirection)

### □ 프로그램의 (표준)출력을 파일로 보내거나 입력하는 기능

- overwrite (>), append (>>) / input(<)
- 예

```
$ ls > test.out
$ ls
$ cat test.dat >> test.out
$ grep "root" < /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

- grep "string" file
  - file 에서 string 이 포함된 줄을 출력

```
embed@host:~$ ls > test.out
embed@host:~$ ls
examples.desktop test.dat vmware-tools-distrib 템플릿
softlink test.out 바탕화면
embed@host:~$ cat test.dat >> test.out
embed@host:~$ cat test.out
examples.desktop
softlink
test.dat
test.out
vmware-tools-distrib
바탕화면
템플릿
This is a test file.
Soonchunhyang University
Computer Engineering
embed@host:~$ grep "root" < /etc/passwd
root:x:0:0:root:/root:/bin/bash
embed@host:~$
```

## 파이프(pipe)와 다중 명령어

- 프로그램의 (표준)출력을 다른 프로그램의 (표준)입력으로 보낼 수 있는 메커니즘

- | 로 표시

- 예

```
$ ls /etc | more
```

```
$ ls /etc | grep "pass "
```

```
passwd
```

```
passwd-
```

```
embed@host: ~$ ls /etc | grep "passwd"
passwd
passwd-
embed@host: ~$
```

- 세미콜론(;)을 사용하여 같은 라인에 여러 개의 명령어들을 한꺼번에 입력

- 예

```
$ ls -l
```

```
$ cat /etc/passwd
```

```
=> $ ls -l; cat /etc/passwd
```

## 백그라운드 명령

- 백그라운드(background)

- 명령 끝에 & 입력

- 프로세스(명령)의 종료 기다리지 않고 바로 다음 명령 기다리는 프롬프트 상태

- 실행이 긴 프로세스(명령)에 유용

- & 없는 일반 명령 실행은 포그라운드(foreground) 라 함

```
[Student@localhost ~]$ find -name a* &
```

```
[2] 2969
```

```
./.gconf/apps
```

```
./.gconf/apps/panel/applets
```

```
./.gconf/desktop/gnome/accessibility
```

```
./.gnome2/accels
```

```
[Student@localhost ~]$ ls
```

```
Desktop
```

```
[2]- Done
```

```
find -name a*
```

```
[Student@localhost ~]$
```

## 프로세스 상태 확인

## ❑ 프로세스 상태 확인 (ps)

일반형식	ps [-] [ options ]
주요옵션	e,A: 모든 프로세스 f: 절대경로 표시 l: 긴 형식으로 표시 u: 실행 유저와 실행 시간 표시 j: 작업 제어(job control) 형식으로 표시 s: 시그널 형식으로 표시 m: 메모리 정보 a: 다른 사용자의 프로세스 x: 터미널 제어 없는 프로세스

