

## Chapter 09 사용자 관리

# 목차

01 사용자 계정 관련 파일

02 사용자 계정 관리 명령

03 그룹 관리 명령

04 사용자 정보 관리 명령

# 학습목표



- 사용자 계정과 관련된 파일의 구조를 이해하고 설명할 수 있다.
- 사용자 계정과 그룹을 생성·수정·삭제할 수 있다.
- UID와 EUID의 차이를 이해하고 설명할 수 있다.
- 사용자 확인 명령으로 사용자 정보를 관리하고 확인할 수 있다.
- 파일 및 디렉터리의 소유자와 소유 그룹을 변경할 수 있다.

# 00 Preview

# 00 Preview

## ■ 9장의 내용 구성

- 다른 계정을 생성하고, 계정의 정보를 수정하며, 더 이상 사용하지 않는 계정을 삭제하는 방법에 대해 배움
- 리눅스에서 사용자는 하나 이상의 그룹에 소속되어 있으므로 그룹을 생성하고 수정하고 삭제하는 방법과 사용자의 소속 그룹을 추가하는 방법을 살펴봄
- 암호의 사용 가능 기간을 설정하는 패스워드 에이징 방법도 알아봄
- 사용자의 정보를 확인하고 소속 그룹을 확인하는 명령을 살펴봄

사용자 계정 관련 파일	/etc/ passwd	/etc/ shadow	/etc/ login.defs	/etc/ group	/etc/ gshadow
사용자 계정 관리 명령	사용자 계정 생성	사용자 계정 정보 수정	패스워드 에이징	사용자 계정 삭제	
그룹 관리 명령	그룹 생성	그룹 정보 수정	그룹 삭제	그룹 암호 설정과 사용	
사용자 정보 관리 명령	UID와 EUID	사용자 확인 명령	root 권한 사용 방법	passwd 명령 사용법	파일 소유자/ 소유 그룹 변경

# 01 사용자 계정 관련 파일

# 01 사용자 계정 관련 파일

## ■ /etc/passwd 파일

- 사용자 계정 정보가 저장된 기본 파일
- 초기의 유닉스에서는 암호도 /etc/passwd 파일에 저장했으나 해킹 위험이 증가하면서 암호는 /etc/shadow 파일에 별도로 저장
- root 계정으로 수정할 수 있지만 사용자 계정이 저장되는 중요한 파일이므로 함부로 수정하지 말고, 사용자 계정의 추가·삭제·수정 등 변경 사항이 있을 때는 관련된 명령으로 작업하는 것이 바람직함

# 01 사용자 계정 관련 파일

## ■ /etc/passwd 파일의 구조

로그인 ID	:	x	:	UID	:	GID	:	설명	:	홈 디렉터리	:	로그인 셸
①		②		③		④		⑤		⑥		⑦

그림 9-1 /etc/passwd 파일의 사용자 계정 정보 구성

- ① 로그인 ID: 사용자 계정의 이름을 뜻하며 사용자 ID, 사용자 이름, 로그인 이름이라고도 함.  
로그인 ID는 32자를 넘을 수 없고 소문자, 대문자, 숫자, 언더스코어( \_ ), 하이픈(-)을 사용할 수 있음.  
다만, 하이픈으로 시작하거나 로그인 ID 전체가 숫자이면 안 되고, 중복되는 이름을 사용하면 안 됨
- ② x: 초기 유닉스 시스템에서 사용자 암호를 저장하던 항목.  
요즘은 보안상의 이유로 사용자 암호를 /etc/shadow 파일에 별도로 보관.  
이 항목은 역사적인 이유로 이전 프로그램과의 호환성을 위해 그대로 유지하고 있으며 x로 표시함



# 01 사용자 계정 관련 파일

## ■ /etc/passwd 파일의 구조

로그인 ID : x : UID : GID : 설명 : 홈 디렉터리 : 로그인 셸

①	②	③	④	⑤	⑥	⑦
---	---	---	---	---	---	---

그림 9-1 /etc/passwd 파일의 사용자 계정 정보 구성

- ③ UID: 사용자 ID 번호로 시스템이 사용자를 구별하기 위해 사용하는 번호.  
일반적으로 0~999번과 65534번은 시스템 사용자를 위한 UID로 예약되어 있음.  
시스템 사용자 계정은 시스템이 관리 업무를 위해 내부적으로 사용하려고 예약되어 있는  
계정이므로 임의로 수정하지 않음. 일반 사용자는 UID 1000번부터 할당됨

# 01 사용자 계정 관련 파일

## ■ /etc/passwd 파일의 구조

로그인 ID : x : UID : GID : 설명 : 홈 디렉터리 : 로그인 셸

①	②	③	④	⑤	⑥	⑦
---	---	---	---	---	---	---

그림 9-1 /etc/passwd 파일의 사용자 계정 정보 구성

- 기본적으로 등록되는 시스템 사용자의 ID 살펴보기
  - 0(root): root 사용자 계정
  - 1(bin): 명령어 관리를 위한 계정
  - 2(daemon): 시스템 데몬 계정
  - 3(adm): 시스템 로그와 관련된 계정
  - 4(lp): 프린트 서비스와 관련된 계정

# 01 사용자 계정 관련 파일

## ■ /etc/passwd 파일의 구조

로그인 ID : x : UID : GID : 설명 : 홈 디렉터리 : 로그인 셸

①	②	③	④	⑤	⑥	⑦
---	---	---	---	---	---	---

그림 9-1 /etc/passwd 파일의 사용자 계정 정보 구성

- 로그인 ID가 다르더라도 UID가 같으면 리눅스 시스템은 같은 사용자로 판단
- 사용자 이름이 root가 아닌 다른 이름, 예를 들어 user1이더라도 user1의 UID가 0번이면 시스템은 root 계정으로 인식한다는 뜻
- 따라서 시스템 관리자는 시스템 계정의 UID 가 중복되지 않도록 주의해야 함

# 01 사용자 계정 관련 파일

## ■ /etc/passwd 파일의 구조

로그인 ID : x : UID : GID : 설명 : 홈 디렉터리 : 로그인 셸

①      ②      ③      ④      ⑤      ⑥      ⑦

그림 9-1 /etc/passwd 파일의 사용자 계정 정보 구성

- ④ GID: 그룹 ID를 나타냄. 리눅스에서 사용자는 무조건 한 개 이상의 그룹에 소속됨.  
사용자의 기본 그룹은 사용자를 등록할 때 정해지며, 특별히 소속 그룹을 지정하지 않으면 자동으로 로그인 ID가 그룹으로 등록됨  
시스템에 등록된 그룹에 대한 정보는 /etc/ group 파일에 저장되어 있음
- ⑤ 설명: 사용자의 실명이나 부서명, 연락처 등 사용자에게 대한 일반적인 정보가 기록되는 부분

# 01 사용자 계정 관련 파일

## ■ /etc/passwd 파일의 구조

로그인 ID : x : UID : GID : 설명 : 홈 디렉터리 : 로그인 셸

①	②	③	④	⑤	⑥	⑦
---	---	---	---	---	---	---

그림 9-1 /etc/passwd 파일의 사용자 계정 정보 구성

- ⑥ 홈 디렉터리: 사용자 계정에 할당된 홈 디렉터리의 절대 경로를 기록.  
사용자가 로그인할 때 자동으로 로그인되는 디렉터리가 홈 디렉터리.  
홈 디렉터리 아래에 다른 디렉터리나 파일을 생성할 수 있음
- ⑦ 로그인 셸: 사용자의 로그인 셸을 지정함.  
로그인 셸은 사용자가 로그인할 때 기본적으로 동작하는 셸.  
현재 로키 리눅스에서는 배시셸(/bin/bash)을 기본 셸로 사용하고 있음

# 01 사용자 계정 관련 파일

## ■ /etc/passwd 파일의 예

- /etc/passwd 파일의 예. /etc/passwd 파일은 누구나 읽을 수 있도록 접근 권한이 설정되어 있음

```
[user1@localhost ~]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
(생략)
user1:x:1000:1000:user1:/home/user1:/bin/bash
```

- adm 계정을 살펴보면 UID가 3번, GID가 4번으로 설정되어 있음. /etc/group 파일을 보면 GID 4는 adm 그룹임
- 홈 디렉터리는 /var/adm으로 되어 있고, 셸이 /sbin/nologin으로 설정되어 있음. root 계정을 제외하고 시스템 계정에는 /bin/bash 같은 정상적인 셸이 설정되지 않도록 해야 함
- 일반 사용자 계정인 user1의 UID가 1000번으로 정의된 것도 확인할 수 있음

# 01 사용자 계정 관련 파일

## ■ /etc/shadow 파일

- /etc/shadow 파일은 보안 문제 때문에 사용자 암호에 관한 정보를 별도로 관리하는 파일
- /etc/passwd 파일은 누구나 읽을 수 있지만 /etc/shadow 파일에는 읽기 권한이 없음
- 그러나 root 계정으로 들어가면 /etc/shadow 파일의 내용을 볼 수 있음

```
[user1@localhost ~]$ ls -l /etc/passwd /etc/shadow
-rw-r--r--. 1 root root 2221  8월 20 16:43 /etc/passwd
-----. 1 root root 1191  8월 31 21:52 /etc/shadow
```

# 01 사용자 계정 관련 파일

## ■ /etc/passwd 파일의 구조

로그인 ID:암호(패스워드):최종 변경일:MIN:MAX:WARNING:INACTIVE:EXPIRE:Flag

①	②	③	④	⑤	⑥	⑦	⑧	⑨
---	---	---	---	---	---	---	---	---

그림 9-2 /etc/shadow 파일의 정보 구성

① 로그인 ID: 사용자 계정의 이름으로 /etc/passwd 파일의 로그인 ID와 같음

② 암호(패스워드): 실제 비밀번호가 암호화되어 저장.

