

05

CHAPTER

리눅스 사용자 관리와 파일 관리



C.ontents

01 사용자 관리

02 파일의 소유와 허가권

03 프로세스, 서비스(데몬)

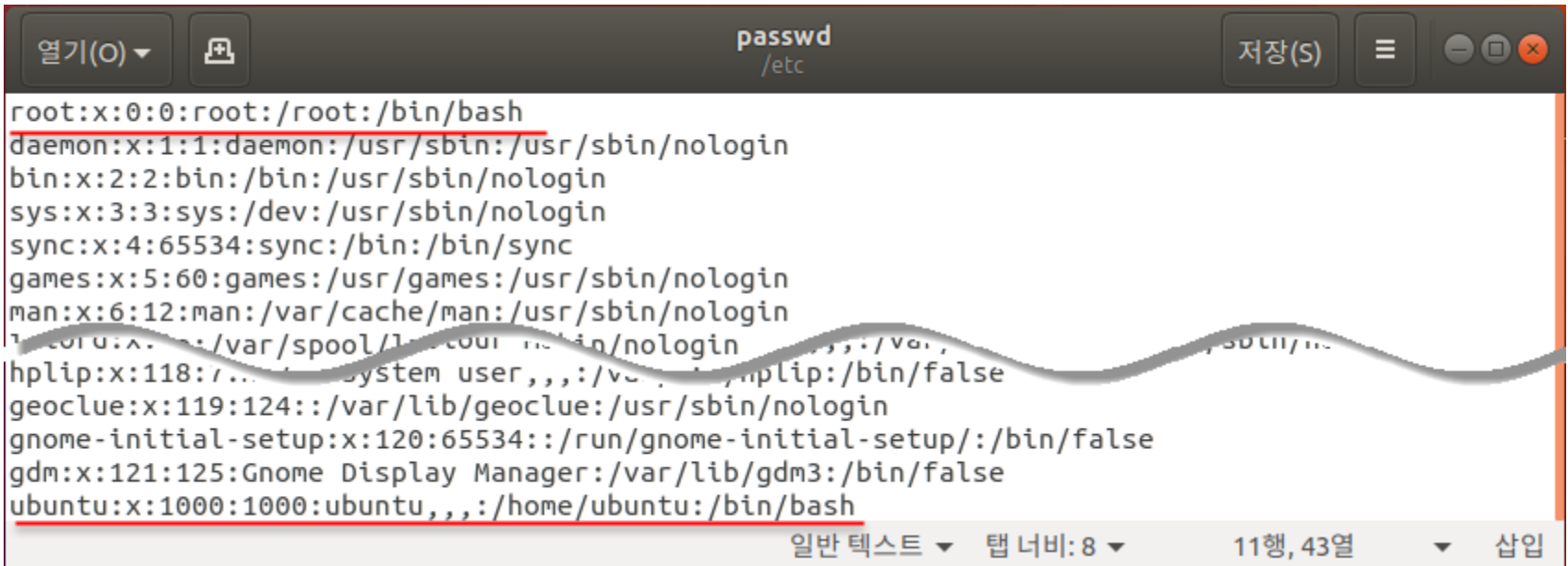
학습목표

- 리눅스의 사용자와 그룹에 대해 알아본다.
- 리눅스의 사용자와 그룹 관련 명령어 사용법을 익힌다.
- 파일의 유형을 파악하고 소유권과 허가권을 이해한다.
- 링크의 개념을 이해한다.
- 프로세스, 서비스(데몬)의 개념을 이해한다.

1-1 사용자와 그룹의 개념

■ 리눅스의 다중 사용자 시스템

- 리눅스 서버 1대에 여러 사용자가 동시에 접속
- 슈퍼 유저에게는 사용자 생성 권한을 포함해, 모든 작업을 실행할 수 있는 권한이 주어짐
- 모든 사용자는 혼자서 존재하는 것이 아니라 하나 이상의 그룹에 소속되어야 함



The screenshot shows a text editor window titled 'passwd /etc'. The window contains the following text:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
hplip:x:118:7:/var/lib/hplip:/bin/false
geoclue:x:119:124:geoclue:/usr/sbin/nologin
gnome-initial-setup:x:120:65534:/run/gnome-initial-setup:/bin/false
gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
ubuntu:x:1000:1000:ubuntu:/home/ubuntu:/bin/bash
```

The bottom status bar of the editor shows '일반 텍스트', '탭 너비: 8', '11행, 43열', and '삽입'.

- vi 에디터 또는 gedit로 /etc/passwd 파일 열기
- 각 행은 '사용자 이름:비밀번호:사용자 ID:사용자 소속 그룹 ID:추가 정보:홈 디렉터리: 기본 셸'을 의미

1-1 사용자와 그룹의 개념

■ ubuntu 사용자 살펴보기

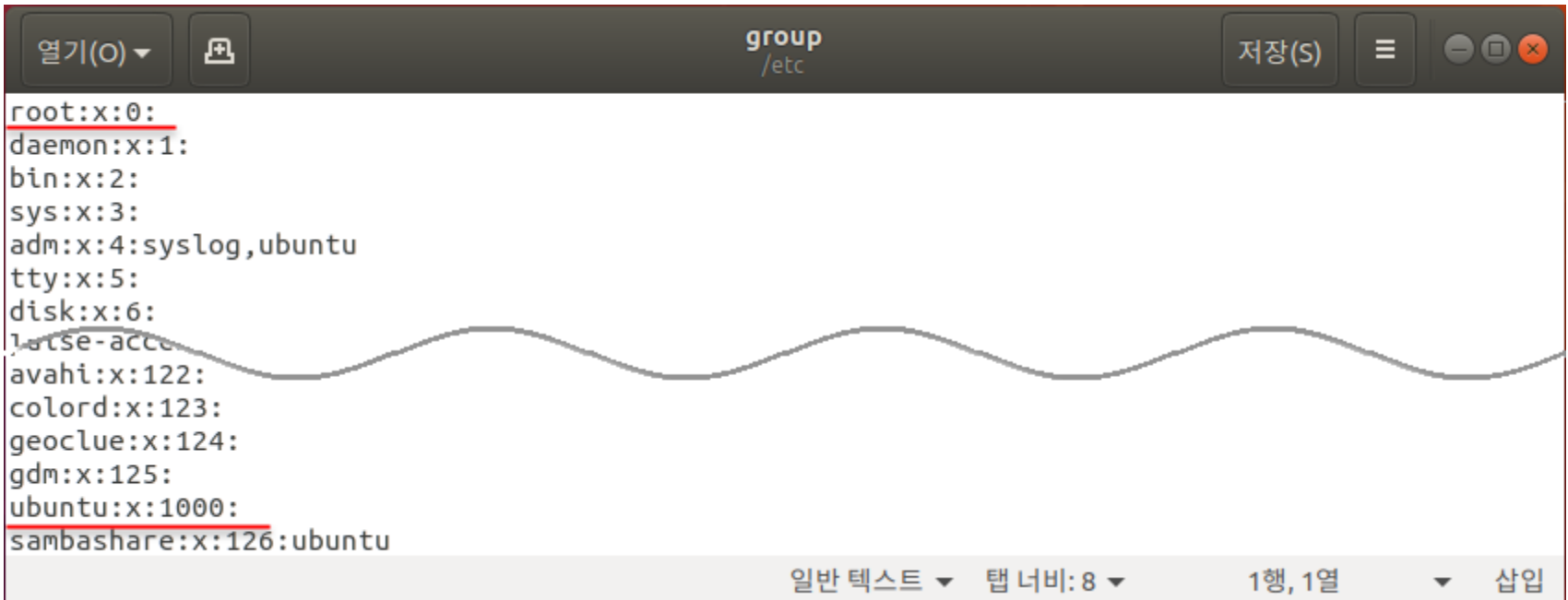
ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash

- 사용자 이름은 ubuntu, 비밀번호는 x
→ /etc/shadow 파일에 비밀번호가 지정되어 있다는 의미
- ubuntu의 사용자 ID는 1000, ubuntu가 속한 그룹의 ID도 1000
- 추가 정보로 '이름, 방 번호, 직장 전화번호, 집 전화번호, 기타'가 있으나 모두 생략됨
- 현재 사용자 이름만 ubuntu로 설정
- ubuntu 사용자의 홈 디렉터리는 /home/ubuntu이고, 로그인 시 제공되는 셸은 /bin/bash
- root 사용자를 살펴보면 사용자 ID와 소속 그룹 ID가 모두 0 으로 설정되어 있음

1-1 사용자와 그룹의 개념

■ /etc/group 파일 열어보기

- 각 행은 '그룹 이름:비밀번호:그룹 ID:보조 그룹 사용자'를 의미
- 보조 그룹 사용자는 이 그룹을 주 그룹이 아닌 보조 그룹으로 사용하는 사용자의 목록
- 여러 명이면 쉼표(,)로 구분