```
lee@leeVB:~$ ps
  PID TTY          TIME CMD
 1959 pts/2        00:00:00 bash
 2338 pts/2        00:00:00 ps
lee@leeVB:~$ ps u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
lee       1959  0.0  0.5  8344  2872 pts/2    Ss   10:42   0:00 bash
lee       2339  0.0  0.2  6148  1160 pts/2    R+   11:21   0:00 ps u
lee@leeVB:~$ ps ax | more
  PID TTY          STAT TIME  COMMAND
   1 ?           Ss    0:00  /sbin/init
   2 ?           S      0:00  [kthreadd]
   3 ?           S      0:00  [ksoftirqd/0]
   5 ?           S      0:00  [kworker/u:0]
```

## 프로세스 종료시키기

## ❑ 프로세스 종료시키기 (kill)

일반형식	kill [-s signal ] process_id
주요옵션	-s 또는 -number : 프로세스에 시그널 번호를 보낸다 -9 를 사용할 경우 프로세스를 강제로 종료시킨다

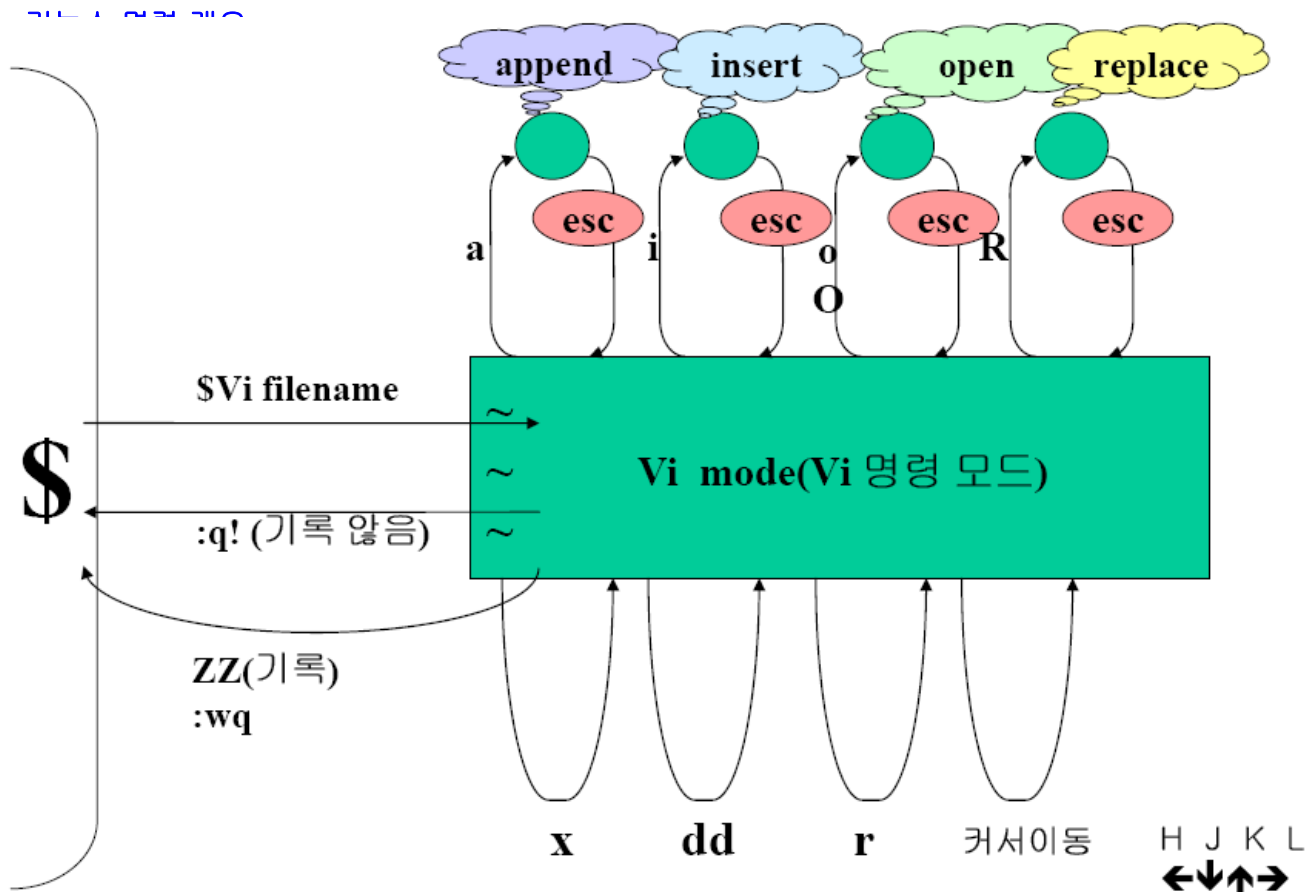
```
lee@leeVB:~$ sleep 1000 &
[1] 2343
lee@leeVB:~$ ps
  PID TTY          TIME CMD
 1959 pts/2        00:00:00 bash
 2343 pts/2        00:00:00 sleep
 2344 pts/2        00:00:00 ps
lee@leeVB:~$ kill -9 2343
[1]+  죽었음                sleep 1000
lee@leeVB:~$ ps
  PID TTY          TIME CMD
 1959 pts/2        00:00:00 bash
 2345 pts/2        00:00:00 ps
lee@leeVB:~$ █
```

## □ 화면 단위 편집기

- vi의 시작  
\$ vi filename  
\$ vi -R filename (읽기 전용)

## □ 프로그램의 상태가 명령모드와 입력모드로 구분

- 명령모드는 편집기 제어명령
- 입력모드는 실제 문자 입력
- 처음 시작 시 명령모드로 진입
- 명령모드에서 입력모드 변환은 삽입명령 후 전환
- 입력모드에서 명령모드는 <ESC> 로 전환
- 이후 명령모드 키입력은 <key>로 표시



## 시작 및 종료

```
$ vi test.txt
<i> This is test text file <ESC>
<Z><Z>
$
```

### □ 삽입명령

```
<i> 커서 위치 삽입
<o> 다음 줄 삽입
```

## 커서 이동 및 삭제 명령

```
<h>
<l>
<j>
<k>

<x>
<dd>
<yy>
<p>
```



```
$ vi test.txt
```

```
<I> <I> <I> <I> <I> <I> <I>
```

```
This is <i>a <ESC><o>test text file
```

```
Using vi editor is not easy but fun.<ESC><y><y>
```

```
<p>
```

```
Using vi editor is not easy but fun.
```

```
<d><d>
```

```
<Z><Z>
```

```
$
```

---

## 2. 기본 관리 명령어

## 수퍼유저 권한 사용

- ❑ 리눅스 시스템의 많은 기능들을 수퍼유저라 불리는 root 계정에만 제한
- ❑ 수퍼유저 권한 사용 방법
  - root 계정으로 로그인
  - su 명령 사용하여 root 계정 권한 획득
  - sudo 명령어 사용해 root 권한으로 명령어 실행
- ❑ 사용 예
  - test.dat 파일의 사용자 소유권 변경 예
  - **chown username filename**
    - 파일 filename의 소유권을 사용자 username으로 지정

## sudo 사용 예