비밀번호는 일방향 암호여서 원래 어떤 문자인지 복호화할 수 없음.

사용자가 로그인할 때 입력하는 비밀번호를 다시 암호화 하여 이 암호와 비교하는 것.

이 암호 항목에 아무 값도 없으면 암호가 지정되지 않은 계정임을 뜻함.

시스템 계정에는 \* 또는 !!가 표시되어 있음



# 01 사용자 계정 관련 파일

## ■ /etc/passwd 파일의 구조

로그인 ID: 암호(패스워드): 최종 변경일: MIN: MAX: WARNING: INACTIVE: EXPIRE: Flag

①      ②      ③      ④      ⑤      ⑥      ⑦      ⑧      ⑨

그림 9-2 /etc/shadow 파일의 정보 구성

- ③ 최종 변경일: 암호가 마지막으로 변경된 날짜를 지정함.

이 날짜는 일반적인 연월일 형태로 기록되는 것이 아니라, 유닉스의 전통을 따라 1970년 1월 1일을 기준으로 날수를 기록

- date 명령을 사용해 1970년 1월 1일을 기준으로 날짜를 알아보는 방법의 예

```
[user1@localhost ~]$ date -d "1970-01-01 19608 days"
2023. 09. 08. (금) 00:00:00 KST
```

# 01 사용자 계정 관련 파일

## ■ /etc/passwd 파일의 구조

로그인 ID:암호(패스워드):최종 변경일:MIN:MAX:WARNING:INACTIVE:EXPIRE:Flag

①	②	③	④	⑤	⑥	⑦	⑧	⑨
---	---	---	---	---	---	---	---	---

그림 9-2 /etc/shadow 파일의 정보 구성

- ④ MIN: MIN을 비롯해 MAX, WARNING, INACTIVE, EXPIRE 항목을 패스워드 에이징이라고 함.  
각각 패스워드와 관련된 사용 기간을 지정.  
MIN은 암호를 변경한 후 사용해야 하는 최소 기간
- ⑤ MAX: 암호를 사용할 수 있는 최대 기간. 이 기간이 지나면 새로운 암호를 입력하게 함
- ⑥ WARNING: 암호가 만료되기 전에 경고를 시작하는 날수를 나타냄
- ⑦ INACTIVE: 암호가 만료된 후에도 이 항목에 지정한 날수 동안은 로그인 가능하도록 함.  
이 기간이 지나면 계정이 잠겨버리는데, 이 경우 관리자에게 문의해야만 해결할 수 있음

# 01 사용자 계정 관련 파일

## ■ /etc/passwd 파일의 구조

로그인 ID: 암호(패스워드): 최종 변경일: MIN: MAX: WARNING: INACTIVE: EXPIRE: Flag

①	②	③	④	⑤	⑥	⑦	⑧	⑨
---	---	---	---	---	---	---	---	---

그림 9-2 /etc/shadow 파일의 정보 구성

- ⑧ EXPIRE: 사용자 계정이 만료되는 날로, 이 날짜가 지나면 해당 계정으로 로그인할 수 없음.  
이 항목은 최종 변경일처럼 1970년 1월 1일을 기준으로 한 날수로 표시됨
- ⑨ Flag: 향후 사용할 목적으로 비워둔 항목

# 01 사용자 계정 관련 파일

## ■ /etc/shadow 파일의 예

- /etc/shadow 파일의 예. 이 파일은 root 계정으로 변경한 뒤 확인할 수 있음

```
[root@localhost ~]# more /etc/shadow
root:$6$dAbZm45DxkEYMU/a$vmqtGqb5k.oJHw13Xj3DI1/aBa70hJE.wgw7qgg4nggh0ucxy8x08wDyIoSt
Hp15kmhbRJ0arAMvnQh288eHZ:19600:0:99999:7:::
bin:!:19469:0:99999:7:::
daemon:!:19469:0:99999:7:::
adm:!:19469:0:99999:7:::
(생략)
dnsmasq:!!:19547:::::::
tcpdump:!!:19547:::::::
user1:$6$NrD5y0Gfn2XIjPTX$ZTpYYZEVwjGHGj60deIjCZAjgHDonYpnmW9hNTJ2ACd2tvaqYJw8FmvYTM
9pZqss/Ni4b7vWZPEWfUXYVYpi6:19608:0:99999:7:::
```

- root 계정과 user1 계정만 암호가 설정되어 있고 나머지 시스템 계정은 암호가 \* 또는 !!로 설정되어 있음을 알 수 있음

# 01 사용자 계정 관련 파일

## ■ /etc/login.defs 파일

- /etc/login.defs 파일은 사용자 계정의 설정과 관련된 기본값을 정의한 파일.  
이 파일은 주석이 잘 되어 있어서 주석만 보아도 각 항목의 역할을 이해할 수 있음

```
[user1@localhost ~]$ cat /etc/login.defs
(생략)
# Password aging controls:
#
#      PASS_MAX_DAYS   Maximum number of days a password may be used.
#      PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#      PASS_MIN_LEN     Minimum acceptable password length.
#      PASS_WARN_AGE    Number of days warning given before a password expires.
#
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_WARN_AGE   7
(생략)
```

# 01 사용자 계정 관련 파일

## ■ /etc/group 파일

- 그룹의 정보가 저장된 파일
- 리눅스에서 사용자는 기본적으로 하나 이상의 그룹에 속해 있음. 사용자가 속한 그룹 중 /etc/passwd 파일의 GID 항목에 지정된 그룹이 기본 그룹이며, 사용자가 속한 2차 그룹은 /etc/group 파일에 지정함

# 01 사용자 계정 관련 파일

## ■ /etc/group 파일의 구조

그룹 이름 : x : GID : 그룹 멤버

①      ②      ③      ④

그림 9-3 /etc/group 파일의 정보 구성

- ① 그룹 이름: 그룹의 이름
- ② x: 그룹의 암호를 저장하는 곳으로, 예전 유닉스에서는 비어 있거나 \* 표시를 했음.  
리눅스에서는 여기에 암호화된 그룹 암호를 저장하거나 /etc/gshadow 파일에 그룹 암호를 저장함.  
그룹 암호는 newgrp 명령으로 자신이 속하지 않은 그룹으로 전환할 때 필요함
- ③ GID: 그룹을 식별하는 번호.  
login.defs 파일에 정의한 대로 1000~60000은 사용자 정의 그룹의 GID로 사용함
- ④ 그룹 멤버: 그룹에 속한 멤버들의 사용자 계정 이름으로 쉼표(,)를 넣어 사용자를 구분함.  
사용자의 2차 그룹을 나타냄

# 01 사용자 계정 관련 파일

## ■ /etc/group 파일의 예

- /etc/group 파일의 예로 user1 그룹도 생성되어 있음

리눅스의 경우, 사용자 계정을 생성할 때 특별히 지정하지 않으면 사용자 이름과 같은 이름으로 그룹을 생성함

```
[user1@localhost ~]$ cat /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:
(생략)
user1:x:1000:
```



# 01 사용자 계정 관련 파일

## ■ /etc/gshadow 파일

- 그룹 암호가 저장된 파일
- 유닉스에서는 그룹에 암호가 필요할 경우 /etc/group 파일의 그룹 암호 자리에 암호화된 그룹 암호를 설정했는데, 리눅스에서는 /etc/shadow 파일과 비슷하게 /etc/gshadow 파일을 만든 것

# 01 사용자 계정 관련 파일

## ■ /etc/gshadow 파일의 구조

그룹 이름 : 그룹 암호 : 관리자 : 그룹 멤버  
①                      ②                      ③                      ④

그림 9-4 /etc/gshadow 파일의 정보 구성

- ① 그룹 이름: 그룹의 이름
- ② 그룹 암호: 암호화된 그룹 암호
- ③ 관리자: 그룹의 암호나 멤버를 바꿀 수 있는 사용자 계정으로, 여러 개일 경우 쉼표로 구분
- ④ 그룹 멤버: 그룹에 속한 멤버들의 사용자 계정 이름이며, 쉼표로 사용자를 구분

# 01 사용자 계정 관련 파일

## ■ /etc/gshadow 파일의 예

- /etc/gshadow 파일의 예로, 아직 암호가 설정된 그룹이 없음

```
[root@localhost ~]# cat /etc/gshadow
root:::
bin:::
daemon:::
sys:::
adm:::
(생략)
user1:!::
```

