

copyright 김성우

# LINUX

리눅스  
기본

## 리눅스 콘솔

- 리눅스 시스템을 제어하는 곳으로, 명령어 사용을 통해 시스템을 제어할 수 있음
- 가상 콘솔(가상의 모니터)이며, 우분투는 총 6개의 가상 콘솔을 제공 (2번~7번)
- X윈도 화면은 가상 콘솔의 2번째 화면 (Ctrl + Alt + F2~F7)로 콘솔 화면 전환 가능

```
root@Server:~/바탕화면#
```

root	: 접속중인 사용자 계정 (root:관리자)
@	: 구분자
Server	: 시스템명
~	: 마지막 작업경로
#	: 계정의 유형 (#은 root, \$는 일반 유저)

```
Ubuntu 20.04.4 LTS Server tty3
```

```
Server login: root
```

```
Password:
```

```
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-40-generic x86_64)
```

```
* Documentation:  https://help.ubuntu.com
```

```
* Management:    https://landscape.canonical.com
```

```
* Support:        https://ubuntu.com/advantage
```

```
0 updates can be applied immediately.
```

```
The list of available updates is more than a week old.
```

```
To check for new updates run: sudo apt update
```

```
Your Hardware Enablement Stack (HWE) is supported until April 2025.
```

```
Last login: Sat Jun 11 19:51:55 KST 2022 on tty2
```

```
root@Server:~# _
```

리눅스 명령어

- 내장 명령어: export, history, logout, exit 등, 파일로 존재하지 않고 셸에 내장되어있는 명령어
- 외장 명령어: 리눅스 대부분을 차지하며 /bin 또는 /sbin 디렉터리에 파일 형태로 존재 (프로세스 발생)
- PATH: 사용자들이 실행 명령어를 입력했을 때 찾는 디렉터리의 목록을 저장하고 있는 환경변수

man

- 내장 명령어 관련 정보를 볼 수 있는 명령어
- ```
#man [명령어]  
[SPACE] : 한 페이지 밑으로 내려간다.  
[ENTER] : 한 줄 밑으로 내려간다.  
[b] : 전 페이지로 올라간다.  
[q] : man 명령을 종료한다.
```

echo

- 문자열을 그대로 출력해주는 명령어.  
\$과 함께 사용하면 \$뒤가 변수일 경우 변수의 값을 출력해줌
- ```
#echo "test"  
#echo $PATH
```

Section	주요 내용	설명
man1	User Commands	ls, cp 등과 같은 사용자 명령어 관련한 매뉴얼이 들어있는 영역
man2	System Calls	리눅스 프로그래밍을 위한 시스템 호출과 관련된 내용이 들어있는 영역
man3	C Library Functions	stdio와 같은 리눅스에서 사용하는 C 언어 라이브러리 함수와 관련된 내용이 들어있는 영역
man4	Devices and Special Files	관련 장치나 장치 드라이버, 특수파일(FIFO, 소켓 등)에 대한 내용이 들어있는 영역
man5	File Formats and Conventions	/etc/passwd, /etc/shadow, xinetd.conf 등의 파일 형식과 관련된 내용이 들어 있는 영역
man6	Games et. Al.	게임을 비롯하여 재미있는 프로그램들과 관련된 내용이 들어 있는 영역이다.
man7	Miscellanea	리눅스 시스템 파일 관련 표준, 규칙, 프로토콜, 문자셋, 시그널 목록 등에 대한 정보가 들어 있는 영역
man8	System Administration tools and Daemons	root 사용자가 사용하는 시스템 관리와 관련된 명령 및 유틸리티에 대한 내용이 들어 있는 영역
man9	Linux kernel routines	리눅스에만 있는 영역으로 커널 프로그래밍과 관련된 정보가 들어 있는 영역

history

- 사용자가 사용한 명령어 내역을 볼 수 있는 명령어
- /root/.bash\_history: root 계정의 명령어 내역
- /home/{계정명}/.bash\_history: 해당 사용자 계정의 명령어 내역