```

embed@host: ~
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)

embed@host:~$ ls -l
합계 20
-rw-r--r-- 1 embed embed 167 2009-12-22 11:20 examples.desktop
lrwxrwxrwx 1 embed embed 8 2009-12-27 16:24 softlink -> test.dat
-rwxr-xr-x 1 embed embed 67 2009-12-27 15:51 test.dat
drwxr-xr-x 7 root root 4096 2009-10-22 16:30 vmware-tools-distrib
drwxr-xr-x 2 embed embed 4096 2009-12-22 11:40 바탕화면
drwxr-xr-x 2 embed embed 4096 2009-12-22 11:36 템플릿

embed@host:~$ chown root test.dat
chown: `test.dat'의 소유자 변경: Operation not permitted
embed@host:~$ sudo chown root test.dat
[sudo] password for embed:
embed@host:~$ ls -l
합계 20
-rw-r--r-- 1 embed embed 167 2009-12-22 11:20 examples.desktop
lrwxrwxrwx 1 embed embed 8 2009-12-27 16:24 softlink -> test.dat
-rwxr-xr-x 1 root embed 67 2009-12-27 15:51 test.dat
drwxr-xr-x 7 root root 4096 2009-10-22 16:30 vmware-tools-distrib
drwxr-xr-x 2 embed embed 4096 2009-12-22 11:40 바탕화면
drwxr-xr-x 2 embed embed 4096 2009-12-22 11:36 템플릿
embed@host:~$
  
```

## APT 패키지 관리 도구

- 우분투에서는 패키지를 설치 및 관리를 위해 **APT (Advanced Package Tool)** 소프트웨어 관리 도구를 사용
  - 패키지 저장소(repository)에서 패키지를 다운로드 받고 설치
- 패키지 저장소(repository)
  - 패키지과 패키지 정보 등을 저장하여 집중 관리하는 서버
  - 저장소의 위치(패키지 인덱스 정보)는 **/etc/apt/sources.list** 파일에 저장
  - 저장 내용 형식  
**deb[deb-src] URI release section**

```
deb http://kr.archive.ubuntu.com/ubuntu/ precise main restricted
deb-src http://kr.archive.ubuntu.com/ubuntu/ precise multiverse
```

## APT 패키지 관리 명령

- **apt-get install *package***
  - 패키지를 다운로드 받아 설치
- **apt-get source *package***
  - 소스코드 다운로드
- **apt-get remove *package***
  - 설치된 패키지를 삭제
- **apt-get update**
  - 패키지 인덱스 정보 업데이트, 인덱스는 **/etc/apt/sources.list** 에 저장
- **apt-get upgrade**
  - 설치되어 있는 패키지를 검사하여 최신 버전으로 업그레이드
- **apt-get search *package***
  - 설치되어 있는 패키지 검색
- **apt-cache pkgnames** 또는 **dpkg -l**
  - 시스템에 설치되어 있는 모든 패키지 목록을 보여줌
- **apt-cache stats**
  - 설치된 패키지의 상태정보 출력
- **apt-cache show *package***
  - 패키지에 대한 소프트웨어 정보 출력

```
embed@host: ~  
파일(E) 편집(E) 보기(V) 터미널(T) 도움말(H)  
embed@host: ~$ x86i nfo  
'x86i nfo' 프로그램은 현재 설치되어 있지 않습니다. 다음을 입력하여 이를 설치할 수 있습니다:  
sudo apt-get install x86i nfo  
x86i nfo: command not found  
embed@host: ~$ sudo apt-get install x86i nfo  
[sudo] password for embed:  
패키지 목록을 읽는 중입니다... 완료  
의존성 트리를 만드는 중입니다  
상태 정보를 읽는 중입니다... 완료  
다음 새 패키지를 설치할 것입니다:  
x86i nfo  
0개 업그레이드, 1개 새로 설치, 0개 지우기 및 161개 업그레이드 안 함.  
40.6k바이트 아카이브를 받아야 합니다.  
이 작업 후 156k바이트의 디스크 공간을 더 사용하게 됩니다.  
받기: 1 http://kr.archive.ubuntu.com/karmic/universe/x86i nfo 1.24-1ubuntu1 [40.6kB]  
내려받기 40.6k바이트, 소요시간 1초 (33.4k바이트/초)  
전에 선택하지 않은 x86i nfo 패키지를 선택합니다.  
(데이터베이스 읽는중 ... 현재 1180357개의 파일과 디렉토리가 설치되어 있습니다.)  
x86i nfo 패키지를 푸는 중입니다 (.../x86i nfo_1.24-1ubuntu1_1386.deb에서) ...  
man-db에 대한 트리거를 처리하는 중입니다 ...  
x86i nfo (1.24-1ubuntu1) 설정하는 중입니다 ...  
embed@host: ~$
```