## 02 사용자 계정 관리 명령

## 02 사용자 계정 관리 명령

### ■ 사용자 계정 생성

- 사용자 계정을 생성할 때는 useradd 명령을 사용
- 우분투와 달리 로키 리눅스의 adduser 명령은 useradd에 대한 심볼릭 링크다

```
[root@localhost ~]# ls -l /usr/sbin/adduser  
lrwxrwxrwx. 1 root root 7  4월 16 14:17 /usr/sbin/adduser -> useradd*
```

- useradd 명령을 사용할 때 지정하는 옵션은 /etc/passwd 파일에 저장되는 각 항목에 해당함

## 02 사용자 계정 관리 명령

### ■ 사용자 계정 생성

#### useradd

- 기능 사용자 계정을 생성한다.
- 형식 `useradd [옵션] [로그인 ID]`
- 옵션
  - u uid: UID를 지정한다.
  - o: UID의 중복을 허용한다.
  - g gid: 기본 그룹의 GID를 지정한다.
  - G gid: 2차 그룹의 GID를 지정한다.
  - d 디렉터리명: 홈 디렉터리를 지정한다.
  - s 셸: 기본 셸을 지정한다.
  - c 설명: 사용자의 이름 등 부가적인 설명을 지정한다.
  - D: 기본값을 설정하거나 출력한다.
  - e 유효 기간: EXPIRE 항목을 설정한다(YYYY-MM-DD).
  - f 비활성 일수: INACTIVE 항목을 설정한다.
  - k 디렉터리: 계정 생성 시 복사할 초기 파일이나 디렉터리를 설정해 놓은 디렉터리를 지정한다.
- 사용 예
  - `useradd user2`
  - `useradd -m -d /home/user2 user2`
  - `useradd -m -d /home/user2 -u 2000 -g 100 -s /bin/ksh user2`
  - `useradd -D -d /export/home`

## 02 사용자 계정 관리 명령

### ■ 옵션 없이 계정 생성하기

- user1 외에 새로운 계정을 만들어보기
- 옵션 없이 useradd 명령으로 user2 계정을 만들고 홈 디렉터리와 /etc/passwd 파일을 확인해 보면, 홈 디렉터리가 생성되고 /etc/passwd 파일에도 user2 계정이 추가되었음을 알 수 있음

```
[root@localhost ~]# useradd user2
[root@localhost ~]# ls /home
user1 user2
[root@localhost ~]# tail -2 /etc/passwd
user1:x:1000:1000:user1:/home/user1:/bin/bash
user2:x:1001:1001::/home/user2:/bin/bash
```

## 02 사용자 계정 관리 명령

### ■ 옵션 없이 계정 생성하기

- 터미널을 하나 더 띄우고 su 명령을 사용하여 user1 사용자에서 user2 사용자로 전환해 보기

```
[user1@localhost ~]$ su user2  
암호:
```

- 기존 터미널의 root 계정으로 돌아가서 /etc/shadow 파일을 확인해 보면 user2 계정은 암호가 설정되지 않고 !!로 잠겨 있음

```
[root@localhost ~]# grep user2 /etc/shadow  
user2:!!:19609:0:99999:7:::
```



## 02 사용자 계정 관리 명령

### ■ 옵션 없이 계정 생성하기

- 시스템 관리자는 사용자 계정을 추가할 때 반드시 초기 암호를 바로 설정해야 함.  
그리고 사용자에게 초기 암호를 알려주고 사용자가 다시 바꾸도록 권고해야 함
- user2 계정의 암호를 설정해 보자

```
[root@localhost ~]# passwd user2
user2 사용자의 비밀번호 변경 중
새 암호:
새 암호 재입력:
passwd: 모든 인증 토큰이 성공적으로 업데이트되었습니다.
```

- 새로 열었던 터미널에서 user2 계정으로 전환해 보자

```
[user1@localhost ~]$ su user2
암호:
[user2@localhost user1]$
```

## 02 사용자 계정 관리 명령

### ■ 기본 설정값 확인하기: -D 옵션

- 사용자 계정을 설정할 때 참고하는 기본 설정값은 -D 옵션으로 확인할 수 있음

```
[root@localhost ~]# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

## 02 사용자 계정 관리 명령

### ■ 기본 설정값 확인하기: -D 옵션

- -D 옵션으로 출력하는 기본값의 의미는 다음과 같음
  - GROUP: 기본 등록 그룹의 GID로, 100은 users 그룹이다.
  - HOME: 홈 디렉터리의 생성 위치다.
  - INACTIVE: -1이면 INACTIVE 기능이 비활성화된다. 0이면 암호가 만료되자마자 바로 계정이 잠긴다.
  - EXPIRE: 계정 종료일을 지정한다.
  - SHELL: 기본 로그인 셸을 지정한다.
  - SKEL: 홈 디렉터리에 복사할 기본 환경 파일의 위치다.
  - CREATE\_MAIL\_SPOOL: 메일 디렉터리의 생성 여부를 지정한다.

## 02 사용자 계정 관리 명령

### ■ 기본 설정값 확인하기: -D 옵션

- 이 항목들은 /etc/default/useradd 파일에 저장되어 있음

```
[root@localhost ~]# cat /etc/default/useradd
# useradd defaults file
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

## 02 사용자 계정 관리 명령

### ■ 기본 설정값 확인하기: -D 옵션

- vi로 이 파일을 수정할 수 있지만 useradd -D 명령으로 수정하는 것이 바람직함
- -D 다음에 -bHOME, -eEXPIRE, -fINACTIVE, -gGID, -sSHELL를 지정하여 수정할 수 있음

```
[root@localhost ~]# useradd -D -e 0
[root@localhost ~]# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=0
(생략)
```

## 02 사용자 계정 관리 명령

### ■ /etc/skel 디렉터리의 역할

- 사용자 계정을 생성할 때 시스템 운영 정책에 따라 사용자 계정의 홈 디렉터리에 공통으로 배포해야 할 파일이 있을 수 있음
- 예를 들어 .bash\_profile 같은 초기화 파일을 공통으로 작성해 놓고 사용자 계정에 복사하는 것. 이런 경우 /etc/skel 디렉터리에 파일을 만들어놓으면 사용자 계정을 생성할 때 자동으로 복사됨
- /etc/skel 디렉터리의 내용을 확인해 보면 다음과 같음. 아래 파일들이 복사되고 있는 것

```
[root@localhost ~]# ls -a /etc/skel  
.  ..  .bash_logout  .bash_profile  .bashrc  .mozilla
```

## 02 사용자 계정 관리 명령

### ■ 옵션을 지정하여 사용자 계정 생성하기

- 다양한 옵션을 지정하여 사용자 계정을 생성해 보자

```
[root@localhost ~]# useradd -s /bin/sh -d /home/user3 -u 2000 -g 1000 -G 3 user3  
[root@localhost ~]# useradd -s /bin/bash -u 1500 user4
```

- user3은 기본 로그인 셸을 본셸로 지정하고 UID는 2000, GID는 1000, 2차 그룹은 3으로 지정함  
user4는 기본 로그인 셸을 배시셸로 지정하고 UID를 1500으로 지정함
- 명령의 실행 결과

```
[root@localhost ~]# tail -2 /etc/passwd  
user3:x:2000:1000::/home/user3:/bin/sh  
user4:x:1500:1500::/home/user4:/bin/bash  
[root@localhost ~]# grep user3 /etc/group  
sys:x:3:user3
```

## 02 사용자 계정 관리 명령

### ■ 옵션을 지정하여 사용자 계정 생성하기

- 다른 옵션을 지정하여 사용자 계정을 생성해 보자.
- -e와 -f 옵션을 사용하여 EXPIRE와 INACTIVE 값을 지정하고, -c 옵션으로 설명 부분도 값을 지정함

```
[root@localhost ~]# useradd -e 2023-12-31 -f 5 -c "user5 test" user5
[root@localhost ~]# grep user5 /etc/passwd
user5:x:2001:2001:user5 test:/home/user5:/bin/bash
[root@localhost ~]# grep user5 /etc/shadow
user5:!!:19609:0:99999:7:5:19722:
```

- 생성된 계정의 정보를 확인해 보면 UID가 2001번
- useradd는 기존 UID 중 가장 큰 값 에 1을 더하여 UID를 지정함
- /etc/shadow 파일에 EXPIRE와 INACTIVE 값이 설정된 것을 볼 수 있음



## 02 사용자 계정 관리 명령

### ■ 사용자 계정 정보 수정

#### usermod

- 기능 사용자 계정 정보를 수정한다.
- 형식 `usermod [옵션] [로그인 ID]`
- 옵션
  - u uid: UID를 수정한다.
  - o: UID의 중복을 허용한다.
  - g gid: 기본 그룹을 수정한다.
  - G gid: 2차 그룹을 수정한다.
  - d 디렉터리명: 홈 디렉터리를 수정한다.
  - s 셸: 기본 셸을 수정한다.
  - c 설명: 추가적인 설명을 수정한다.
  - f inactive: 계정 비활성화(INACTIVE) 날짜를 수정한다.
  - e expire: 계정 만료(EXPIRE) 날짜를 수정한다.
  - l: 계정 이름을 바꾼다.
- 사용 예
  - `usermod -u 1111 user1`
  - `usermod -d /home/user111 user1`
  - `usermod -l user111 user1`