```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,ubuntu
tty:x:5:
disk:x:6:
lxdse-accu
avahi:x:122:
colord:x:123:
geoclue:x:124:
gdm:x:125:
ubuntu:x:1000:
sambashare:x:126:ubuntu
```

1-2 사용자와 그룹 관련 명령어

■ adduser

- 새로운 사용자를 추가하는 **adduser** 명령어를 실행하면 /etc/passwd, /etc/shadow, /etc/group 파일에 새로운 행이 추가됨

```
# adduser cookuser1      -- cookuser1 사용자 생성
  새 UNIX 암호 입력 :    -- 비밀번호 설정(입력하는 글자가 보이지 않음)
  새 UNIX 암호 재입력 :  -- 다시 입력
  이름 [ ] :            -- 전체 이름 입력(생략 가능)
  방 번호 [ ]           -- 사무실 호수 입력(생략 가능)
  직장 전화번호 [ ]     -- 직장 전화번호 입력(생략 가능)
  집 전화번호 [ ]      -- 집 전화번호 입력(생략 가능)
  기타 [ ]             -- 기타 내용 입력(생략 가능)
  정보가 올바릅니까? [Y/n] -- 확인(Enter)

# adduser --uid 1111 cookuser2 -- cookuser2 사용자를 생성하고 사용자 ID를 1111로 지정
# adduser --gid 1000 cookuser3 -- cookuser3 사용자를 생성하고 그룹 ID가 1000인 그룹에 포함
                                (실습에서 그룹 ID 1000은 ubuntu 그룹임)

# adduser --home /newhome cookuser4 -- cookuser4 사용자를 생성하고 홈 디렉터리를 /newhome으
                                로 지정

# adduser --shell /bin/csh cookuser5 -- cookuser5 사용자를 생성하고 기본 셸을 /bin/csh로 지정
```

1-2 사용자와 그룹 관련 명령어

■ passwd

- 사용자의 비밀번호를 변경하는 명령어

```
# passwd cookuser1 -- cookuser1 사용자의 비밀번호 설정 또는 변경
```

■ usermod

- 사용자의 속성을 변경하는 명령어

```
# usermod --shell /bin/csh cookuser1 -- cookuser1 사용자의 기본 셸을 /bin/chs로 변경  
# usermod --groups ubuntu cookuser1 -- cookuser1 사용자의 보조 그룹에 ubuntu 그룹 추가
```

■ userdel

- 사용자를 삭제하는 명령어

```
# userdel cookuser2 -- cookuser2 사용자 삭제  
# userdel -r cookuser3 -- cookuser3 사용자 삭제 및 홈 디렉터리 삭제
```


1-2 사용자와 그룹 관련 명령어

■ chage

- 사용자의 비밀번호를 주기적으로 변경하도록 설정하는 명령어

```
# chage -l cookuser1 -- cookuser1 사용자에게 설정된 내용 확인
# chage -m 2 cookuser1 -- cookuser1 사용자에게 설정한 비밀번호를 사용해야 하는 최소 일자(변경 후 최소 2일은 사용해야 함)
# chage -M 30 cookuser1 -- cookuser1 사용자에게 설정한 비밀번호를 사용할 수 있는 최대 일자(변경 후 최대 30일 사용 가능)
# chage -E 2025/12/12 cookuser1 -- cookuser1 사용자에게 설정한 비밀번호 만료일(2025년 12월 12일까지 사용 가능)
# chage -W 10 cookuser1 -- cookuser1 사용자에게 설정한 비밀번호 만료 전의 경고 기간(암호 만료일 10일 전부터 경고 메시지 출력). 지정하지 않으면 기본 값은 7일
```

■ groups

- 사용자가 소속된 그룹을 보여주는 명령어

```
# groups -- 현재 사용자가 소속된 그룹 출력
# groups cookuser1 -- cookuser1 사용자가 소속된 그룹 출력
```

1-2 사용자와 그룹 관련 명령어

■ groupadd

- 새로운 그룹을 생성하는 명령어

```
# groupadd newgroup1          -- newgroup1 그룹 생성
# groupadd --gid 2222 newgroup2 -- newgroup2 그룹을 생성하고 그룹 ID를 2222로 지정
```

■ groupmod

- 그룹의 속성을 변경하는 명령어