옵션/명령어	의미	사용 예
-c	기존 history 전부 삭제	history -c
-d offset	해당 offset에 해당하는 history 제거	history -d 3
숫자	최근 사용한 명령어를 해당 개수만큼 출력	history 10
-w	현재 history를 파일로 저장	history -w HistoryFile
!offset	offset에 해당하는 명령어 실행	!30
!string	string으로 시작하는 가장 최근 명령을 실행	!hi

clear

- 현재 사용 중인 터미널 화면의 내용을 전부 지워줌

시스템 종료 및 재시작

shutdown (시스템 종료)

- 시스템 종료와 관련된 명령어. 예약 가능.  
#shutdown [option] [time] [message]

로그아웃

- #logout  
#exit

옵션/명령어	의미	사용 예
-P +숫자	숫자(분) 이후 종료 (P: poweroff)	shutdown -P +10 (10분 후 종료)
-r 시간	shutdown 후 해당 시간에 재부팅 (r: reboot)	shutdown -r 22:00 (22시에 재부팅)
-c	예약된 shutdown 취소 (c: cacle)	shutdown -c (예약된 shutdown 취소)
-k +숫자	현재 접속한 사용자에게 해당 분 후 종료된다는 메세지만 보내고 실제로는 종료되지 않음 (k: knock)	shutdown -k +15 (15분 후 종료된다고 알림)
-h	shutdown 후 halt 실행	shutdown -h now
-f	재부팅 시 fsck(file system check)를 안하고 빠르게 재부팅	shutdown -f

reboot (시스템 종료)

- #reboot [option]  
#reboot -f: shutdown 단계를 거치지 않고 강제 재시작

halt (시스템 종료)

- #halt [option]  
#halt -p: 시스템 즉시 종료

init (시스템 시작)

#init [option] [run level] (해당 run level에 해당하는 모드로 시스템을 시작)

#startx : X윈도우로 시스템을 재가동

/lib/systemd/system/default.target : 시스템에 설정된 런레벨 확인

/lib/systemd/system/runlevel?.target : 가능한 런레벨 모드 확인

런레벨	영문모드	설명	비고
0	Power Off	종료모드 (HDD 동기화 후 정상 종료)	
1	Rescue	시스템 복구 모드	단일 사용자모드
2	Multi-User	NFS/X-window를 제외한 부팅	사용하지 않음
3	Multi-User	텍스트 모드의 다중 사용자 모드 (CLI)	
4	Multi-User		사용하지 않음
5	Graphical	그래픽 모드의 다중 사용자 모드 (X-window)	
6	Reboot	재부팅 모드 (시스템 재시작)	

## 시스템 부팅 설정

```
root@server-b:~# cd /lib/systemd/system
root@server-b:/lib/systemd/system# ls -l runlevel?.target
lrwxrwxrwx 1 root root 15 Jan 10 04:56 runlevel0.target -> poweroff.target
lrwxrwxrwx 1 root root 13 Jan 10 04:56 runlevel1.target -> rescue.target
lrwxrwxrwx 1 root root 17 Jan 10 04:56 runlevel2.target -> multi-user.target
lrwxrwxrwx 1 root root 17 Jan 10 04:56 runlevel3.target -> multi-user.target
lrwxrwxrwx 1 root root 17 Jan 10 04:56 runlevel4.target -> multi-user.target
lrwxrwxrwx 1 root root 16 Jan 10 04:56 runlevel5.target -> graphical.target
lrwxrwxrwx 1 root root 13 Jan 10 04:56 runlevel6.target -> reboot.target
```

```
root@server-b:/lib/systemd/system# ls -l default.target
lrwxrwxrwx 1 root root 16 Jan 10 04:56 default.target -> graphical.target
```

```
root@server-b:/lib/systemd/system# ln -sf multi-user.target default.target
root@server-b:/lib/systemd/system# ls -l default.target
lrwxrwxrwx 1 root root 17 Jun 11 09:31 default.target -> multi-user.target
```