```
embed@host: ~$ x86i nfo  
x86i nfo v1.24. Dave Jones 2001-2009  
Feedback to <davej@edhat.com>.  
  
Found 1 CPU  
-----  
EFamily: 0 EModel: 1 Family: 6 Model: 23 Stepping: 6  
CPU Model: Core 2 Duo P8600  
Processor name string: Intel(R) Core(TM)2 Duo CPU T8100 @ 2.10GHz  
Type: 0 (Original OEM) Brand: 0 (Unsupported)  
Number of cores per physical package=2  
Number of logical processors per socket=1  
Number of logical processors per core=0  
API C ID: 0x0 Package: 0 Core: 0 SMT ID 0  
embed@host: ~$
```

## 리눅스 명령 개요

/etc/apt/sources.list

```
sjlee@beak-du: ~  
# deb http://kr.archive.ubuntu.com/ubuntu/ precise main restricted  
# deb http://kr.archive.ubuntu.com/ubuntu/ precise-updates main restricted  
# deb http://security.ubuntu.com/ubuntu precise-security main restricted  
  
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to  
# newer versions of the distribution.  
deb http://kr.archive.ubuntu.com/ubuntu/ precise main restricted  
deb-src http://kr.archive.ubuntu.com/ubuntu/ precise main restricted  
  
## Major bug fix updates produced after the final release of the  
## distribution.  
deb http://kr.archive.ubuntu.com/ubuntu/ precise-updates main restricted  
deb-src http://kr.archive.ubuntu.com/ubuntu/ precise-updates main restricted  
  
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu  
## team. Also, please note that software in universe WILL NOT receive any  
## review or updates from the Ubuntu security team.  
deb http://kr.archive.ubuntu.com/ubuntu/ precise universe  
deb-src http://kr.archive.ubuntu.com/ubuntu/ precise universe  
deb http://kr.archive.ubuntu.com/ubuntu/ precise-updates universe  
deb-src http://kr.archive.ubuntu.com/ubuntu/ precise-updates universe  
  
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu  
## team, and may not be under a free licence. Please satisfy yourself as to  
## your rights to use the software. Also, please note that software in  
## multiverse WILL NOT receive any review or updates from the Ubuntu  
## security team.  
deb http://kr.archive.ubuntu.com/ubuntu/ precise multiverse  
deb-src http://kr.archive.ubuntu.com/ubuntu/ precise multiverse  
deb http://kr.archive.ubuntu.com/ubuntu/ precise-updates multiverse  
deb-src http://kr.archive.ubuntu.com/ubuntu/ precise-updates multiverse  
  
## N.B. software from this repository may not have been tested as  
## extensively as that contained in the main release, although it includes  
## newer versions of some applications which may provide useful features.  
--More--(58%)
```

## 저장소 추가

## ❑ /etc/apt/sources.list에 저장소를 추가

- 추가하기 전 기존 저장소 백업  
\$ sudo cp /etc/apt/sources.list /etc/apt/sources.list.backup
- 추가 후 업데이트  
\$ sudo apt-get update

## ❑ 추가 방법

- 기존의 파일의 주석(#)을 제거
- add-apt-repository 명령 사용  
\$ sudo add-apt-repository " deb http://kr.archive.ubuntu.com/ubuntu/ precise main restricted"
- ppa(personal package archive) 추가
  - 우분투 공식 패키지 저장소에 없는 개인 패키지 저장소  
add-apt-repository ppa:<repository-name>  
\$ sudo add-apt-repository ppa:ondrej/php5

## gzip / gunzip 명령어

## gzip 명령어

- 기    능    ➊ 파일을 압축한다.
- 형    식    ➋ gzip [옵션] 파일명

## gunzip 명령어

- 기    능    ➊ .gz로 압축된 파일의 압축을 푼다.
- 형    식    gunzip [옵션] 파일명

## gzip / gunzip 명령어 사용 예