## 02 사용자 계정 관리 명령

### ■ UID 변경하기: -u 옵션

- 사용자 계정의 UID는 -u 옵션을 사용하여 바꿀 수 있음
- user3 계정의 UID 를 2000번에서 1002번으로 바꿈

```
[root@localhost ~]# usermod -u 1002 user3
[root@localhost ~]# grep user3 /etc/passwd
user3:x:1002:1000::/home/user3:/bin/sh
```

- 사용자 계정의 UID는 -o 옵션을 사용하면 중복할 수 있음
- 리눅스는 UID가 같으면 로그인 ID가 달라도 같은 사용자로 인식하므로 보안에 문제가 생길 수 있음

```
[root@localhost ~]# usermod -u 1001 -o user3
[root@localhost ~]# tail -5 /etc/passwd
user1:x:1000:1000:user1:/home/user1:/bin/bash
user2:x:1001:1001::/home/user2:/bin/bash
user3:x:1001:1000::/home/user3:/bin/sh
user4:x:1500:1500::/home/user4:/bin/bash
user5:x:2001:2001:user5 test:/home/user5:/bin/bash
```

## 02 사용자 계정 관리 명령

### ■ 홈 디렉터리 변경하기: -d 옵션

- 사용자 계정의 홈 디렉터를 변경하려면 -d 옵션을 사용함
- user4 계정의 홈 디렉터를 /home/user4에서 /home/user41로 바꿈

```
[root@localhost ~]# usermod -d /home/user41 user4
[root@localhost ~]# grep user4 /etc/passwd
user4:x:1500:1500::/home/user41:/bin/bash
```

- user4의 홈 디렉터리가 /home/user41로 바뀌어 설정되었지만 이는 /etc/passwd 파일의 설정값만 바꾼 것이며 실제로 /home/user41 디렉터리가 생성되지는 않음. 별도로 디렉터를 만들고 파일을 옮겨야 함

```
[root@localhost ~]# ls /home/user41
ls: cannot access '/home/user41': 그런 파일이나 디렉터리가 없습니다
```

## 02 사용자 계정 관리 명령

### ■ 로그인 ID 변경하기: -l 옵션

- -l 옵션으로 사용자 계정의 로그인 ID를 바꿀 수 있음
- 로그인 ID를 바꿀 때는 홈 디렉터리도 같이 바꿔주는 것이 좋음
- 기존 홈 디렉터리에 있던 파일이 자동으로 옮겨지지 않으므로 별도로 디렉터리를 생성하고 이동
- user4 계정의 로그인 ID를 user44로 바꾸면서 홈 디렉터리도 같이 변경함

```
[root@localhost ~]# usermod -d /home/user44 -l user44 user4  
[root@localhost ~]# grep user44 /etc/passwd  
user44:x:1500:1500::/home/user44:/bin/bash
```

## 02 사용자 계정 관리 명령

### ■ 패스워드 에이징 관련 명령

- 패스워드 에이징은 useradd, usermod, passwd, chage 명령으로 설정할 수 있음.  
chage는 패스워드 에이징을 관리하는 별도의 명령
- 패스워드 에이징 항목과 명령의 관계

표 9-1 패스워드 에이징 관련 명령

항목	useradd, usermod, passwd 명령	chage 명령
MIN	passwd -n 날수	chage -m
MAX	passwd -x 날수	chage -M
WARNING	passwd -w 날수	chage -W
INACTIVE	useradd -f 날수	chage -I(대문자 i)
	usermod -f 날수	
EXPIRE	useradd -e 날짜(YYYY-MM-DD)	chage -E
	usermod -e 날짜(YYYY-MM-DD)	

## 02 사용자 계정 관리 명령

### ■ 기존 명령으로 패스워드 에이징 변경하기

- user3 계정의 패스워드 에이징을 변경해 보자. 현재 user3의 패스워드 에이징 설정값은 다음과 같음

```
[root@localhost ~]# grep user3 /etc/shadow
user3:!!:19609:0:99999:7::0:
```

- usermod 명령과 passwd 명령으로 MIN, MAX, WARNING, INACTIVE, EXPIRE를 다음과 같이 설정할 수 있음

```
[root@localhost ~]# usermod -f 10 -e 2023-12-31 user3
[root@localhost ~]# passwd -n 2 -x 100 -w 5 user3
user3 사용자의 사용 기한 데이터 조정 중
passwd: 성공
[root@localhost ~]# grep user3 /etc/shadow
user3:!!:19609:2:100:5:10:19722:
```

- user3 계정의 패스워드 에이징이 MIN은 2일, MAX는 100일, WARNING은 5일, INACTIVE는 10일, EXPIRE는 2023년 12월 31일로 설정되었음

## 02 사용자 계정 관리 명령

### ■ 패스워드 에이징 변경 및 설정 내용 확인하기: chage

- chage 명령을 사용하여 user44 계정의 패스워드 에이징을 user3과 같은 값으로 설정해 보자

```
[root@localhost ~]# chage -m 2 -M 100 -W 5 -I 10 -E 2023-12-31 user44
[root@localhost ~]# grep user44 /etc/shadow
user44:!!:19609:2:100:5:10:19722:
```

- chage 명령에는 설정 내용을 확인하는 -l(소문자 l) 옵션과 사용자가 암호를 변경한 마지막 날짜를 알려주는 -d 옵션이 있음. chage 명령으로 user44 계정의 패스워드 에이징을 확인해 보기

```
[root@localhost ~]# chage -l user44
마지막으로 암호를 바꾼 날      : 9월 09, 2023
암호 만료                      :12월 18, 2023
암호가 비활성화 기간          :12월 28, 2023
계정 만료                      :12월 31, 2023
암호를 바꿀 수 있는 최소 날 수 : 2
암호를 바꿔야 하는 최대 날 수  : 100
암호 만료 예고를 하는 날 수    : 5
```

## 02 사용자 계정 관리 명령

### ■ 사용자 계정 삭제

- userdel 명령은 사용자 계정을 삭제함
- 사용자 계정을 삭제할 때 홈 디렉터리를 함께 삭제 할 것인지 결정해야 함

#### userdel

- 기능 사용자 계정을 삭제한다.
- 형식 userdel [옵션] [로그인 ID]
- 옵션 -r: 홈 디렉터리를 삭제한다.  
-f: 사용자가 로그인 중이어도 강제로 삭제한다.
- 사용 예 userdel user4  
userdel -r user4



## 02 사용자 계정 관리 명령

### ■ 사용자 계정과 홈 디렉터리 삭제하기

- userdel 명령에서 -r 옵션을 지정하면 사용자 계정과 홈 디렉터리가 함께 삭제됨
- 다음 예에서는 user3 사용자 계정을 홈 디렉터리와 함께 삭제함.

그런데 user3을 삭제했더니 현재 사용 중이라고 함.

이는 다른 터미널에서 su 명령을 입력하여 user2 계정으로 접속해서 로그인한 상태인데, 앞의 실습에서 user2와 user3의 UID를 중복해 놓았기 때문에 같은 계정으로 처리된 것

```
[root@localhost ~]# userdel -r user3
userdel: user user3 is currently used by process 46218
```

- 다른 터미널의 user2 계정에서 exit를 한 뒤에 userdel 명령을 다시 실행하면 정상적으로 삭제됨

```
[root@localhost ~]# userdel -r user3
[root@localhost ~]# grep user3 /etc/passwd
[root@localhost ~]# ls /home
user1 user2 user4 user5
```

## 02 사용자 계정 관리 명령

### ■ 사용자 계정과 홈 디렉터리 삭제하기

- 홈 디렉터리가 아닌 다른 곳에 위치한 사용자 계정 소유의 파일이 있을 수 있음
- -r 옵션은 홈 디렉터리의 파일은 삭제하지만 다른 디렉터리에 있는 파일은 삭제하지 않음
- 관리자는 find 명령으로 해당 계정 소유의 파일을 모두 찾아서 삭제하는 것이 바람직

```
find / -user UID -exec rm -r {} \;
```

- user5(UID 2001) 계정과 관련된 파일을 검색하여 삭제하려면 다음과 같이 명령함

```
[root@localhost ~]# find / -user 2001 -exec rm -r {} \;  
[root@localhost ~]# ls /home  
user1  user2  user4
```

- 실행 결과 /home/user5 디렉터리가 삭제됨

## 02 사용자 계정 관리 명령

### ■ 사용자 계정 생성·수정·삭제하기

#### ① 다음 표를 참고하여 test01, test02, test03 계정을 만듦

로그인 ID	로그인 셸	UID	2차 그룹	설명
test01	본셸	2100	3	test01 user
test02	배시셸	2200	4	test02 user
test03	배시셸	2300	5	test03 user

```
[root@localhost ~]# useradd -u 2100 -G 3 -s /bin/sh -c "test01 user" test01
[root@localhost ~]# useradd -u 2200 -G 4 -s /bin/bash -c "test02 user" test02
[root@localhost ~]# useradd -u 2300 -G 5 -s /bin/bash -c "test03 user" test03
```