## 리눅스의 Editor

- 우분투에서 사용 가능한 에디터는 nano, gedit, emacs, vi, vim(vi improved) 등이 존재

### vi (vim)

- 명령모드로 파일 안에서 하고 싶은 작업을 선택한 후 작업하는 것이 특징

### nano

- 기본적인 편집 기능 위에 잘라내기나 붙이기 정도의 기능이 제공되는 편집기

### gedit

- X윈도 모드에서만 사용가능한 GUI버전 편집기

### emacs

- 사용자가 많은 부분을 설정하고 변경할 수 있는 편집기
- vim과 같이 기능이 강력하지만 단축키가 많고 Lisp라는 언어를 알아야 함



```
    :::  
    iLE88Dj. :jd88888Dj:  
    .LGitE888D.f8GjjjL8888E;  
    iE :8888Et. .G8888.  
    ;i E888, ,8888,  
    D888, :8888:  
    D888, :8888:  
    D888, :8888:  
    D888, :8888:  
    888W, :8888:  
    W88W, :8888:  
    W88W, :8888:  
    DGGD: :8888:  
    :8888:  
    :W888:  
    :8888:  
    E888i  
    tW88D
```





vi

- 명령모드로 파일 안에서 하고 싶은 작업을 선택한 후 작업
- 명령모드: 여러가지 단축키를 통해 문서의 내용을 수정하고 편집하는 모드
- 입력모드: 키보드로 입력하는 모든 글자를 문서로 입력하는 모드
- Ex:모드: 명령어를 입력하고 엔터를 쳐서 해당 명령을 수행하는 모드



키	설명	키	설명
Ex 모드			
/문자열 Enter	해당 문자열을 찾음(현재 커서 이후로)	n	찾은 문자 중에서 다음 문자로 이동
:w	문서 저장하기	:q	현재 문서 닫기
:wq	저장하고 닫기	:q!	저장하지 않고 닫기
:숫자	지정한 라인넘버로 이동	:vs	세로로 창 분할
:e 파일명	파일 새로 열기	:%s/old/new/g	old라는 문자열을 new로 치환

# 리눅스 명령어

# 에디터(Editor)

키	설명	키	설명
명령 모드에서 입력 모드로 전환			
i	현재 커서의 위치부터 입력(I)	l	현재 커서 줄의 맨 앞에서부터 입력 (Shift + l)
a	현재 커서의 위치 다음 칸부터 입력(A)	A	현재 커서 줄의 맨 마지막부터 입력 (Shift + A)
o	현재 커서의 다음 줄에 입력(O)	O	현재 커서의 이전 줄에 입력 (Shift + O)
s	현재 커서 위치의 한 글자를 지우고 입력(S)	S	현재 커서의 한 줄을 지우고 입력 (Shift + S)

키	설명	키	설명
명령 모드에서 삭제, 복사, 붙여넣기 관련 키			
x	현재 커서가 위치한 글자 삭제	X	현재 커서가 위치한 앞 글자 삭제
dd	현재 커서의 행 삭제	숫자 dd	현재 커서부터 숫자만큼의 행 삭제
yy	현재 커서가 있는 행을 복사	숫자 yy	현재 커서로부터 숫자만큼의 행을 복사
p	복사한 내용을 현재 행 이후에 붙여넣기	P	복사한 내용을 현재 행 이전에 붙여넣기
u	이전으로 되돌리기 (Undo)	Ctrl + r	되돌린 것 다시 실행하기 (Redo)
>	들여쓰기	<	내어쓰기
.	이전 명령어를 다시 실행	cw	단어 잘라내기

