

13-4. 사용자 정보 관리 명령

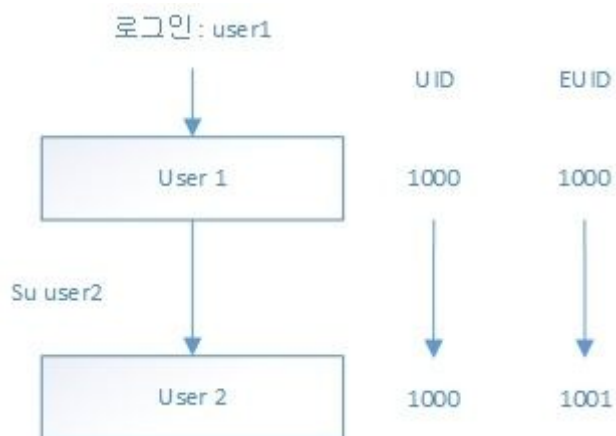
UID와 EUID

UID

- 실제 UID로 RUID라고도 한다.
- 사용자가 로그인할 때 사용한 계정의 UID를 의미한다.

EUID(Effective UID)

- 현재 명령을 수행하는 주체의 UID를 의미한다.
- 일반적으로 UID와 EUID는 같으며, UID와 EUID가 달라지는 경우는 2가지이다.
 - 실행 파일에 setuid가 설정되어 있는 경우
 - su 명령을 사용하여 다른 계정으로 전환한 경우



사용자 확인 명령

사용자 로그인 정보 확인하기: who, w

- **who**
 - 현재 시스템에 로그인한 사용자 이름과 로그인한 단말기 번호, 로그인 시간 등의 정보를 출력한다.

| | |
|------|---|
| 기능 | 현재 시스템을 사용하는 사용자의 정보를 출력한다. |
| 형식 | who 옵션 |
| 옵션 | -q: 사용자의 수와 이름만 출력 -H: 출력 항목의 제목도 함께 출력 -b: 마지막으로 재시작한 날짜와 시간을 출력 -m: 현재 사용자 계정의 정보를 출력 -r: 현재 런 레벨을 출력 |
| 사용 예 | who who -b |

- **w**

- 현재 시스템에 로그인한 사용자의 정보 외에 사용자가 현재 무엇을 하고 있는지 실행 중인 작업의 정보를 알려준다.
- who 명령에서 나온 출력 외의 활동에 대한 정보와 CPU 사용량, 시스템 로드 정보 등이 추가로 출력된다.

| | |
|----|-----------------------------------|
| 기능 | 현재 시스템에 로그인 사용자의 정보와 작업 정보를 출력한다. |
| 형식 | w 사용자명 |

- **last**

- 사용자의 이름과 로그인한 시간, 로그아웃한 시간, 터미널 번호나 IP 주소를 출력한다.
- 누가 언제 로그인해서 로그아웃했는지를 한눈에 알 수 있다.

| | |
|----|----------------------------|
| 기능 | 시스템에 로그인하고 로그아웃한 정보를 출력한다. |
| 형식 | last |

UID와 EUID 확인하기: whoami, who am i, id

- UID 출력: who am i, who -m
- EUID 출력: whoami, id

- **groups**

| | |
|----|----------------------------|
| 기능 | 현재 사용자 계정이 속해 있는 그룹을 확인한다. |
| 형식 | groups (사용자명) |

root 권한 사용하기

su 명령 사용하기

- su 명령을 사용하면 root 계정으로 전환하는 것으로 모든 권한을 가지게 된다.
- **su**
 - substitute user
 - 다른 계정으로 전환하는 리눅스 명령어
 - 로그아웃하지 않고 다른 사용자의 권한으로 셸을 실행
 - 하이픈(-)을 붙이면 변경하려는 사용자의 환경으로 셸 실행
 - logout 또는 exit로 복귀할 수 있음
- 일반 사용자에서 root로 전환하고자 할 때

```
[user1@itserver ~]$ su -  
암호:  
[root@itserver ~]#
```