#### ② 앞의 ①서 생성한 사용자 계정을 확인

```
[root@localhost ~]# grep test /etc/passwd
test01:x:2100:2100:test01 user:/home/test01:/bin/sh
test02:x:2200:2200:test02 user:/home/test02:/bin/bash
test03:x:2300:2300:test03 user:/home/test03:/bin/bash
```

## 02 사용자 계정 관리 명령

### ■ 사용자 계정 생성·수정·삭제하기

#### ③ test01 사용자에게 다음과 같은 패스워드 에이징을 설정

항목	MIN	MAX	WARNING	INACTIVE	EXPIRE
설정값	4	200	10	5	2024-03-31

```
[root@localhost ~]# chage -m 4 -M 200 -W 10 -I 5 -E 2024-03-31 test01
[root@localhost ~]# grep test01 /etc/shadow
test01:!!:19609:4:200:10:5:19813:
```

#### ④ test03 계정의 UID를 2010으로, 계정의 이름을 test33으로 수정

```
[root@localhost ~]# usermod -u 2010 -l test33 test03
[root@localhost ~]# grep test33 /etc/passwd
test33:x:2010:2300:test03 user:/home/test03:/bin/bash
```

## 02 사용자 계정 관리 명령

### ■ 사용자 계정 생성·수정·삭제하기

#### ⑤ test02 사용자를 홈 디렉터리까지 삭제함

```
[root@localhost ~]# userdel -r test02
[root@localhost ~]# grep test /etc/passwd
test01:x:2100:2100:test01 user:/home/test01:/bin/sh
test33:x:2010:2300:test03 user:/home/test03:/bin/bash
[root@localhost ~]# ls /home
test01 test03 user1 user2 user4
```

## 02 사용자 계정 관리 명령

### ■ 사용자 계정 생성·수정·삭제하기

혼자해보기 사용자 계정 생성 · 수정 · 삭제하기

❶ 다음 표를 참조하여 사용자 계정을 생성한다.

로그인 ID	로그인 셸	UID	2차 그룹	설명
han01	본셸	1500	1000	han01 user
han02	배시셸	1501	1001	han02 user

❷ 앞의 ❶번에서 생성한 사용자 계정을 `/etc/passwd` 파일에서 확인한다.

❸ han01 사용자 계정의 이름을 han11로, han02 사용자의 UID를 1600으로 수정한다.

❹ `usermod`, `passwd` 명령으로 han02 사용자의 패스워드 에이징을 설정한다.

항목	MIN	MAX	WARNING	INACTIVE	EXPIRE
설정값	3	150	5	7	2020-10-31

❺ `chage` 명령으로 han02 사용자의 패스워드 에이징 정보를 확인한다.

❻ han01, han02 사용자 계정을 `홈 디렉터리`까지 모두 삭제한다.

## 03 그룹 관리 명령

## 03 그룹 관리 명령

### ■ 그룹 생성

- 새로 생성된 그룹은 /etc/group 파일에 저장됨

#### groupadd

- 기능 그룹을 생성한다.
- 형식 `groupadd [옵션] [그룹명]`
- 옵션 `-g gid`: 그룹의 GID를 지정한다.  
`-o`: GID의 중복을 허용한다.
- 사용 예 `groupadd gtest`  
`groupadd -g 2020 gtest`  
`groupadd -g 2020 -o gtest`



## 03 그룹 관리 명령

### ■ 옵션 없이 그룹 생성하기

- 옵션을 지정하지 않고 groupadd로 새로운 그룹을 생성해 보자.  
옵션이 없으면 GID를 가장 마지막 번호의 다음 번호로 자동 설정함

```
[root@localhost ~]# groupadd gtest01
[root@localhost ~]# grep gtest /etc/group
gtest01:x:2301:
```

### ■ GID를 지정하여 그룹 생성하기

- 특정 GID를 지정하여 그룹을 생성할 수도 있음. GID를 지정하려면 -g 옵션을 사용함

```
[root@localhost ~]# groupadd -g 3000 gtest02
[root@localhost ~]# grep gtest02 /etc/group
gtest02:x:3000:
```

## 03 그룹 관리 명령

### ■ GID를 중복하여 지정하기

- 그룹을 생성할 때 GID를 기존 GID와 중복으로 지정할 수 있음.  
gtest02 그룹과 GID가 같은 gtest03 그룹을 생성해 보자

```
[root@localhost ~]# groupadd -g 3000 -o gtest03
[root@localhost ~]# grep gtest /etc/group
gtest01:x:2301:
gtest02:x:3000:
gtest03:x:3000:
```

## 03 그룹 관리 명령

### ■ 그룹 정보 수정

- groupmod 명령은 그룹의 GID나 그룹명을 수정할 때 사용

#### groupmod

- 기능 그룹 정보를 수정한다.
- 형식 `groupmod [옵션] [그룹명]`
- 옵션
  - g gid: 그룹의 GID를 수정한다.
  - o: GID의 중복을 허용한다.
  - n 그룹명: 그룹명을 다른 이름으로 바꾼다.
- 사용 예
  - `groupmod -g 3001 gtest`
  - `groupmod -g 3001 -o gtest2`
  - `groupmod -n gtest1 gtest`

## 03 그룹 관리 명령

### ■ GID 바꾸기

- groupmod 명령에 -g 옵션을 지정하면 GID를 바꿀 수 있음
- 다음 예에서는 gtest01 그룹의 GID를 2301에서 2500으로 바꿨음

```
[root@localhost ~]# groupmod -g 2500 gtest01
[root@localhost ~]# grep gtest01 /etc/group
gtest01:x:2500:
```

### ■ 그룹명 바꾸기

- 그룹명을 바꾸려면 -n 옵션을 사용함

그룹명이 바뀌면 기존 파일이나 디렉터리의 소속 그룹명도 자동으로 바뀜

```
[root@localhost ~]# groupmod -n gtest11 gtest01
[root@localhost ~]# grep gtest /etc/group
gtest02:x:3000:
gtest03:x:3000:
gtest11:x:2500:
```

## 03 그룹 관리 명령

### ■ 그룹 삭제

#### groupdel

- 기능 그룹을 삭제한다.
- 형식 groupdel [그룹명]
- 사용 예 groupdel gtest

- 삭제할 그룹의 이름만 지정하면 해당 그룹이 삭제됨

```
[root@localhost ~]# groupdel gtest03
[root@localhost ~]# grep gtest /etc/group
gtest02:x:3000:
gtest11:x:2500:
```

## 03 그룹 관리 명령

### ■ 그룹 암호 설정하고 사용하기

- 리눅스는 유닉스와 달리 그룹의 암호를 설정하는 gpasswd 명령을 제공
- gpasswd 명령은 그룹의 암호를 삭제하고 그룹에 멤버를 추가하거나 삭제하는 기능도 수행

#### gpasswd

- 기능 /etc/group와 /etc/gshadow 파일을 관리한다.
- 형식 gpasswd [옵션] [그룹명]
- 옵션
  - a 사용자 계정: 사용자 계정을 그룹에 추가한다.
  - d 사용자 계정: 사용자 계정을 그룹에서 삭제한다.
  - r: 그룹 암호를 삭제한다.
- 사용 예
  - gpasswd gtest01
  - gpasswd -a test01 gtest01
  - gpasswd -r gtest01

## 03 그룹 관리 명령

### ■ 그룹에 멤버 추가하기

- 그룹에 멤버를 추가하려면 -a 옵션을 사용
- test01, test33 계정을 gtest11 그룹에 추가해보자

```
[root@localhost ~]# grep gtest11 /etc/group
gtest11:x:2500:
```

- gtest11 그룹을 확인해 보면 멤버가 하나도 없음
- gtest11 그룹에 멤버를 추가하는 명령은 다음과 같음

```
[root@localhost ~]# gpasswd -a test01 gtest11
사용자 test01을(를) gtest11 그룹에 등록 중
[root@localhost ~]# gpasswd -a test33 gtest11
사용자 test33을(를) gtest11 그룹에 등록 중
[root@localhost ~]# grep gtest11 /etc/group
gtest11:x:2500:test01,test33
```

## 03 그룹 관리 명령

### ■ 그룹의 멤버 삭제하기

- 그룹의 멤버를 삭제하려면 -d 옵션을 사용
- gtest11 그룹에 추가한 test01, test33 계정 중에서 test33 사용자를 삭제해 보자

```
[root@localhost ~]# gpasswd -d test33 gtest11
사용자 test33을(를) 그룹 gtest11에서 제거하는 중
[root@localhost ~]# grep gtest11 /etc/group
gtest11:x:2500:test01
```