# 리눅스 명령어

# 에디터(Editor)

키	설명	키	설명
명령 모드에서 커서를 이동하기 위한 키			
h	커서를 왼쪽으로 한 칸 이동	j	커서를 아래로 한 칸 이동
k	커서를 위로 한 칸 이동	l	커서를 오른쪽으로 한 칸 이동
ctrl + F Page down	다음 화면으로 이동	Ctrl + B page up	이전 화면으로 이동
^ Home	현재 행의 처음으로 이동	\$ End	현재 행의 마지막으로 이동
gg	제일 첫 행으로 이동	G	제일 끝 행으로 이동
숫자G	해당 숫자의 행으로 이동	:숫자 Enter	해당 숫자의 행으로 이동

version 1.1  
April 1st, 06

## vi / vim 단축키 모음

Esc  
명령 모드

~ 대소문자 전환	! 외부 명령	@ 매크로 실행	# 이전 검색	\$ 줄 끝으로 이동	% 일치하는 괄호 찾기	^ 줄의 첫 글자	& :s 반복	* 다음 검색	( 문장 시작	) 문장 끝	아래줄로 이동	+ 다음 줄
\. 매크로 이동	1	2	3	4	5	6	7	8	9	0 줄의 처음	- 이전 줄	= 자동 들여쓰기 <sup>3</sup>
Q 실행 모드	W 다음 WORD	E 끝 WORD	R 수정 모드	T 뒤로 검색	Y 줄단위 복사	U 줄 단위 실행취소	I 줄 시작에서 삽입	O 행 위에 삽입	P 커서 이전에 붙여넣기	{ 문단 시작	}	문단 끝
q 매크로 기록	w 다음 단어	e 단어 끝	r 한 문자 교체	t 한 문자 검색	y 복사 <sup>1,3</sup>	u 실행취소	i 편집 모드	o 행 아래에 삽입	p 커서 이후에 붙여넣기 <sup>1</sup>	[ 기타	]	기타
A 줄 끝에 덧붙이기	S 줄 삭제후 편집모드	D 줄 끝까지 삭제	F 뒤로 검색	G 파일 끝/ 줄로 이동	H 화면 상단	J 줄 합치기	K 도움말	L 화면 하단	: ex 명령줄	". 레지스터 지정 <sup>1</sup>	열 이동	
a 덧붙이기	s 단어 삭제후 편집모드	d 삭제 <sup>1,3</sup>	f 한 문자 찾기	g 확장 명령 <sup>6</sup>	h ←	j ↓	k ↑	l →	; t/T/f/F 명령 반복	'. 매크로 이동	\. 사용 안함	
Z 종료 <sup>4</sup>	X 백스페이스	C 줄 끝까지 바꾸기	V 줄단위 비주얼모드	B 이전 WORD	N 이전 (찾기)	M 화면 가운데	< <sup>3</sup> 내어쓰기	> <sup>3</sup> 들여쓰기	? 찾기 (뒤로)			
Z 확장 <sup>5</sup>	X 글자 삭제	c 바꾸기 <sup>1,3</sup>	v 비주얼 모드	b 이전 단어	n 다음 (찾기)	m 마크 설정	, 역순 검색	. 명령 반복	/ 찾기			

## 동작

커서를 이동하거나, 연산자가  
동작할 범위를 지정합니다.

## 명령

바로 동작하는 명령,  
빨간색은 편집 모드로 변경됩니다.

## 연산자

이동 관련 문자(숫자나 커서 이동)와  
함께 사용되어야 하며, 커서의 위치부터  
목적지까지 연산합니다.

## 확장

특별한 키 함수로,  
추가적인 키 입력이 필요합니다.

q.

입력후 (숫자를 제외한 .으로 끝낼수  
있는) 글자를 입력하여야 합니다.words: 구분자로 공백, 특수기호 모두 사용  
WORDS: 구분자로 공백 문자만 사용words: quux(foo, bar, baz);  
WORDS: quux(foo, bar, baz);

## 주요 명령행 명령 ('ex'):

:w (저장), :q (종료), :q! (저장하지 않고 종료)  
:e f (파일 f 열기),  
:%s/x/y/g (파일 전체에서 'x' 를 'y' 로 교체),  
:h (vim 도움말), :new (새 파일)

## 그외 중요한 명령들:

CTRL-R: 재실행 (vim),  
CTRL-F/-B: 페이지 위로/아래로,  
CTRL-E/-Y: 줄 스크롤 위로/아래로,  
CTRL-V: 블록-비주얼 모드 (vim 전용)