```
# groupmod --new-name mygroup1 newgroup1 -- newgroup1 그룹의 이름을 mygroup1로 변경
```

■ groupdel

- 그룹을 삭제하는 명령어

```
# groupdel newgroup2 -- newgroup2 그룹 삭제(newgroup2 그룹을 주 그룹으로 지정한 사용자가 없어야 함)
```

■ gpasswd

- 그룹의 비밀번호를 설정하거나 그룹 관리를 수행하는 명령어

```
# gpasswd mygroup1          -- mygroup1 그룹 비밀번호 설정
# gpasswd -A cookuser1 mygroup1 -- cookuser1 사용자를 mygroup1 그룹 관리자로 지정
# gpasswd -a cookuser4 mygroup1 -- cookuser4 사용자를 mygroup1 그룹에 추가
# gpasswd -d cookuser4 mygroup1 -- cookuser4 사용자를 mygroup1 그룹에서 제거
```

1. Server 초기화하기

1-1 Server를 처음 설치 상태로 초기화

VMware 종료 후 C:\Linux\Server 폴더 삭제

C:\Linux(백업)\Server 폴더를 C:\Linux\ 폴더에 통째로 복사

1-2 Server를 부팅하면 root 사용자로 자동 접속됨

1-3 바탕화면에서 마우스 오른쪽 버튼을 클릭하여 [터미널 열기] 선택

2. 새로운 사용자 만들기

2-1 **adduser hanbit1** 명령을 입력하여 hanbit1 사용자 생성
비밀번호를 '**1234**'로 설정하고 나머지는 계속 Enter 누름

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# adduser hanbit1
'hanbit1' 사용자를 추가 중...
새 그룹 'hanbit1' (1001) 추가 ...
새 사용자 'hanbit1' (1001) 을(를) 그룹 'hanbit1' (으)로 추가 ...
'/home/hanbit1' 홈 디렉터리를 생성하는 중...
'/etc/skel'에서 파일들을 복사하는 중...
새 UNIX 암호 입력:
새 UNIX 암호 재입력:
passwd: 암호를 성공적으로 업데이트했습니다
hanbit1의 사용자의 정보를 바꿉니다
새로운 값을 넣거나, 기본값을 원하시면 엔터를 치세요
이름 []:
방 번호 []:
직장 전화번호 []:
집 전화번호 []:
기타 []:
정보가 올바릅니까? [Y/n]
root@server:~#
```

2-2 hanbit1 사용자가 추가되었는지 **tail /etc/passwd** 명령으로 확인

tail은 끝부분의 10행을 보여주는 명령어, 마지막 행을 보면 사용자가 추가된 것 확인 가능

사용자 이름은 앞에서 지정한 hanbit1, 비밀번호는 /etc/shadow 파일에 지정되어 있음

세 번째 열을 보면 hanbit1 사용자의 ID가 1001 → 그 앞에 있는 ubuntu 사용자의 ID

1000 다음에 자동으로 1을 더하여 할당한 것

그룹 ID도 1001로 지정됨

사용자의 홈 디렉터리는 기본 설정인 '/home/사용자명', 셸은 기본 설정인 /bin/bash

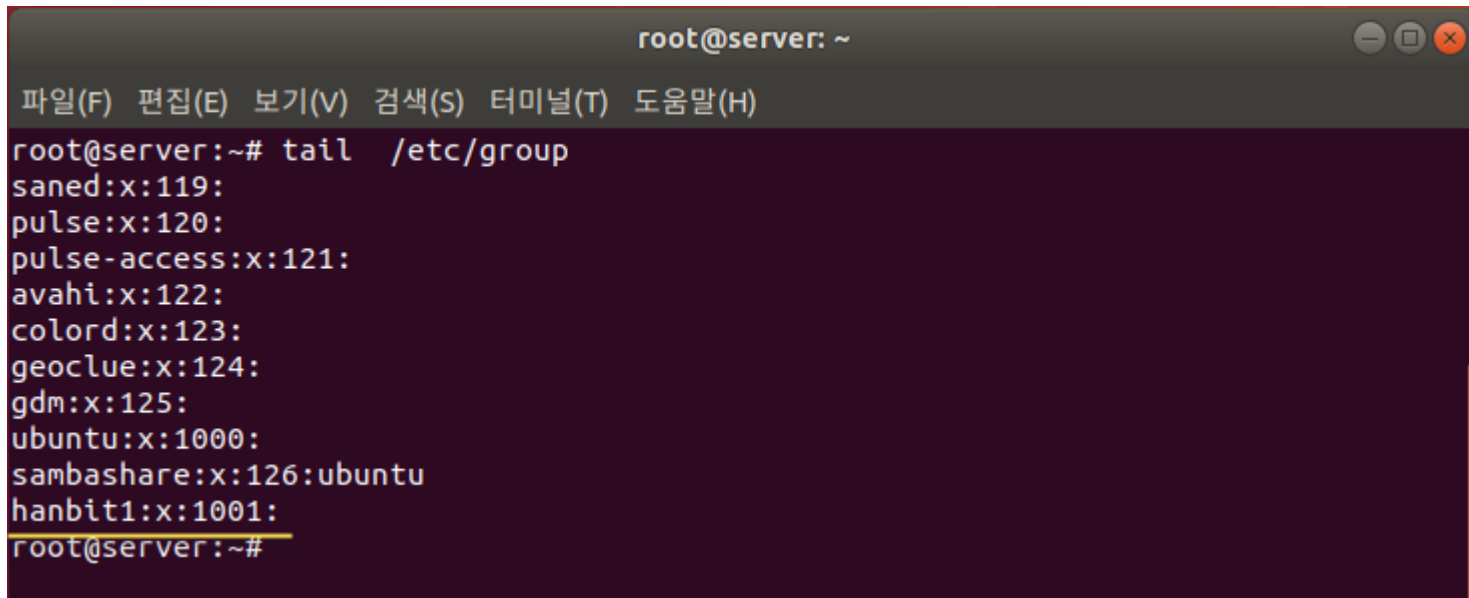
```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# tail /etc/passwd
saned:x:114:119::/var/lib/saned:/usr/sbin/nologin
pulse:x:115:120:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
avahi:x:116:122:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
colord:x:117:123:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
hplip:x:118:7:HPLIP system user,,,:/var/run/hplip:/bin/false
geoclue:x:119:124::/var/lib/geoclue:/usr/sbin/nologin
gnome-initial-setup:x:120:65534::/run/gnome-initial-setup:/bin/false
gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
hanbit1:x:1001:1001,,,:/home/hanbit1:/bin/bash
root@server:~#
```