```

embed@host: ~/test
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)
embed@host:~$ mkdir test
embed@host:~$ cd test
embed@host:~/test$ cp /etc/sensors.conf .
embed@host:~/test$ ls -l
합계 84
-rw-r--r-- 1 embed embed 85602 2009-12-28 22:59 sensors.conf
embed@host:~/test$ gzip sensors.conf
embed@host:~/test$ ls -l
합계 24
-rw-r--r-- 1 embed embed 22487 2009-12-28 22:59 sensors.conf.gz
embed@host:~/test$ gunzip sensors.conf.gz
embed@host:~/test$ ls -l
합계 84
-rw-r--r-- 1 embed embed 85602 2009-12-28 22:59 sensors.conf
embed@host:~/test$

```

## tar 명령어

## tar 명령어

- 기 능 ○ 파일이나 디렉토리를 하나로 묶거나 툰다.
- 형 식 ○ tar 옵션 파일명 [위치]
- 옵션 ○
  - c : 하나의 파일로 묶기(compress)
  - x : 묶인 파일 풀기(extract)
  - v : 파일을 묶거나 풀 때 진행 과정을 자세히 보여줌(verbose)
  - f : 묶음 파일명, tar 명령어를 사용할 때 반드시 사용(file)
  - z : gzip과 관련하여 압축/복원을 동시에 수행

## tar 명령어 사용 예 (1)

```

embed@host: ~
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)

embed@host:~$ ls
exampl es. deskto p test test.out vmware-tool s- dl strib
sof tli nk test. dat
embed@host:~$ mv test.out test.dat test
embed@host:~$ ls test
sensor s. conf test. dat test. out
embed@host:~$ tar -cvf test.tar test
test /
test /test. dat
test /test. out
test /sensor s. conf
embed@host:~$ ls
exampl es. deskto p sof tli nk test test. tar vmware-tool s- dl strib
embed@host:~$ rm -rf test
embed@host:~$ ls
exampl es. deskto p sof tli nk test. tar vmware-tool s- dl strib
embed@host:~$ gzip test.tar
embed@host:~$ ls
exampl es. deskto p sof tli nk test. tar. gz vmware-tool s- dl strib
embed@host:~$

```

## tar 명령어 사용 예 (2)

```

embed@host: ~
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)

embed@host:~$ ls
exampl es. deskto p sof tli nk test. tar. gz vmware-tool s- dl strib
embed@host:~$ tar -zxf test.tar.gz
embed@host:~$ ls
exampl es. deskto p test test. tar. gz vmware-tool s- dl strib
sof tli nk test. tar. gz
embed@host:~$ rm test.tar.gz
embed@host:~$ ls
exampl es. deskto p sof tli nk test vmware-tool s- dl strib
embed@host:~$ ls test
sensor s. conf test. dat test. out
embed@host:~$

```

## □ 도움말 찾기

- `man`
- `Info`
- 예  
`$ man grep`  
`$ info grep`
- 온라인 매뉴얼  
<http://man.he.net/>

```

embed@host: ~
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)
GREP(1) User Commands GREP(1)

NAME
    grep, egrep, fgrep, rgrep - print lines matching a pattern

SYNOPSIS
    grep [OPTIONS] PATTERN [FILE...]
    grep [OPTIONS] [-e PATTERN | -f FILE] [FILE...]

DESCRIPTION
    grep searches the named input FILES (or standard input if no files are
    named, or if a single hyphen-minus (-) is given as file name) for lines
    containing a match to the given PATTERN. By default, grep prints the
    matching lines.

    In addition, three variant programs egrep, fgrep and rgrep are
    available. egrep is the same as grep -E. fgrep is the same as
    grep -F. rgrep is the same as grep -r. Direct invocation as either
    egrep or fgrep is deprecated, but is provided to allow historical
    applications that rely on them to run unmodified.