## 비주얼 모드:

커서를 움직여 지정한 범위에 연산자를  
적용합니다. (vim 전용)

## 참고:

- (1) 복사/붙여넣기/지우기 명령어를 사용하기 전에 "x"를  
입력하여 레지스터(클립보드)를 지정하세요. (x는  
a에서 z 또는 \* 을 사용할 수 있음) (예: "ay\$ 를 입력하면  
현재 커서에서 라인 끝까지의 내용을 레지스터 'a'에  
저장합니다.)
- (2) 어떤 명령을 입력하기 전에 횡수를 지정하면, 횡수만큼  
반복하게 됩니다.(예: 2p, d2w, 5l, d4j)
- (3) 연속으로 입력하는 명령은 현재의 라인에 반영됩니다.  
예시: dd(현재 라인 지우기), >>(들여쓰기)
- (4) ZZ 는 저장후 종료, ZQ는 저장하지 않고 종료.
- (5) zt : 커서가 위치한 곳을 제일위로 올리기,  
zb : 바닥으로, zz : 가운데로
- (6) gg : 파일의 처음으로(Vim 전용),  
gf : 커서가 위치한 곳의 파일 열기(Vim 전용)

### 리눅스의 사용자와 그룹 관리

- 리눅스는 다중 사용자 시스템(multi-user system)으로 여러 명이 동시 접속 가능
- root: Super user로 관리자를 의미. 시스템의 모든 작업 실행 가능

#### /etc/passwd

- 시스템에 존재하는 모든 사용자 정보가 저장되어 있는 파일 (수정은 root만, 읽기는 누구나 가능)

```
root:x:0:0:root:/root:/bin/bash
```

사용자명:암호:사용자ID:사용자가 소속된 그룹ID:추가정보:홈디렉터리:기본셸정보

#### /etc/group

- 시스템에 존재하는 모든 그룹 정보다 저장되어 있는 파일 (수정은 root만, 읽기는 누구나 가능)

```
root:x:0:
```

그룹이름:그룹비밀번호:그룹ID:그룹에속한사용자이름



리눅스의 사용자와 그룹 관리

/etc/shadow

- 리눅스의 계정의 패스워드 정보를 담고 있는 파일 (root 계정 이외의 계정은 읽을 수 없고 접근 불가능)

```
root:$6$IsQWPvLyCh/sUZYU$qGe1aSkK4EzUqdEgd50kC7VMKFeaPa31h0deCauAyQvbGLp63LfayZsJc01nB/-6v23ao4fSE1LZqD97W3P/qB1:19118:0:99999:7:::
```

사용자명:해싱알고리즘:salt값:해싱된비밀번호:마지막변경날짜:최소사용일수:최대사용일수:만료전경고일수:비밀번호만료후유예기간:계정의만료기간

항목	설명
해싱알고리즘	1 : MD5(가장 취약한 일방향 해쉬로 사용하지 않음) / 2 : BlowFish / 5 : SHA-256 / 6 : SHA-512
salt값	각 해쉬에 첨가할 랜덤값으로 비밀번호 생성 시 salt값을 첨가
해싱된비밀번호	해싱알고리즘 + salt 값을 통해 암호화된 비밀번호. *: 패스워드가 잠긴 상태, !: 사용자를 생성하고 패스워드를 설정하지 않은 상태, empty: 패스워드 미설정이지만 로그인 가능
마지막변경날짜	1970년 1월 1일 기준으로 비밀번호를 변경한 후 지난 날짜를 일수로 표시
최소사용일수	패스워드의 최소 사용기간 설정으로 패스워드를 변경한 이후 최소 이 기간은 사용해야한다는 것을 의미
최대사용일수	마지막 패스워드 변경 이후 만료일수
만료전경고일수	패스워드 만료 이전에 경고할 일수
비밀번호만료후유예기간	패스워드가 만료된 이후에 계정이 잠기기 전까지 비활성 일수. 해당 비활성 기간동안에 패스워드를 변경해야 계정이 잠기지 않음
계정의만료기간	계정 만료일. 1970년 1월 1일 기준으로 일수로 표시