2-3 **tail /etc/group** 명령을 입력하여 그룹 확인

그룹 이름이 사용자 이름과 동일한 hanbit1, 그룹 ID는 ubuntu의 그룹 ID인 1000에 1을 더한 1001로 생성

adduser 명령 실행 → 별도로 그룹을 지정하지 않으면 자동으로 사용자 이름과 동일한 그룹이 생성됨, 새로운 사용자는 생성된 그룹에 자동으로 포함

새로 생성된 그룹(hanbit1)은 소속된 사용자가 1명



```
root@server: ~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
root@server:~# tail /etc/group  
saned:x:119:  
pulse:x:120:  
pulse-access:x:121:  
avahi:x:122:  
colord:x:123:  
geoclue:x:124:  
gdm:x:125:  
ubuntu:x:1000:  
smbashare:x:126:ubuntu  
hanbit1:x:1001:  
root@server:~#
```

2-4 많은 사용자를 관리할 땐 사용자 이름과 그룹 이름이 같아 관리 불편

그룹을 먼저 만든 후 사용자를 그 그룹에 넣는 것이 좋음

3. 그룹을 생성하고 소속된 다수 사용자 관리하기

3-1 **userdel -r hanbit1** 명령으로 hanbit1 사용자 삭제

groupadd ubuntuGroup 명령으로 ubuntuGroup 그룹을 만든 후,

tail -5 /etc/group 명령으로 확인

```
root@server: ~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
root@server:~# userdel -r hanbit1  
userdel: hanbit1 mail spool (/var/mail/hanbit1) not found  
root@server:~# groupadd ubuntuGroup  
root@server:~# tail -5 /etc/group  
geoclue:x:124:  
gdm:x:125:  
ubuntu:x:1000:  
sambashare:x:126:ubuntu  
ubuntuGroup:x:1001:  
root@server:~#
```

- 3-2 **adduser --gid 그룹ID 사용자** 명령으로 새로운 hanbit1, hanbit2 사용자 생성,
ubuntuGroup 그룹(그룹 ID는 1001)으로 지정
둘 다 암호를 '**1234**'로 설정하고 나머지는 모두 Enter 누름

```
root@server: ~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
root@server:~# adduser --gid 1001 hanbit1  
'hanbit1' 사용자를 추가 중...  
새 사용자 'hanbit1' (1001) 을(를) 그룹 'ubuntuGroup' (으)로 추가 ...  
'/home/hanbit1' 홈 디렉터리를 생성하는 중...  
'/etc/passwd'에서 파일들을 복사하는 중...  
접근 권한 부여 []:  
기타 []:  
정보가 올바릅니까? [Y/n]  
root@server:~# adduser --gid 1001 hanbit2  
'hanbit2' 사용자를 추가 중...  
새 사용자 'hanbit2' (1002) 을(를) 그룹 'ubuntuGroup' (으)로 추가 ...  
'/home/hanbit2' 홈 디렉터리를 생성하는 중...  
'/etc/passwd'에서 파일들을 복사하는 중...  
접근 권한 부여 []:  
기타 []:  
정보가 올바릅니까? [Y/n]  
root@server:~#
```


3-3 **tail -5 /etc/passwd** 명령으로 passwd 파일 확인

그룹 ID는 모두 1001

앞에서 /etc/group 파일의 1001이 ubuntuGroup임을 확인

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# tail -5 /etc/passwd
gnome-initial-setup:x:120:65534:./run/gnome-initial-setup:/bin/false
gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
hanbit1:x:1001:1001:,,,:/home/hanbit1:/bin/bash
hanbit2:x:1002:1001:,,,:/home/hanbit2:/bin/bash
root@server:~#
```

3-4 비밀번호가 설정된 파일을 **tail -5 /etc/shadow** 명령으로 확인

hanbit1, hanbit2 사용자에게 비밀번호가 설정되어 있음 확인 가능

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# tail -5 /etc/shadow
gnome-initial-setup*:17937:0:99999:7:::
gdm*:17937:0:99999:7:::
ubuntu:$6$6SZLoYwD$YXGDswe6tgnWIh1vTEgiicjj2bgKqc0rUjrp4eH8wHjFF0ErJ0sL.YeSNq.pW
MsgwTdgmIo6eA4poVj5s4Fcb.:18003:0:99999:7:::
hanbit1:$6$vINL76yl$LfSuNmfc8GU3G8i0q51xaLNypRXnQEw0k8E/5EwtZ2fhaz0miQRlgajH2wWI
9vmRlc3drSVR8u6Xhe7SfA5120:18076:0:99999:7:::
hanbit2:$6$0lgU8Tor$hlztc7fqiyIhYphD3g09Vv95mSEh7603kMSJvM5IWHwHLWcyyuSeknPRsAvD
s7t2ufEp8woSLR.qkYnwrHnLj1:18076:0:99999:7:::
root@server:~#
```