## 03 그룹 관리 명령

### ■ 그룹 암호 설정하고 제거하기

- 그룹에 암호를 설정하고 제거해 보자
- gtest11 그룹에 암호를 설정함

```
[root@localhost ~]# gpasswd gtest11
gtest11 그룹의 암호를 바꾸는 중
새 암호:
새 암호를 다시 입력하십시오:
[root@localhost ~]# grep gtest11 /etc/gshadow
gtest11:$6$rKmSMEzmf6fF6/ar$.Wn88gehlun1NCbsMcFv0zfhkjTFRIZ/yU4Rw0ptexizkywVZR08mUUK
JWgacykuBKsuj01LFHLGcVXwb8J.::test01
```

- /etc/gshadow 파일에 암호가 설정되었음

## 03 그룹 관리 명령

### ■ 소속 그룹 변경하기: newgrp

- 사용자가 작업을 할 때 현재 소속 그룹을 기본 그룹이 아닌 다른 그룹으로 변경해야 할 일이 생길 수 있음. 이때 사용하는 명령이 newgrp 명령

#### newgrp

- 기능 소속 그룹을 다른 그룹으로 바꾼다.
- 형식 newgrp [그룹명]
- 사용 예 newgrp adm

- user1의 현재 소속 그룹을 확인해 보자. 소속 그룹을 확인하려면 id 명령을 사용함

```
[root@localhost ~]# id user1
uid=1000(user1) gid=1000(user1) groups=1000(user1),10(wheel)
```

## 03 그룹 관리 명령

### ■ 소속 그룹 변경하기: newgrp

- user1의 소속 그룹은 user1과 wheel. user1의 2차 그룹으로 adm을 추가로 등록함

```
[root@localhost ~]# gpasswd -a user1 adm
사용자 user1을(를) adm 그룹에 등록 중
[root@localhost ~]# id user1
uid=1000(user1) gid=1000(user1) groups=1000(user1),4(adm),10(wheel)
```

- user1의 소속 그룹을 2차 그룹인 adm으로 바꿔보자

```
[user1@localhost ~]$ newgrp adm
[user1@localhost ~]$ id
uid=1000(user1) gid=4(adm) groups=4(adm),10(wheel),1000(user1) context=unconfined_
u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

- 이제 newgrp 명령으로 user1의 소속 그룹을 user1 계정의 2차 그룹이 아닌 gtest11로 바꿔보자

```
[user1@localhost ~]$ newgrp gtest11
암호:
```

## 03 그룹 관리 명령

### ■ 소속 그룹 변경하기: newgrp

- adm 그룹은 user1의 2차 그룹이기 때문에 바로 바뀌었지만, gtest11 그룹은 user1의 2차 그룹이 아니기 때문에 암호를 입력하라함. 이때 입력하는 것이 그룹 암호.  
gtest11의 그룹 암호를 입력하면 소속 그룹이 변경됨

```
[user1@localhost ~]$ id
uid=1000(user1) gid=2500(gtest11) groups=2500(gtest11),4(adm),10(wheel),1000(user1)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

- user1 계정의 그룹을 처음의 그룹으로 변경함

```
[user1@localhost ~]$ newgrp user1
[user1@localhost ~]$ id
uid=1000(user1) gid=1000(user1) groups=1000(user1),4(adm),10(wheel),2500(gtest11)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

## 03 그룹 관리 명령

### ■ 그룹 암호 삭제하기: -r 옵션

- gtest11 그룹에 설정된 암호를 삭제해보자.  
그룹 암호를 삭제하려면 -r 옵션을 사용함

```
[root@localhost ~]# gpasswd -r gtest11
[root@localhost ~]# grep gtest11 /etc/gshadow
gtest11:::test01
```

## 03 그룹 관리 명령

### ■ 그룹 생성·수정·삭제하고 암호 설정하기

#### ① grp01, grp02 그룹을 생성하고 확인

```
[root@localhost ~]# groupadd grp01
[root@localhost ~]# groupadd grp02
[root@localhost ~]# grep grp /etc/group
grp01:x:3001:
grp02:x:3002:
```

#### ② grp02의 GID를 3100으로 바꿈

```
[root@localhost ~]# groupmod -g 3100 grp02
[root@localhost ~]# grep grp /etc/group
grp01:x:3001:
grp02:x:3100:
```

## 03 그룹 관리 명령

### ■ 그룹 생성·수정·삭제하고 암호 설정하기

#### ③ grp02를 grp22로 바꿈

```
[root@localhost ~]# groupmod -n grp22 grp02
[root@localhost ~]# grep grp /etc/group
grp01:x:3001:
grp22:x:3100:
```

#### ④ grp01 그룹에 user1 사용자를, grp22 그룹에 user2 사용자를 추가

```
[root@localhost ~]# gpasswd -a user1 grp01
사용자 user1을(를) grp01 그룹에 등록 중
[root@localhost ~]# gpasswd -a user2 grp22
사용자 user2을(를) grp22 그룹에 등록 중
[root@localhost ~]# grep grp /etc/group
grp01:x:3001:user1
grp22:x:3100:user2
```

## 03 그룹 관리 명령

### ■ 그룹 생성·수정·삭제하고 암호 설정하기

#### ⑤ grp22 그룹에 암호를 설정

```
[root@localhost ~]# gpasswd grp22
grp22 그룹의 암호를 바꾸는 중
새 암호:
새 암호를 다시 입력하십시오:
```

#### ⑥ 두 번째 터미널에서 user1의 소속 그룹을 grp22로 변경

```
[user1@localhost ~]$ newgrp grp22
암호:
[user1@localhost ~]$ id
uid=1000(user1) gid=3100(grp22) groups=3100(grp22),4(adm),10(wheel),1000(user1),3001(g
rp01) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```



## 03 그룹 관리 명령

### ■ 그룹 생성·수정·삭제하고 암호 설정하기

#### ⑦ grp01, grp22 그룹을 삭제

```
[root@localhost ~]# groupdel grp01
[root@localhost ~]# groupdel grp22
[root@localhost ~]# grep grp /etc/group
[root@localhost ~]#
```

**혼자해보기** 그룹 생성 · 수정 · 삭제하기, 멤버 등록 · 삭제하기, 암호 설정하기

- ① grphan 그룹을 생성하고 grpbook으로 변경한다.
- ② user1, user2 계정을 grpbook 그룹에 추가한다.
- ③ grpbook 그룹에 암호를 설정한다.
- ④ user2 계정을 grpbook 그룹에서 삭제한다.
- ⑤ user1, user2 계정의 현재 소속 그룹을 grpbook 그룹으로 변경한다.
- ⑥ grpbook 그룹을 삭제한다.

## 04 사용자 정보 관리 명령

## 04 사용자 정보 관리 명령

### ■ UID와 EUID

- UID는 실제 UID로 RUID라고도 하며, EUID는 유효 사용자 ID라고 함
- UID(RUID)는 사용자가 로그인할 때 사용한 계정의 UID를 의미하고, EUID는 현재 명령을 수행하는 주체의 UID를 의미
- 보통의 경우 UID와 EUID는 같으나, UID 와 EUID가 달라지는 두 가지 경우가 있음
  - **실행 파일에 SetUID가 설정된 경우:** 해당 파일을 실행한 프로세스의 UID는 사용자 계정의 UID가 아니라 실행 파일 소유자의 UID가 됨. 이때 실행 파일 소유자의 UID를 EUID라고 함.
  - **su 명령을 사용하여 다른 계정으로 전환한 경우:** 사용자가 로그인하면 처음에는 UID와 EUID가 같은 값을 갖음. 하지만 su 명령을 사용하여 다른 사용자로 전환하면 UID와 EUID가 달라짐

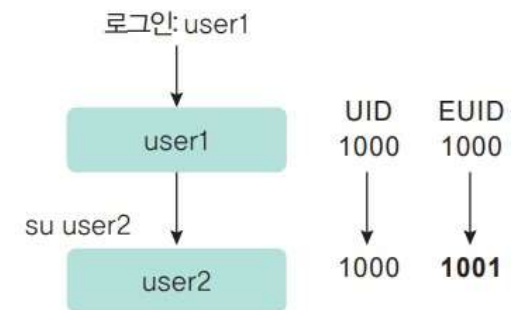


그림 9-5 UID와 EUID의 관계

## 04 사용자 정보 관리 명령

### ■ who 명령

- 현재 시스템에 로그인한 사용자의 이름, 로그인한 단말기 번호, 로그인 시간 등의 정보를 출력

#### who

- 기능 현재 시스템을 사용하는 사용자의 정보를 출력한다.
- 형식 `who [옵션]`
- 옵션
  - q: 사용자명만 출력한다.
  - H: 출력 항목의 제목도 함께 출력한다.
  - b: 마지막으로 재부팅한 날짜와 시간을 출력한다.
  - m: 현재 사용자 계정의 정보를 출력한다.
  - r: 현재 런레벨을 출력한다.
- 사용 예 `who`  
`who -b`