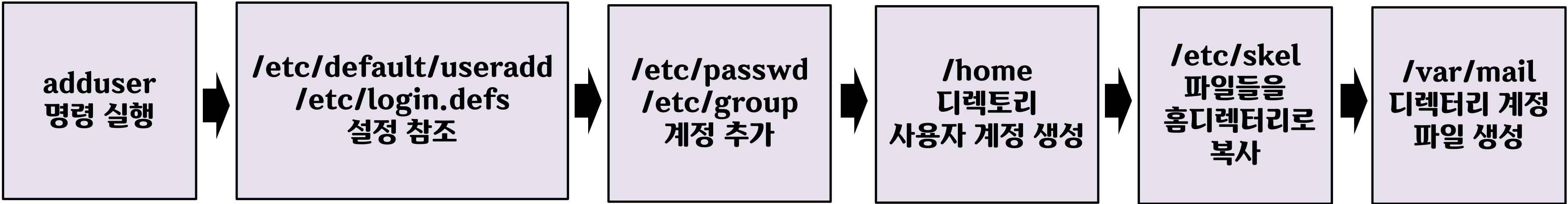
사용자 및 그룹 관련 명령어

adduser (추천)

- 새로운 사용자를 추가함. 추가시 유저 관련 정보 함께 입력.
- /etc/passwd, /etc/shadow, /etc/group 파일에 새로운 행이 추가됨
- /etc/skel의 내용을 토대로 home디렉토리가 생성됨

#adduser [사용자명]

옵션/명령어	의미	사용 예
--uid [uid] [유저명]	사용자를 생성하면서 사용자 ID를 지정	adduser --uid 1111 user1
--gid [gid] [유저명]	사용자를 생성하면서 그룹 ID를 지정	adduser --gid 1000 user1
--home [디렉터리] [유저명]	사용자를 생성하면서 홈 디렉터리를 지정	adduser --home /newhome user1
--shell [셸] [유저명]	사용자를 생성하면서 기본 셸 종류를 지정	adduser --shell /bin/csh user1



사용자 및 그룹 관련 명령어

useradd

- 새로운 사용자를 추가함
  - 홈디렉토리생성x, bash등 환경설정 x, 데몬 서비스 등을 위해 존재함
- #useradd [옵션] [사용자명]

옵션/명령어	의미	사용 예
--uid [uid]	사용자를 생성하면서 사용자 ID를 지정	useradd --uid 1111 user1
--gid [gid]	사용자를 생성하면서 그룹 ID를 지정	useradd --gid 1000 user1
-m	사용자의 홈디렉터리가 없으면 생성함	useradd -m user1
-p	사용자의 비밀번호를 설정. 비밀번호 정책에 부합해야 함	useradd -p 123 user1
-s [셸]	사용자의 기본 셸을 설정	useradd -s /bin/bash
-d [디렉터리]	사용자의 홈 디렉터리를 지정	useradd -d /home/test user1
-G [그룹명]	사용자의 그룹을 지정	useradd -G newGroup user1
-e [YYYY-MM-DD]	사용자 계정의 만료일을 지정	useradd -e 2030-05-20
-f [일수]	사용자 계정의 비밀번호 만료일을 지정 (해당 일수 이후에 만료)	useradd -f 90



사용자 및 그룹 관련 명령어

usermod

- 사용자의 속성을 변경
- #usermod [옵션] [사용자명]

옵션/명령어	의미	사용 예
--shell	사용자의 기본 셸을 변경	usermod --shell /bin/csh user1
--groups	사용자의 보조 그룹 설정	usermod --groups ubuntu user1
-l [새계정명]	사용자의 계정명을 변경	usermod -l newUser oriUser
-f [일수]	사용자 계정의 비밀번호 만료일을 지정 (해당 일수 이후에 만료)	usermod -f 90 user1
--uid [uid]	사용자 ID를 지정	usermod --uid 1111 user1
--gid [gid]	그룹 ID를 지정	usermod --gid 1000 user1
-e [YYYY-MM-DD]	사용자 계정의 만료일을 지정	usermod -e 2030-05-20
-d [디렉터리]	사용자의 홈 디렉터리를 지정	usermod -d /home/test user1