3-5 **ls -l** 디렉터리 명령

hanbit1 사용자의 홈 디렉터리인 /home/hanbit1과 /etc/skel 디렉터리 비교

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# ls -l /home/hanbit1
합계 12
-rw-r--r-- 1 hanbit1 ubuntuGroup 8980  6월 29 18:49 examples.desktop
root@server:~# ls -l /etc/skel
합계 12
-rw-r--r-- 1 root root 8980  4월 16 2018 examples.desktop
root@server:~#
```

4. X 윈도우 환경에서 사용자 관리하기

4-1 바탕화면에서 오른쪽 위의 ▼ 아이콘에 이어 * 아이콘 클릭

4-2 왼쪽에서 아래로 스크롤하여 [자세히 보기] 선택 후,
[사용자]에 이어 <사용자 추가> 클릭, 사용자 추가

The image displays two screenshots of a user management interface in an X Window environment.

Left Screenshot (자세히 보기 - Details View):

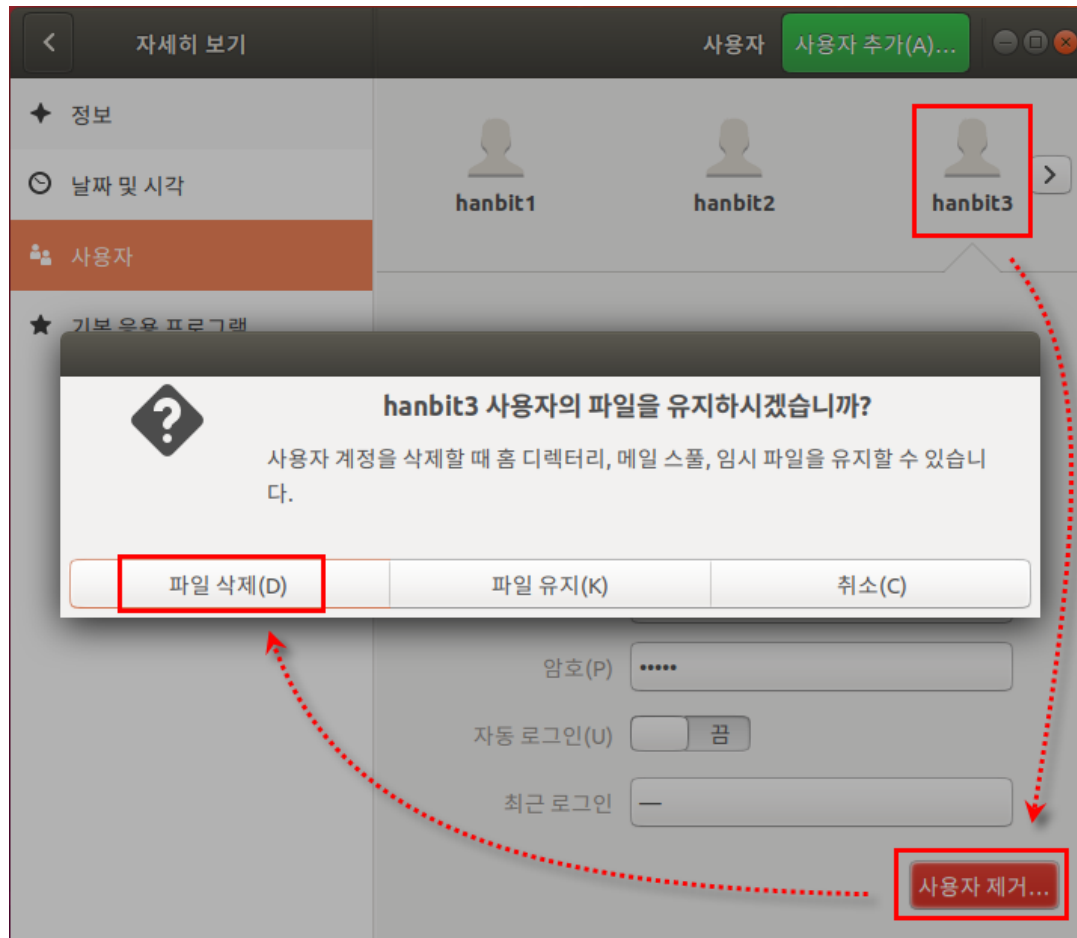
- Top bar: < (Back), 자세히 보기 (Details View), 사용자 (Users), 사용자 추가(A)... (Add User...).
- Left sidebar: 정보 (Info), 날짜 및 시각 (Date and Time), **사용자 (Users)** (highlighted), 기본 응용 프로그램 (Default Applications).
- Main area: Shows three users: hanbit1, hanbit2, and hanbit3. Below them is a detailed view for 'hanbit3' with fields for: 계정 종류(T) (Account Type: 표준 (Standard) / 관리자 (Administrator)), 언어(L) (Language: 한국어 (Korean)), 암호(P) (Password: masked), 자동 로그인(U) (Auto Login: off), and 최근 로그인 (Last Login: —).
- Bottom right: 사용자 제거... (Remove User...).

Right Screenshot (사용자 추가(A)... - Add User Dialog):

- Top bar: 취소(C) (Cancel), 사용자 추가 (Add User), **추가(A)** (Add) (highlighted).
- Form fields: 계정 종류(T) (Account Type: 표준 (Standard) / 관리자 (Administrator)), 전체 이름(F) (Full Name: hanbit3), 사용자 이름(U) (Username: hanbit3), and a note: 홈 폴더의 이름입니다. 바꿀 수 없습니다. (This is the name of the home folder. It cannot be changed.).
- 암호 (Password) section: ☐ 다음 로그인에 암호를 설정하도록 허용(L) (Allow setting password for next login), ☒ **지금 암호 설정(N)** (Set password now). The password field shows masked characters and a strength indicator. A note below says: 영문자, 숫자, 문장 기호를 더 쓰면 더 강력합니다. (The more you use uppercase letters, numbers, and punctuation, the stronger it is.).
- Bottom: 확인(C) (Confirm) button.

Red dashed arrows indicate the workflow: from the '사용자' tab in the left window to the '사용자 추가(A)...' button, then to the '전체 이름(F)' and '사용자 이름(U)' fields, then to the '암호' section, and finally to the '추가(A)' button in the right window.

4-3 사용자를 삭제하려면 위쪽에서 사용자를 선택, <사용자 제거> 클릭
사용자 홈 디렉터리가 필요 없다면 <파일 삭제> 클릭

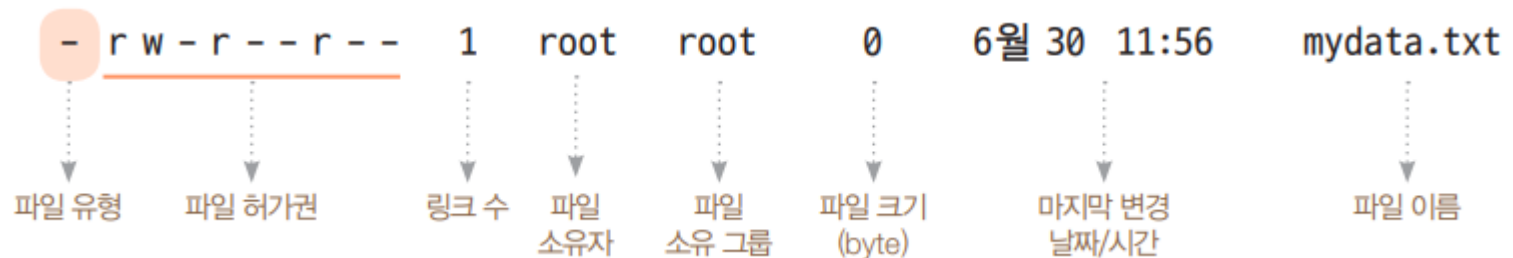


4-4 오른쪽 위의 X를 클릭, 설정 창 닫기

2-1 파일 유형

- 소유권(ownership)과 허가권(permission)
 - root 사용자가 자신의 홈 디렉터리에서 **touch mydata.txt** 명령으로 빈 파일을 만들고 **ls -l** 명령을 실행하면 다음과 같이 나타남

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# touch mydata.txt
root@server:~# ls -l mydata.txt
-rw-r--r-- 1 root root 0  6월 30 11:56 mydata.txt
root@server:~#
```