OPTIONS
    Generic Program Information
    Manual page grep(1) line 1
  
```

## 3. 사용자 관리



## 사용자

- 리눅스에서 모든 파일과 프로그램은 어떤 사용자(user)에 의해 소유
  - 사용자 ID(UID)라 불리는 고유한 숫자로 식별됨.
  - 각 사용자는 적어도 하나의 그룹(시스템 관리자가 만드는 사용자들의 집합)에 소속
- 사용자의 권한
  - 일반 사용자
    - 자신이 접근할 수 있도록 허용된 자원만 접근 가능
  - root 사용자(root user): 슈퍼유저
    - 모든 자원에 액세스 가능
- **/etc/passwd** 파일에 사용자 계정 정보 등록

## 홈 디렉토리(Home Directories)

- 각 사용자가 다른 사용자의 구성 환경을 방해하지 않으면서(여러 사용자가 시스템에 동시에 로그인 할지라도) 자신만의 환경에서 실행할 수 있도록 해주는 역할
  - 자신들의 설정 파일 / 일상적인 작업 파일을 저장.
- 홈 디렉토리의 위치
  - 일반적으로 **/home** 에 각 사용자들의 로그인 이름의 디렉토리 예) **/home/lee**
  - 사용자의 홈 디렉토리는 관리자가 변경 가능
    - **/etc/passwd** 파일에 등록
    - **useradd** 명령의 **-d** 옵션 사용하여 변경

## 셸(shell) (1)

## □ 셸이란?

- 명령어 라인 해석기(command-line interpreter, CLI)
- 운영체제가 수행할 명령어를 직접 입력하는 수단 제공

## □ 셸 프로그램의 종류

- Bourne shell: /bin/sh
- Bourne-Again shell: /bin/bash
- Korn shell: /bin/ksh (Bourne Shell 포함)
- C shell: /bin/csh (대화형 작업에 유리)
- 일반적인 프롬프트 구분
  - \$ (Bourne shell, Korn shell)
  - % (C shell)

## 셸(shell) (2)

## □ 셸의 선택

- 디폴트 로그인 셸은 시스템 관리자가 지정
  - /etc/passwd 패스워드 파일에 등록되어 있는 셸 프로그램 의미
- chsh (change shell) 명령으로 디폴트 셸
  - \$ chsh -s /bin/csh  
=> /bin/csh 이 없는 경우 apt-get install csh로 설치
  - 로그 아웃 후 다시 로그인
  - % ps로 확인
  - 다시 bash로 회복하려면 csh대신 bash로 위 과정 반복
- \$SHELL: 로그인 shell 절대경로 저장
  - \$ echo \$SHELL

## 셸 스크립트 (shell script)

- 셸 명령들의 리스트로 작성된 프로그램
- test.sh 예

```
#!/bin/sh    <- 해당스크립트를 실행할 인터프리터의
              절대경로와 실행옵션을 지정
# test shell  <- #은 주석 표시
pwd
ls -l
```

```
$ chmod +x test.sh    <- 실행 권한 부여
$ ./test.sh           <- 스크립트 실행
```

## test.sh 실행 예

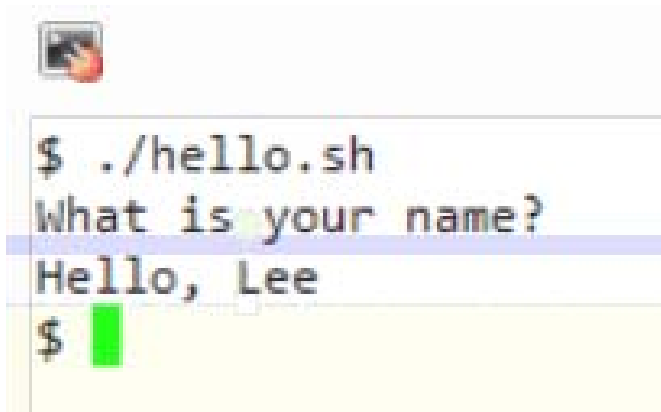
```
$ vi test.sh
$ cat test.sh
#!/bin/sh

# test shell
pwd
ls -l
$ chmod +x test.sh
$ ./test.sh
/home/sjlee/shell
합계 4
-rwxrwxr-x 1 sjlee sjlee 34  2월 28 12:20 test.sh
$ █
```

## hello.sh 예

```
#!/bin/sh
echo "What is your name?"
PERSON="Lee"
echo "Hello, $PERSON"
```

<- 변수 값 지정  
<- 변수 참조



```
$ ./hello.sh
What is your name?
Hello, Lee
$
```

## 시작 스크립트

## □ 시작 스크립트(Startup Scripts)

- 과거 도스(DOS)의 autoexec.bat, config.sys 파일과 유사
- 리눅스 같은 다중 사용자 환경에서는 사용자는 자신만을 위한 환경 설정 파일들을 소유해야 함
- 사용자 별 환경 설정 파일은 셸 스크립트의 형태로 존재
- 시작 스크립트는 사용자가 로그인할 때 셸이 실행할 일련의 명령들이 저장
  - 시작 스크립트 파일은 사용자의 홈 디렉터리에 존재
- 사용자 계정을 만들 때는 사용자가 로그인 시 사용할 기본 닷 파일들을 관리자가 제공해야 한다.
  - 리눅스에 있는 정식 도구를 사용하여 사용자 계정을 만든다면 이 도구들이 자동으로 생성

## bash 시작 스크립트

### □ Bourne-Again shell (bash)

- bash는 기본적인 리눅스 셸이며 root 셸
- 리눅스의 bash는 Unix의 Bourne Shell 문법을 모두 만족하며 그 외에 더 나은 C Shell의 일부분도 포함

### □ bash 시작 스크립트

- `.profile`(또는 `bash_profile`)
  - 로그인 셸로 로그인 시 사용자 인증 후 실행
  - `.bashrc` 셸을 실행하고 명령의 경로 등 지정
- `.bashrc`
  - 비로그인 셸로 예를들어 터미널을 오픈하는 경우처럼 인증이 없이 실행
  - `/etc/bash.bashrc` 셸을 포함(또는 복사)
    - 프롬프트 등의 사용자 기본 환경 설정

## bash 시작 스크립트 예 (1)

### □ 사용자 계정으로 로그인