사용자 및 그룹 관련 명령어

deluser

- 사용자를 삭제 (우분투 명령어)
- #deluser [사용자명]

옵션/명령어	의미	사용 예
--remove-home	사용자의 홈디렉토리를 제거	deluser --remove-home user1
--remove-all-files	사용자의 모든 파일을 제거	deluser --remove-all-files user1

userdel

- 사용자를 삭제 (리눅스 명령어)
- #userdel [사용자명]
- #userdel -r [사용자명](=> 사용자를 삭제하면서 홈 디렉터리까지 삭제)

옵션/명령어	의미	사용 예
-r	사용자를 삭제하면서 홈 디렉터리까지 삭제	userdel -r user1

groups

- 사용자가 소속된 그룹을 출력
- #groups
- #groups [사용자명]

사용자 및 그룹 관련 명령어

addgroup / groupadd

- 새로운 그룹 생성
- #groupadd [옵션] [그룹명]

옵션/명령어	의미	사용 예
--gid [gid]	그룹을 생성하며 그룹 ID를 지정	groupadd --gid 2222 group1

groupmod

- 그룹의 속성을 변경
- #groupmod [옵션] [그룹명]

옵션/명령어	의미	사용 예
-g [gid]	지정한 gid로 그룹의 ID를 변경	groupmod -g 1111 group1
-n [새이름] [기존이름]	지정한 name으로 그룹명을 변경한다.	groupmod -n group2 group1

delgroup / groupdel

- 그룹을 삭제 (해당 그룹을 주 그룹으로 지정한 사용자가 없어야 함)
- #groupdel [그룹명]

사용자 및 그룹 관련 명령어

gpsswd

- 그룹의 암호를 설정하거나 그룹 관리를 수행

#gpsswd [옵션] [사용자명] [그룹명]

옵션/명령어	의미	사용 예
gpsswd [그룹명]	그룹의 암호를 설정	gpsswd group1
-A [사용자명] [그룹명]	사용자를 그룹의 관리자로 지정 (Admin)	gpsswd -A user1 group1
-a [사용자명] [그룹명]	사용자를 그룹의 사용자로 추가 (add)	gpsswd -a user1 group1
-d [사용자명] [그룹명]	사용자를 그룹의 사용자에서 제거 (delete)	gpsswd -d user1 group1
-r [그룹명]	그룹의 암호를 제거	gpsswd -r group1
-R [그룹명]	그룹에 들어오는 것을 제한함 (패스워드 입력 필수)	gpsswd -R group1

passwd

- 사용자의 비밀번호를 변경함

#passwd [사용자명]

옵션/명령어	의미	사용 예
-l [사용자명]	사용자에 락을 걸어서 로그인을 못하게 만듦	passwd -l user
-u [사용자명]	사용자의 락을 해제시킴	passwd -u user

사용자 및 그룹 관련 명령어

chage

- 사용자의 암호 변경 주기를 설정
- #change [옵션] [일자] [유저명]

옵션/명령어	의미	사용 예
-l [유저명]	사용자에 설정된 사항 확인	chage -l user1
-m [일자] [유저명]	사용자에 설정한 암호를 사용해야 하는 최소 일자	chage -m 2 user1 (변경 후 최소 2일은 사용해야 함)
-M [일자] [유저명]	사용자에 설정한 암호를 사용해야 하는 최대 일자	chage -M 10 user1 (변경 후 최대 10일 사용가능)
-E [날짜] [유저명]	사용자에 설정한 암호가 만료되는 날짜	chage -E 2025/10/10 user1 (2025/10/10까지만 사용가능)
-W [일자] [유저명]	사용자에 설정한 암호가 만료되기 전에 경고하는 기간	chage -W 10 user1 (암호가 만료되기 10일전부터 경고메세지가 나옴. 기본값은 7)

root계정으로 진행합니다.