2-2 파일 허가권

- 파일 소유권(ownership)과 허가권(permission)
 - 파일 허가권
 - 파일 허가권은 'rw-', 'r--', 'r--'와 같이 3개씩 끊어서 구분
 - r은 read, w는 write, x는 execute의 약자
 - 'rw-'는 읽거나 쓸 수 있지만 실행할 수는 없다는 의미
 - 읽고 쓰고 실행할 수 있는 파일은 'rwx'로 표시

소유자			그룹			그 외 사용자		
r	w	-	r	-	-	r	-	-
4	2	0	4	0	0	4	0	0
6			4			4		

2-2 파일 허가권

소유자			그룹			그 외 사용자		
r	w	-	r	-	-	r	-	-
4	2	0	4	0	0	4	0	0
6			4			4		

- 첫 번째의 'rw-'는 소유자(user)의 파일 접근 권한을, 두 번째의 'r--'는 그룹(group)의 파일 접근 권한을, 세 번째의 'r--'는 그 외 사용자(other)의 파일 접근 권한을 의미
- 소유자의 허가권을 나타내는 6은 2진수로 110이므로 'rw-'로, 그룹의 허가권을 나타내는 4는 2진수로 100이므로 'r--'로, 그 외 사용자의 허가권을 나타내는 4도 2진수로 100이므로 'r--'로 표현
- 디렉터리(폴더)를 해당 디렉터리로 이동하려면 실행(x) 권한이 반드시 있어야 함
- 따라서 일반적으로 디렉터리에는 소유자, 그룹, 그 외 사용자의 실행(x) 권한이 설정됨

2-2 파일 허가권

- chmod
 - 파일 허가권을 변경하는 명령어
 - root 사용자 또는 해당 파일의 소유자만 실행 가능
 - **chmod 777 mydata.txt** 명령을 실행하면 모든 사용자가 mydata.txt 파일은 읽고 쓰고 실행할 수 있음
 - 상대 모드(symbolic method)로도 사용
 - **chmod u+x 파일명** 명령은 소유자(User)에게 실행(eXecute) 권한을 허가하라(+)는 의미

2-3 파일 소유권

■ 파일 소유권 개념

- 파일 소유권은 특정 사용자와 그룹이 파일에 대한 소유 권한을 가지는 것을 의미
- Mydata.txt 파일의 경우, 소유자가 root 사용자이고 소유 그룹도 root

■ chown

- 파일 소유권을 변경하는 명령어
- 'chown 새사용자명(.새그룹명) 파일명'과 같은 형식으로 사용
- **chown ubuntu mydata.txt** 명령은 mydata.txt 파일의 소유자를 ubuntu 사용자로 바꾸라는 의미
- **chown ubuntu.ubuntu mydata.txt** 명령은 파일 그룹도 ubuntu 그룹으로 바꾸라는 의미
- **chgrp ubuntu mydata.txt** 명령은 그룹만 ubuntu 그룹으로 바꾸라는 의미

[실습 5-2] 파일 허가권과 소유권 실행하기

교재 187~190p 참고

1. 파일 속성 확인하기

1-1 Server 실행, root 사용자로 접속한 후 터미널 열기, 연습용 파일 하나 생성

vi sample 명령을 입력하고 대문자 I를 누른 후 다음 내용 입력

':wq'를 입력하여 저장 및 종료

(한글/영문 전환 키는 Shift + Space bar)

```
북북 리눅스를 학습 중입니다.
```

```
ls /var
```

1-2 **ls -l sample** 명령을 입력하여 파일 속성 확인

파일 허가권이 'rw-r--r--'로 설정되어 있고 소유자는 root 사용자, 그룹은 root 그룹

```
root@server: ~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
root@server:~# ls -l sample  
-rw-r--r-- 1 root root 49  6월 30 12:40 sample  
root@server:~#
```

[실습 5-2] 파일 허가권과 소유권 실행하기

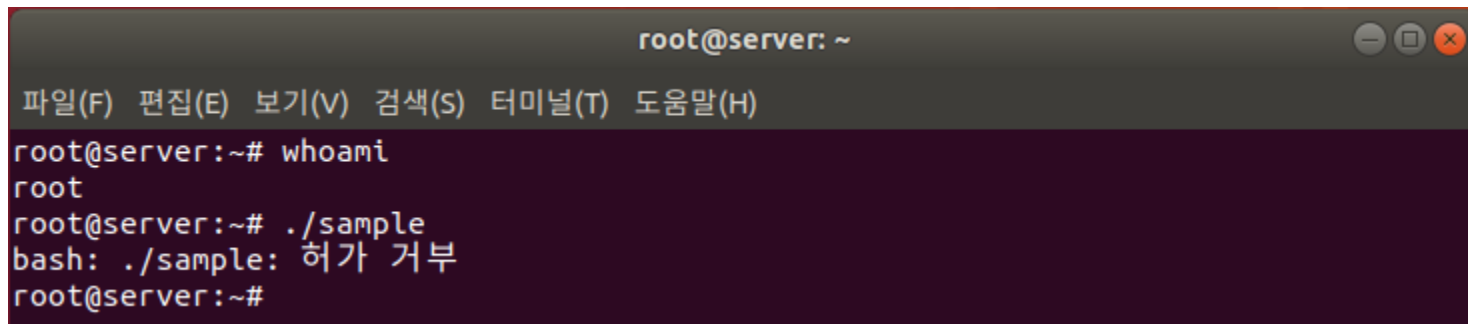
교재 187~190p 참고

2. 파일 속성 변경하기

- 2-1 **./sample** 명령을 입력하여 sample 파일 실행
허가가 거부되었다는 메시지가 나타남

whoami -- 현재 사용자를 알려줌

./sample -- 현재 디렉터리의 sample 파일 실행(./는 현재 디렉터리에 있는 파일을 의미)



```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# whoami
root
root@server:~# ./sample
bash: ./sample: 허가 거부
root@server:~#
```

[실습 5-2] 파일 허가권과 소유권 실행하기

교재 187~190p 참고

2-2 **chmod 755 sample** 명령 입력

ls -l sample 명령으로 sample 파일에 변경 사항이 있는지 확인
그리고 **./sample** 명령을 입력하여 다시 파일 실행

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# chmod 755 sample
root@server:~# ls -l sample
-rwxr-xr-x 1 root root 49 6월 30 12:40 sample
root@server:~# ./sample
./sample: 줄 1: 쿼크: 명령어를 찾을 수 없음
backups crash local log metrics run spool
cache lib lock mail opt snap tmp
root@server:~#
```

3. 소유권 변경하기

3-1 **chown ubuntu sample** 명령 입력, sample 파일의 소유권을 ubuntu 사용자로 변경

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# ls -l sample
-rwxr-xr-x 1 root root 49  6월 30 12:40 sample
root@server:~# chown ubuntu sample
root@server:~# ls -l sample
-rwxr-xr-x 1 ubuntu root 49  6월 30 12:40 sample
root@server:~#
```