- root에서 일반 사용자로 전환하고자 할 때

```
[root@itserver ~]# su - user1  
[user1@itserver ~]$
```

sudo 명령 사용하기

- 일반 사용자에게 시스템 관리 작업 중 특정 작업만 수행할 수 있는 권한을 주는 것이다.
- 이 권한은 **/etc/sudoers** 파일에 설정하는데, root 계정만이 수정 가능하다.
- /etc/sudoers 파일은 vi로 편집이 가능하나 **visudo** 명령을 사용할 것을 권장한다.
- /etc/sudoers 파일의 설정형식
 - 사용자계정 호스트=명령어
 - root 계정 설정: 어디서나 어느 명령이든지 실행
 - root ALL=(ALL:ALL) ALL
 - 특정 사용자 설정: user1에 대해 사용자를 추가, 수정 권한을 부여
 - user2 ALL=/sbin/useradd, /sbin/usermod
 - 명령은 절대 경로로 표시하고, 명령이 여러 개일 경우 쉼표로 구분
- sudo 명령 사용하기
 - **sudo 명령**
 - sudo 명령 사용시 암호를 입력하는 과정을 생략하려면 /etc/sudoers 파일에 설정
 - user2 ALL=NOPASSWORD:/sbin/useradd, /sbin/usermod
- sudo 명령 사용시 주의사항
 - 일반 사용자에게 모든 권한을 부여하는 것은 매우 위험하다.

passwd 명령 활용하기

- **passwd**

| | |
|----|---|
| 기능 | 사용자 계정의 암호를 수정한다. |
| 형식 | passwd 옵션 사용자계정 |
| 옵션 | -l: 지정한 계정 잠금 -u: 지정한 계정의 잠금을 해제 -d: 지정한 계정의 암호를 삭제 |

암호를 잠그고 해제하기: -l 옵션/-u 옵션

- 사용자 계정의 암호를 잠글 때는 -l 옵션을 사용한다.
- 사용자 계정의 암호 잠금을 해제할 때는 -u 옵션을 사용한다.

- /etc/shadow 파일을 보면 암호 맨앞에 !!가 있으면 암호가 잠겼음을 의미한다.

암호 삭제하기: -d 옵션

- 사용자 계정의 암호를 삭제할 때는 -d 옵션을 사용한다.
- 암호 삭제 후 /etc/shadow 파일을 보면 암호 자리가 비어 있는데 이는 암호가 설정되지 않았음을 의미한다.

소유자와 소유 그룹 변경하기

• chown

| | |
|------|---|
| 기능 | 파일과 디렉터리의 소유자와 소유 그룹을 변경한다. |
| 형식 | chown 옵션 사용자계정 파일명 |
| 옵션 | -R: 서브 디렉터리의 소유자 및 소유 그룹도 변경 |
| 사용 예 | chown user2 file1 chown user2:grp1 file1 chown -R user2 file1 |

- chown 명령은 일반 사용자는 사용할 수 없고 root 권한으로만 사용할 수 있다.

```
[root@itserver ~]# chown user1 file1
```

- 소유자와 그룹을 동시에 바꾸려면 '사용자계정:그룹명'으로 지정한다.

```
[root@itserver ~]# chown user1:group1 file1
```

- 서브 디렉터리의 파일까지 한번에 변경하려면 -R 옵션을 사용한다.

```
[root@itserver ~]# chown -R user1:group1 dir1
```

• chgrp

| | |
|------|--|
| 기능 | 파일과 디렉터리의 소유 그룹을 변경한다. |
| 형식 | chgrp 옵션 그룹명 파일명/디렉터리명 |
| 옵션 | -R: 서브 디렉터리의 소유 그룹도 변경 |
| 사용 예 | chgrp grp1 file1 chgrp -R grp1 dir1 |

```
[root@itserver ~]# chgrp -R user2 dir1
```