1. korea 사용자 생성 (홈 디렉터리 자동 생성 및 이름 등의 정보가 있는 사용자로)
2. korea 사용자 비밀번호를 'test1'로 변경
3. 계정명을 japan으로 변경. 계정이 변경되었으니 korea 그룹명도 japan으로 변경
4. 그룹 KoreaIT생성
5. 그룹 KoreaIT에 japan 사용자를 사용자로 그룹에 포함시킴
6. japan 사용자의 비밀번호 정책을 다음과 같이 변경하고 변경된 내역 출력
  - 최소 사용일자 : 30일
  - 최대 사용일자 : 90일
  - 만료 전 경고일자 : 10일
7. 생성된 사용자(japan)와 그룹(KoreaIT) 삭제

① !al                      ② !?al?

③ !!al                  ④ history al

① vi                      ② pico  
③ gedit                ④ emacs

① :                      ② ;  
③ !                      ④ /

①	⌈ : vi,	⌊ : vim,	⊞ : gVim
②	⌈ : emacs,	⌊ : GNU emacs,	⊞ : Xemacs
③	⌈ : pico,	⌊ : nano,	⊞ : emacs
④	⌈ : vi,	⌊ : emacs,	⊞ : pico

① :w                  ② :w!  
③ :q!                ④ :wq

6. 다음 명령에 대한 설명으로 알맞은 것은?

- ① 최근에 실행한 마지막 5개의 명령어 목록을 출력한다.
- ② 히스토리 명령 목록의 번호 중에서 5번에 해당하는 명령을 실행한다.
- ③ 히스토리 명령 목록에서 5만큼 거슬러 올라가서 해당 명령을 실행한다.
- ④ 히스토리 명령 목록에서 번호가 1번부터 5번에 해당하는 명령을 출력한다.

```
[ihduser@www ~]$ history 5
```

7. 다음 중 바로 직전에 내린 명령을 재실행할 때 사용하는 명령으로 알맞은 것은?

- ① !1                      ② !0
- ③ !!                      ④ history -1

8. 다음 중 vi 편집기 실행 후 명령모드에서 입력 모드로 전환하는 키로 틀린 것은?

- ① a                      ② e
- ③ i                      ④ o

9. 다음 중 vi 편집기에서 변경된 내용을 저장하지 않고 강제로 종료할 때 사용하는 명령으로 알맞은 것은?

- ① :w!                      ② :e!
- ③ :q!                      ④ :x!

10. vi 편집기 사용 중 비정상적인 종료로 인해 작업이 중단되었다. 다음 중 생성된 스왑 파일 목록을 확인하는 방법으로 알맞은 것은?

- ① vi +                      ② vi -r
- ③ vi -R                      ④ vi -s



11. 특정 사용자의 로그인 시에 부여되는 셸 정보를 확인하려고 할 때 파일명으로 알맞은 것은?

- ① /etc/shells                      ② /etc/passwd
- ③ /etc/shadow                    ④ /etc/login.defs

12. 원격지에서 vi편집기를 이용하여 lin.txt 파일을 편집 중에 네트워크 단절로 중단되었다. 작업중이던 파일 내용을 불러오려고 할 때 ( 괄호 ) 안에 들어갈 내용으로 알맞은 것은?

- ① +                                  ② -s
- ③ -r                                ④ -R

```
$ vi ( 괄호 ) lin.txt
```

13. 다음 명령에 대한 설명으로 알맞은 것은?

- ① 최근에 실행한 마지막 5개의 명령어 목록을 출력한다.
- ② 히스토리 명령 목록의 번호 중에서 5번에 해당하는 명령을 실행한다.
- ③ 히스토리 명령 목록에서 5만큼 거슬러 올라가서 해당 명령을 실행한다.
- ④ 히스토리 명령 목록에서 번호가 1번부터 5번에 해당하는 명령을 출력한다.

```
[ihduser@www ~]$ !5
```

수고하셨습니다