3-2 **chown ubuntu sample** 명령 입력, 그룹을 ubuntu 그룹으로 변경

사용자와 그룹을 한꺼번에 바꾸려면 **chown ubuntu.ubuntu sample** 명령 실행

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# chgrp ubuntu sample
root@server:~# ls -l sample
-rwxr-xr-x 1 ubuntu ubuntu 49  6월 30 12:40 sample
root@server:~#
```

[실습 5-2] 파일 허가권과 소유권 실행하기

교재 187~190p 참고

3-3 ubuntu 사용자로 접속,

sample 파일의 속성을 모두가 읽고 쓰고 실행(777)할 수 있도록 변경

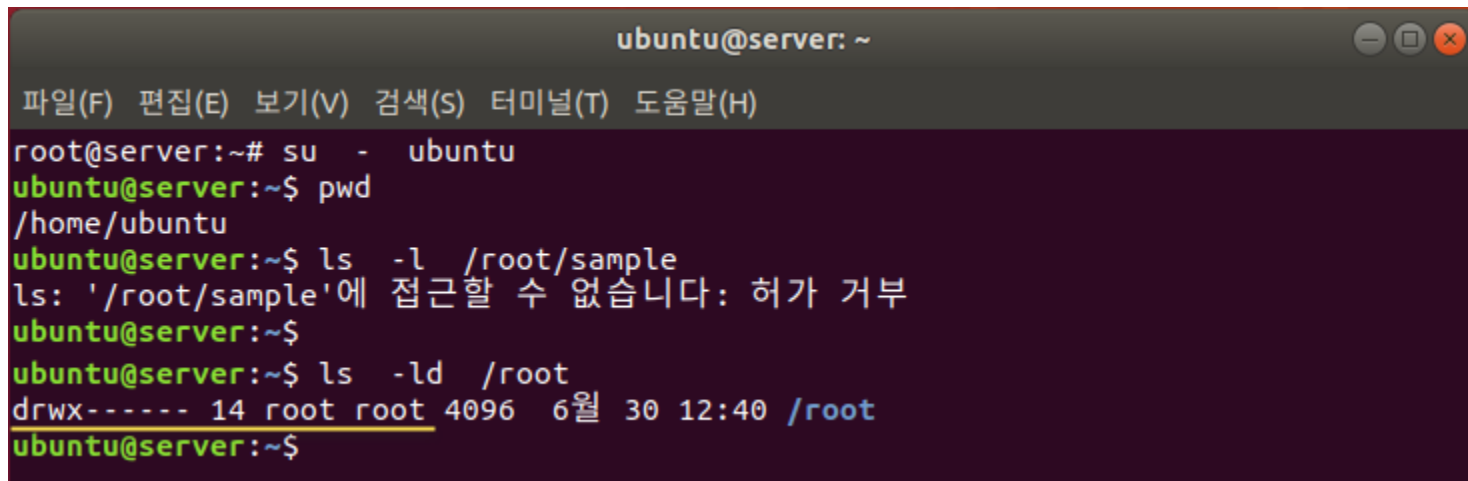
그런데 앞에서 sample 파일의 소유권을 ubuntu에게 확실히 넘겨주었는데도

ubuntu 사용자의 /root/sample 파일 접근이 거부됨

→ /root 디렉터리의 속성이 'rwx-----'이기 때문

그 외 사용자의 허가권이 '---'이므로, 읽기, 쓰기, 실행이 불가능해 ubuntu 사용자의 /root 디렉터리 접근이 거부된 것

```
su - ubuntu -- ubuntu 사용자로 접속(root가 접속하는 경우에는 비밀번호를 물어보지 않음)
pwd          -- 현재 디렉터리 확인. ubuntu 사용자의 홈 디렉터리가 나옴
ls -l /root/sample
ls -ld /root -- /root 디렉터리의 속성 확인
```



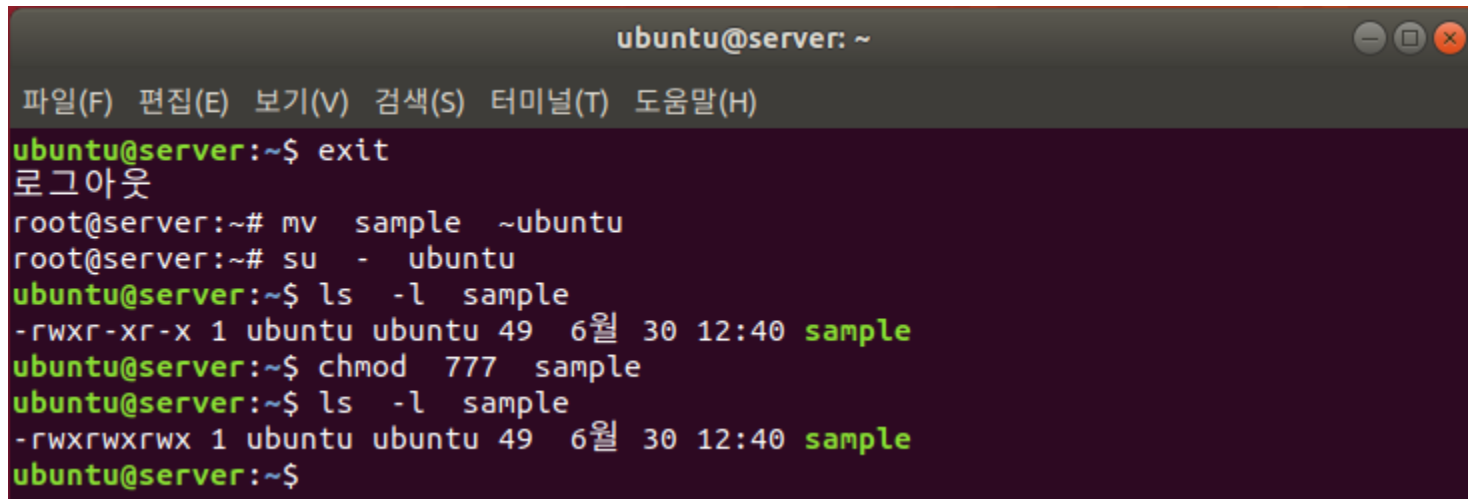
```
ubuntu@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# su - ubuntu
ubuntu@server:~$ pwd
/home/ubuntu
ubuntu@server:~$ ls -l /root/sample
ls: '/root/sample'에 접근할 수 없습니다: 허가 거부
ubuntu@server:~$
ubuntu@server:~$ ls -ld /root
drwx----- 14 root root 4096  6월 30 12:40 /root
ubuntu@server:~$
```

[실습 5-2] 파일 허가권과 소유권 실행하기

교재 187~190p 참고

3-4 root 사용자로 접속, sample 파일을 /home/ubuntu 디렉터리로 옮긴 후 허가권 변경