```
$ pwd
/home/lee
$ ls
$ ls -al
152
drwx----- 11 lee    lee    4096 3    8 17:25 .
drwxr-xr-x  5 root   root   4096 3    5 11:20 ..
-rw-----  1 lee    lee     0 3    8 15:17 .ICEauthority
-rw-----  1 lee    lee     0 3    8 15:17 .Xauthority
drwxr-x---  2 lee    lee   4096 3    8 15:11 .ami
-rw-----  1 lee    lee    708 3    9 09:37 .bash_history
-rw-r--r--  1 lee    lee    24 2   11 2003 .bash_logout
-rw-r--r--  1 lee    lee   201 3    8 17:24 .bash_profile
-rw-r--r--  1 lee    lee   124 2   11 2003 .bashrc
.....
$ ls ..
lee lost+found kim
```

## bash 시작 스크립트 예 (2)

```
$ echo $SHELL
/bin/bash
```

```
$ cat .bash_profile
# .bash_profile
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi
# User specific environment and startup programs
PATH=$PATH:$HOME/bin:/usr/sbin
export PATH
unset USERNAME
```

```
$ cat .bashrc
# .bashrc
# User specific aliases and functions
# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
```

## bash 시작 스크립트 예 (3)

```
$ more /etc/bashrc
# /etc/bashrc

# System wide functions and aliases
# Environment stuff goes in /etc/profile

# by default, we want this to get set.
# Even for non-interactive, non-login shells.
if [ "`id -gn`" = "`id -un`" -a "`id -u`" -gt 99 ]; then
    umask 002
else
    umask 022
fi
.....
```

## /etc/passwd 파일

### □ /etc/passwd 파일은 사용자의 계정 등록 정보를 저장하는 파일

- 사용자의 로그인 이름
- 암호화된 패스워드
- UID (사용자 ID)
- GID (그룹 ID)
- 사용자 이름이나 코멘트
- 홈 디렉토리
- 로그인 셸 경로명

### □ 한 줄에 한 사용자의 정보가 모두 들어 있음

- 각 항목은 콜론(:)으로 구분
- sshah:boQavhhaCKaXg:100:102:Steve Shah:/home/sshah:/bin/bash

## /etc/passwd 파일 예

### \$ cat /etc/passwd

```
root:AgQ/IJgASeW1M:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/adm:
lp:*:4:7:lp:/var/spool/lpd:
sync:*:5:0:sync:/sbin:/bin/sync
shutdown:*:6:0:shutdown:/sbin:/sbin/shutdown
halt:*:7:0:halt:/sbin:/sbin/halt
mail:*:8:12:mail:/var/spool/mail:
news:*:9:13:news:/var/spool/news:
uucp:*:10:14:uucp:/var/spool/uucp:
operator:*:11:0:operator:/root:
games:*:12:100:games:/usr/games:
gopher:*:13:30:gopher:/usr/lib/gopher-data:
ftp:*:14:50:FTP User:/home/ftp:
pop:*:15:15:APOP Admin:/tmp:/bin/tcsh
nobody:*:99:99:Nobody:/:
sshah:Kss9Ere9b1Ejs:500:500:Steve Shah:/home/sshah:/bin/tcsh
hdc:bfCABlvZBIbFM:501:501:H. D. Core:/home/hdc:/bin/bash
jyom:*:502:502:Mr. Yom:/home/jyom:/bin/bash
```

## 기타 계정 등록 관련 파일

### □ /etc/shadow 파일

- 패스워드 엔트리를 /etc/passwd 파일에서 분리하여 /etc/shadow 파일에 관리
- /etc/passwd 파일은 모든 사용자들이 읽을 수 있으나 /etc/shadow 파일은 root 권한이 있는 프로그램만 읽을 수 있음

### □ /etc/group 파일

- 각 사용자는 적어도 하나의 그룹에 속해 있어야 함
- 사용자 그룹에 관한 정보 저장

## 사용자 계정 추가 명령

### □ adduser

- **adduser username**