## 04 사용자 정보 관리 명령

### ■ who 명령

- who 명령을 옵션 없이 사용하면 사용자명 외에 부가적인 정보가 출력됨
- 사용자명, 사용자가 접속한 단말기 번호, 로그인한 시간, 외부 네트워크에서 접속했을 경우 외부 컴퓨터 이름이나 IP 주소가 출력됨

```
[root@localhost ~]# who
user1 pts/0 2023-09-09 19:02 (192.168.147.1)
user1 seat0 2023-09-03 12:34 (login screen)
user1 tty2 2023-09-03 12:34 (tty2)
user1 pts/3 2023-09-10 11:01 (192.168.147.1)
user1 pts/4 2023-09-10 12:24 (192.168.147.1)
```

## 04 사용자 정보 관리 명령

### ■ who 명령

- who -H 명령은 다음과 같이 각 항목의 제목을 출력

```
[root@localhost ~]# who -H
```

NAME	LINE	TIME	COMMENT
user1	pts/0	2023-09-09 19:02	(192.168.147.1)
user1	seat0	2023-09-03 12:34	(login screen)

(생략)

- who -q 명령은 로그인한 사용자 수와 계정 이름을 출력

```
[root@localhost ~]# who -q
```

user1 user1 user1 user1 user1

# users=5

## 04 사용자 정보 관리 명령

### ■ who 명령

- who -b 명령은 사용자 정보가 아닌, 시스템이 마지막으로 부팅한 날짜와 시간 정보를 출력

```
[root@localhost ~]# who -b
system boot 2023-08-31 21:54
```

- who -r 명령은 현재 런레벨을 출력

```
[root@localhost ~]# who -r
run-level 5 2023-08-31 21:54
```

## 04 사용자 정보 관리 명령

### ■ w 명령

- 현재 시스템에 로그인한 사용자의 정보 외에 사용자가 현재 실행 중인 작업에 대한 정보를 알려줌
- w 명령의 인자로 사용자명을 지정하면 그 사용자의 정보만 출력함

w

- 기능 현재 시스템을 사용하는 사용자의 정보와 작업 정보를 출력한다.
- 형식 w [사용자명]

```
[root@localhost ~]# w
12:39:11 up 1 day, 1:28, 5 users, load average: 0.36, 0.23, 0.19
USER  TTY      LOGIN@  IDLE   JCPU   PCPU   WHAT
user1 pts/0    토19    14:35m  0.16s  0.04s  sshd: user1 [priv]
user1 seat0 03  9월23   0.00s  0.00s  0.03s  /usr/libexec/gdm-wayland-session
--register-session gn
user1 tty2 03  9월23   9days  0.10s  0.07s  /usr/libexec/gnome-session-binary
user1 pts/3    11:01   0.00s  0.09s  0.03s  sshd: user1 [priv]
user1 pts/4    12:24   14:38  0.02s  0.01s  bash
```

who 명령에서 나온 출력 외에  
활동에 대한 정보와 CPU 사용량,  
시스템 로드 정보 등이 추가로  
출력됨



## 04 사용자 정보 관리 명령

### ■ last 명령

- 사용자명, 로그인한 시간, 로그아웃한 시간, 터미널 번호나 IP 주소를 출력  
누가 언제 로그인해서 로그아웃했는지를 한눈에 알 수 있음

#### last

- 기능 시스템에 로그인하고 로그아웃한 정보를 출력한다.
- 형식 last

```
[root@localhost ~]# last
user1 pts/4 192.168.147.1 Sun Sep 10 12:24 still logged in
user1 pts/3 192.168.147.1 Sun Sep 10 11:01 still logged in
(생략)
user1 tty2 tty2 Sun Jul 9 10:48 - down (00:08)
user1 seat0 login screen Sun Jul 9 10:48 - down (00:08)
reboot system boot 5.14.0-284.11.1. Sun Jul 9 10:47 - 10:57 (00:10)

wtmp begins Sun Jul 9 10:47:19 2023
```

시스템이 다운된 시간, 다시  
부팅한 시간, root 사용자가  
로그인한 시간 등 을 알 수 있음

## 04 사용자 정보 관리 명령

### ■ UID와 EUID 확인하기: whoami, who am i, id 명령

- 현재 작업 중인 내가 누구인지, UID와 EUID는 무엇인지 확인하는 명령은 whoami, who am i, id 명령
  - UID 출력: who am i, who -m
  - EUID 출력: whoami, id
- user1로 로그인하여 위의 세 가지 명령을 실행하면 다음과 같이 모두 user1(1000)을 출력함

```
[user1@localhost ~]$ whoami
user1
[user1@localhost ~]$ id
uid=1000(user1) gid=1000(user1) groups=1000(user1),4(adm),10(wheel),2500(gtest11)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[user1@localhost ~]$ who am i
user1 pts/3 2023-09-10 11:01 (192.168.147.1)
```

## 04 사용자 정보 관리 명령

### ■ UID와 EUID 확인하기: whoami, who am i, id 명령

- su 명령을 사용하여 사용자를 user2로 전환하고 다시 앞의 세 가지 명령을 실행해 보면 다음과 같이 출력

```
[user1@localhost ~]$ su - user2
암호:
[user2@localhost ~]$ whoami
user2
[user2@localhost ~]$ id
uid=1001(user2) gid=1001(user2) groups=1001(user2) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[user2@localhost ~]$ who am i
user1 pts/3 2023-09-10 11:01 (192.168.147.1)
[user2@localhost ~]$ who -m
user1 pts/3 2023-09-10 11:01 (192.168.147.1)
```

- whoami와 id 명령은 user2를 출력하지만 who am i(who -m과 동일) 명령은 user1을 출력함
- 즉, who am i는 UID(RUID)를 출력하고 whoami와 id는 EUID를 출력함

## 04 사용자 정보 관리 명령

### ■ 소속 그룹 확인하기: groups 명령

#### groups

- 기능 사용자 계정이 속한 그룹을 출력한다.
- 형식 groups [계정명]

- 특별히 사용자를 지정하지 않고 groups 명령을 실행하면 현재 사용자 계정이 속한 그룹을 출력함

```
[user1@localhost ~]$ groups  
user1 adm wheel gtest11
```

- 사용자 계정을 인수로 지정하면 해당 사용자 계정이 속한 그룹 정보를 출력함 □

```
[user1@localhost ~]$ groups user2  
user2 : user2
```

## 04 사용자 정보 관리 명령

### ■ root 권한 사용 방법

- 일반 사용자가 root 권한을 사용하는 방법은 두 가지
- **su 명령을 사용하여 root 계정으로 전환하는 것:** 간단하지만, root 계정으로 전환하는 것이므로 일반 사용자가 모든 시스템 관리 권한을 가지게 되어 보안상 매우 위험  
예를 들어 시스템 관리자가 출장을 가면서 필요시 시스템 관리 작업을 할 수 있도록 su 명령을 사용하여 일반 사용자가 root로 전환되게 했다면 위험한 일이기 때문
- **일반 사용자에게 시스템 관리 작업 중 특정 작업만 수행할 수 있는 권한을 주는 것:**  
예를 들면 사용자를 추가하거나 시스템을 종료할 수 있는 권한만 부여하는 것인데, sudo 명령으로 이와 같은 제한적인 권한 부여가 가능

## 04 사용자 정보 관리 명령

### ■ sudo 권한 설정하기

- 일반 사용자가 sudo 명령으로 root 권한을 실행하려면 특정 권한을 부여받아야 함
- 이 권한은 /etc/sudoers 파일에 설정함. 이 파일은 root 계정으로만 수정할 수 있음.  
/etc/sudoers 파일은 vi로 편집할 수 있지만 visudo 명령을 사용해 수정하길 권함
- visudo 명령은 파일을 수정한 후 문법이 맞는지 확인하고 저장
- /etc/sudoers 파일의 설정 형식은 다음과 같음

```
사용자 계정 호스트 = 명령어
```

- etc/sudoers 파일을 보면 root 계정의 경우 다음과 같이 설정되어 있음.  
어디서나 어느 명령이든지 실행할 수 있음을 뜻함

```
root ALL= (ALL) ALL
```

## 04 사용자 정보 관리 명령

### ■ sudo 권한 설정하기

- 특정 사용자에게 특정 명령을 수행할 수 있는 권한을 부여하려면 /etc/sudoers 파일에 다음과 같이 추가함
- user2 사용자에게 localhost에서 사용자를 추가하고 수정할 수 있는 권한을 부여하는 예

```
user2 localhost= /sbin/useradd, /sbin/usermod
```

- 명령은 절대 경로로 표시하고, 명령이 여러 개이면 쉼표로 구분함

## 04 사용자 정보 관리 명령

### ■ sudo 명령 사용하기

- sudo 명령은 다음과 같은 형식으로 사용함

```
sudo 명령
```

- 예를 들어 user2 사용자가 사용자 계정을 추가하려고 한다면 다음과 같이 작성함. 잠깐이기는 하지만 root 계정의 권한을 사용하는 것이므로 주의하라는 메시지가 뜬

```
[user2@localhost ~]$ sudo useradd han11
```

로컬 시스템 관리자에게 일반적인 지침을 받았으리라 믿습니다.  
보통 세가지로 요약합니다:

- #1) 타인의 사생활을 존중하십시오.
- #2) 입력하기 전에 한 번 더 생각하십시오.
- #3) 막강한 힘에는 상당한 책임이 뒤따릅니다.

```
[sudo] user2의 암호:
```