```
# adduser -h
adduser [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID]
[--firstuid ID] [--lastuid ID] [--gecos GECOS] [--ingroup GROUP | --gid ID]
[--disabled-password] [--disabled-login] [--encrypt-home] USER
    Add a normal user

adduser --system [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID]
[--gecos GECOS] [--group | --ingroup GROUP | --gid ID] [--disabled-password]
[--disabled-login] USER
    Add a system user

adduser --group [--gid ID] GROUP
addgroup [--gid ID] GROUP
    Add a user group

addgroup --system [--gid ID] GROUP
    Add a system group

adduser USER GROUP
    Add an existing user to an existing group

general options:
--quiet | -q      don't give process information to stdout
--force-badname   allow usernames which do not match the
                  NAME_REGEX[_SYSTEM] configuration variable
--help | -h       usage message
--version | -v     version number and copyright
--conf | -c FILE  use FILE as configuration file
```



## 사용자 계정 관련 명령

### 예

- 로그인 이름이 embed인 새로운 사용자를 추가

# adduser embed

```
# adduser embed
Adding user `embed' ...
Adding new group `embed' (1002) ...
Adding new user `embed' (1002) with group `embed' ...
Creating home directory `/home/embed' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for embed
Enter the new value, or press ENTER for the default
  Full Name []: embed
    Room Number []:
    Work Phone []:
    Home Phone []:
      Other []:
Is the information correct? [Y/n] y
#
```

### deluser

- 기존 사용자들을 삭제하기 위한 명령
- deluser username

## shutdown (1)

### /sbin/shutdown

- 여러 가지 필요(시스템 업그레이드 및 패치등)에 의해서 시스템을 종료시켜야하는 경우에 사용

### 관리자가 시스템을 종료할때의 작업순서와 shutdown을 했을 때 커널에서 내부적으로 이루어지는 작업은 다음과 같다

- 1. 접속해 있는 사용자들에게 시스템이 종료된다는 메시지를 전송
  2. 새로운 사용자의 로그인을 금지
  3. 지정된 시간 내에 종료되지 않은 프로세스를 강제 종료
  4. 지정된 시간 내에 로그아웃 하지 않은 사용자를 강제 종료
  5. 메모리 남아있는 데이터를 디스크에 저장(디스크동기화)
  6. 시스템에 종료에 관련된 정보를 시스템 로그파일에 기록
  7. 마운트되어 있는 디바이스들의 마운트를 해제
  8. 시스템을 종료

## shutdown (2)

## □ /sbin/shutdown [-t sec] [-rkhncfF] time [warning-message]

- -k : 실제로 종료하지 않고 모든 사용자에게 경고 메시지만을 전송
- -r : 시스템 종료 후에 재부팅(reboot)
- -h : shutdown 후에 시스템을 종료(halt)
- -f : 재 부팅할 때 fsck(file system check)를 하지 않음
- -c : shutdown명령어를 취소
- time : 몇 분후에 시스템을 종료할 것인가를 지정
- warning-message : 사용자에게 보내질 종료 메시지

## □ 예

- 접속한 사용자에게 경고메시지를 보내고 5분후에 시스템을 종료  
# shutdown -h -t 5 "System down for regular maintenance"
- 시스템을 즉시 종료  
# shutdown -h now
- 시스템을 즉시 재부팅  
# shutdown -r now

## 기타 시스템 종료 명령

## □ halt, reboot, poweroff

- 시스템을 종료하는 명령으로 reboot과 poweroff가 halt 명령에 링크되어 있음
- 형식  
/sbin/halt [-n] [-w] [-d] [-f] [-i] [-p] [-h]  
/sbin/reboot [-n] [-w] [-d] [-f] [-i]  
/sbin/poweroff [-n] [-w] [-d] [-f] [-i] [-h]
  - n : 재부팅이나 종료 전에 sync를 수행하지 않음
    - 현재 메모리의 내용 등을 디스크에 저장하지 않음
  - w: 재부팅이나 종료하지 않고 /var/log/wtmp 파일에 종료 기록만 저장
  - d : wtmp 파일에 기록을 저장하지 않음. -n 옵션은 -d 옵션을 동반
  - f : 강제로 halt나 reboot 수행. shutdown을 호출하지 않음
  - i : 모든 네트워크 인터페이스 종료
  - h : 종료 전에 하드디스크를 준비 모드로 설정
  - p : 종료 시 전원을 끄. 이는 디폴트로 세팅

- ❑ 강의 시간의 실습 내용을 정리하여 제출
- ❑ 앞에서 소개하지 않은 임의 리눅스 명령어 3개를 조사하고 실행
- ❑ API 패키지를 사용하여 설치할 수 있는 유용한 임의의 패키지를 조사하고 설치하여 실행
- ❑ 실습 내용 및 과제 캡처 시 주의 사항
  - 사용자 셸 프롬프트에 자기 이름 표시
  - 프롬프트 변경 예  
\$ export PS1="(이름)\$ "