## 04 사용자 정보 관리 명령

### ■ sudo 명령 사용하기

- 이때 root 계정의 암호가 아니라 user2 계정의 암호를 입력하면 사용자 계정이 추가됨

```
[user2@localhost ~]$ grep han /etc/passwd  
han11:x:2101:2101::/home/han11:/bin/bash
```

- sudo 명령 사용 시 암호를 입력하는 과정을 생략하려면 /etc/sudoers 파일에 다음과 같이 설정함

```
user2 localhost=NOPASSWD:/sbin/useradd, /sbin/usermod
```

- user2는 sudo 명령으로 useradd, usermod만 수행할 수 있음.  
만약 userdel로 han11 계정을 삭제하려고 하면 어떻게 될까?

```
[user2@localhost ~]$ sudo userdel han11  
죄송하지만 user2 사용자는 '/sbin/userdel han11'을(를) localhost.localdomain의 root(으)로  
실행하도록 허가받지 않았습니다.
```

- userdel 명령에 대한 권한이 없다는 메시지가 출력됨

## 04 사용자 정보 관리 명령

### ■ sudo 명령 사용 시 주의 사항

- sudo 명령을 사용할 때 일반 사용자에게 일부 권한을 적절하게 지정하여 효과를 볼 수 있지만 이에 따른 문제도 있으므로 주의해야 함
- 특히 관리하기 귀찮다는 이유로 일반 사용자에게 모든 권한을 부여하는 것은 매우 위험.  
해당 사용자 계정의 암호가 유출되면 root 계정의 암호가 유출된 것이나 마찬가지이기 때문
- 따라서 sudo 명령을 사용할 수 있는 사용자의 선정과 명령 선택은 매우 신중하게 처리해야 한다는 점을 명심해야함

## 04 사용자 정보 관리 명령

### ■ passwd 명령 활용하기

- passwd 명령은 암호를 설정할 때만 사용하는 명령이 아님

#### passwd

- 기능 사용자 계정의 암호를 수정한다.
- 형식 passwd [옵션] [사용자 계정]
- 옵션
  - l 사용자 계정: 지정한 계정의 암호를 잠근다.
  - u 사용자 계정: 암호 잠금을 해제한다.
  - d 사용자 계정: 지정한 계정의 암호를 삭제한다.

## 04 사용자 정보 관리 명령

### ■ 암호 잠그고 해제하기: -l 옵션과 -u 옵션

- 사용자 계정의 암호를 잠글 때는 -l 옵션을 사용함
- 다음은 user2 계정의 암호를 잠그는 예  
/etc/shadow 파일을 보면 암호의 맨 앞에 !!가 있는데 이는 암호가 잠겼음을 의미

```
[root@localhost user1]# passwd -l user2
user2 사용자의 비밀 번호를 잠급니다
passwd: 성공
[root@localhost user1]# grep user2 /etc/shadow
user2:!!$6$ieCx5vmzGWWxhtYV$pmhQZizXoEK9R49W.i7NhnUzniS6b80xFqSKJpgA2y8bps7f4tvv40w
1x/gQNdwI9jkWugNsRWI59yT76YGYL:19609:0:99999:7:::
```

## 04 사용자 정보 관리 명령

### ■ 암호 잠그고 해제하기: -l 옵션과 -u 옵션

- 암호의 잠금을 해제하려면 -u 옵션을 사용
- 다음은 앞에서 수행한 user2의 암호 잠금을 해제하는 예.  
암호 항목에서 !!가 사라짐

```
[root@localhost user1]# passwd -u user2
user2 사용자의 비밀번호 잠금 해제 중
passwd: 성공
[root@localhost user1]# grep user2 /etc/shadow
user2:$6$ieCx5vmzGWWxhtYV$pmhQZizXoEK9R49W.i7NhnUzniS6b80xFqSKJpgA2y8bps7f4tvv40w1x/
gQNdwI9jkWugNsRWI59yT76YGYL:19609:0:99999:7:::
```

## 04 사용자 정보 관리 명령

### ■ 암호 삭제하기: -d 옵션

- 사용자 계정의 암호를 삭제할 때는 -d 옵션을 사용
- 다음은 user2 계정의 암호를 삭제하는 예  
/etc/shadow 파일을 보면 암호 자리가 비어 있는데 이는 암호가 설정되어 있지 않음을 의미

```
[root@localhost user1]# passwd -d user2
user2 사용자의 비밀 번호 삭제 중
passwd: 성공
[root@localhost user1]# grep user2 /etc/shadow
user2::19610:0:99999:7:::
```

- 사용자 계정의 암호가 없으면, user2로 로그인하거나 su user2로 사용자를 전환할 때 암호를 묻지 않음. 두 번째 터미널에서 실행하여 확인해 보기

## 04 사용자 정보 관리 명령

### ■ chown 명령

#### chown

- 기능 파일과 디렉터리의 소유자와 소유 그룹을 변경한다.
- 형식 `chown [옵션] [사용자 계정] [파일명/디렉터리명]`
- 옵션 `-R`: 서브 디렉터리의 소유자와 소유 그룹도 변경한다.
- 사용 예 `chown user2 file1`  
`chown user2:grp01 file1`  
`chown -R user2 file1`

- 예제를 통해 chown 명령 사용법을 알아보자.  
먼저 user2 계정에 파일과 디렉터를 생성함

```
[user2@localhost ~]$ mkdir temp
[user2@localhost ~]$ cp /etc/hosts .
[user2@localhost ~]$ cp /etc/services temp
[user2@localhost ~]$ ls
hosts temp
```

## 04 사용자 정보 관리 명령

### ■ chown 명령

- hosts 파일과 temp 디렉터리의 소유자와 그룹을 확인해 보면 다음과 같이 user2로 되어 있음

```
[user2@localhost ~]$ ls -l
합계 4
-rw-r--r--. 1 user2 user2 158 9월 10 13:00 hosts
drwxr-xr-x. 2 user2 user2 22 9월 10 13:01 temp
```

- chown 명령은 일반 사용자가 사용할 수 없고 root 계정에서 쓸 수 있음.  
root 계정으로 로그인된 터미널에서 hosts 파일의 소유자를 user1로 바꿈

```
[root@localhost user1]# cd ~user2
[root@localhost user2]# chown user1 hosts
[root@localhost user2]# ls -l
합계 4
-rw-r--r--. 1 user1 user2 158 9월 10 13:00 hosts
drwxr-xr-x. 2 user2 user2 22 9월 10 13:01 temp
```



## 04 사용자 정보 관리 명령

### ■ chown 명령

- chown 명령으로 소유자와 그룹을 동시에 바꾸려면 다음과 같이 '사용자 계정:그룹명'으로 지정함
- 다음은 hosts 파일의 소유자를 다시 user2로 바꾸고 그룹은 han11로 동시에 바꾸는 예

```
[root@localhost user2]# chown user2:han11 hosts
[root@localhost user2]# ls -l
```

합계 4

-rw-r--r--.	1	user2	han11	158	9월	10	13:00	hosts
drwxr-xr-x.	2	user2	user2	22	9월	10	13:01	temp

## 04 사용자 정보 관리 명령

### ■ chown 명령

- 서브 디렉터리의 파일까지 한 번에 소유자나 소유 그룹을 변경하려면 -R 옵션을 사용함
- temp 디렉터리와 그 안의 파일까지 한 번에 소유자를 user1로, 그룹을 han11로 바꾸려면 다음과 같이 작성함
- temp 디렉터리와 그 아래에 있는 services 파일의 소유자와 그룹이 모두 바뀐 것을 확인할 수 있음

```
[root@localhost user2]# chown -R user1:han11 temp
[root@localhost user2]# ls -l
합계 4
-rw-r--r--. 1 user2 han11 158  9월 10 13:00 hosts
drwxr-xr-x. 2 user1 han11  22  9월 10 13:01 temp
[root@localhost user2]# ls -l temp
합계 680
-rw-r--r--. 1 user1 han11 692252  9월 10 13:01 services
```

## 04 사용자 정보 관리 명령

### ■ chgrp 명령

#### chgrp

- 기능 파일과 디렉터리의 소유 그룹을 변경한다.
- 형식 `chgrp [옵션] [사용자 계정] [파일명/디렉터리명]`
- 옵션 `-R`: 서브 디렉터리의 소유 그룹도 변경한다.
- 사용 예 `chgrp han01 file1`  
`chgrp -R han01 file1`

- 다음 예에서는 temp 디렉터리와 그 아래의 파일까지 한 번에 그룹을 user2로 바꿨음

```
[root@localhost user2]# chgrp -R user2 temp
[root@localhost user2]# ls -l
합계 4
-rw-r--r--. 1 user2 han11 158 9월 10 13:00 hosts
drwxr-xr-x. 2 user1 user2 22 9월 10 13:01 temp
[root@localhost user2]# ls -l temp
합계 680
-rw-r--r--. 1 user1 user2 692252 9월 10 13:01 services
```

# Thank you!