```
exit          -- 원래 접속한 사용자(여기서는 root)로 돌아감
mv sample ~ubuntu -- ~ubuntu는 ubuntu 사용자의 홈 디렉터리(/home/ubuntu와 동일)
su - ubuntu
ls -l sample
chmod 777 sample
ls -l sample
```



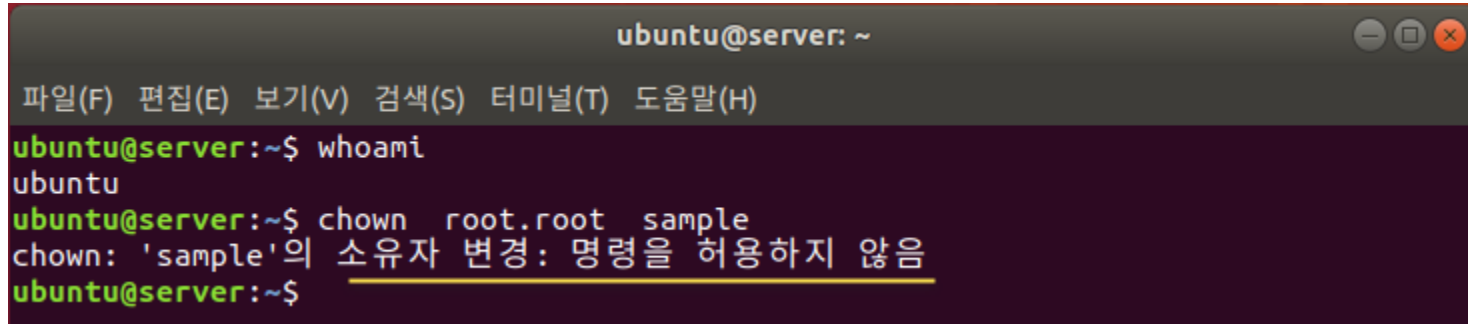
```
ubuntu@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
ubuntu@server:~$ exit
로그아웃
root@server:~# mv sample ~ubuntu
root@server:~# su - ubuntu
ubuntu@server:~$ ls -l sample
-rwxr-xr-x 1 ubuntu ubuntu 49  6월 30 12:40 sample
ubuntu@server:~$ chmod 777 sample
ubuntu@server:~$ ls -l sample
-rwxrwxrwx 1 ubuntu ubuntu 49  6월 30 12:40 sample
ubuntu@server:~$
```

[실습 5-2] 파일 허가권과 소유권 실행하기

교재 187~190p 참고

3-5 **chown root.root sample** 명령 입력

‘명령을 허용하지 않음’이라는 메시지가 나타나는 것이 정상

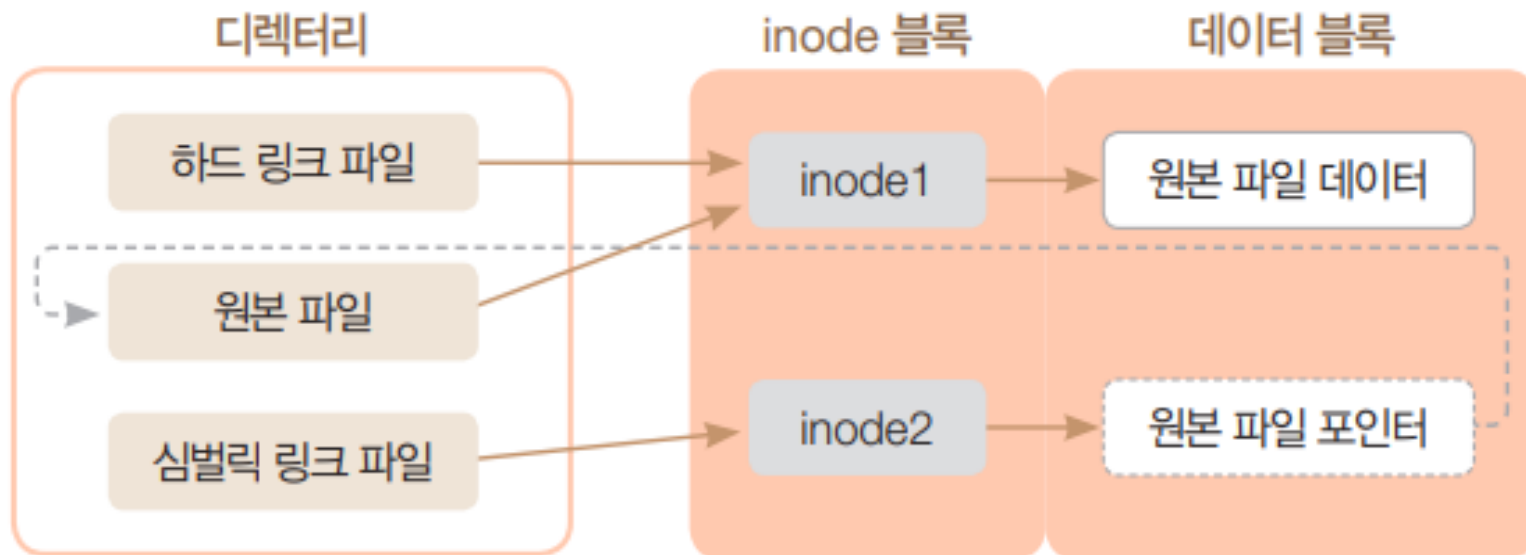
A terminal window titled 'ubuntu@server: ~' with standard Ubuntu window controls. The menu bar shows '파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)'. The command history shows: 'ubuntu@server:~\$ whoami' followed by 'ubuntu', then 'ubuntu@server:~\$ chown root.root sample'. The output is 'chown: 'sample'의 소유자 변경: 명령을 허용하지 않음' (chown: 'sample' owner change: command not allowed), which is underlined. The prompt returns to 'ubuntu@server:~\$'.

```
ubuntu@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
ubuntu@server:~$ whoami
ubuntu
ubuntu@server:~$ chown root.root sample
chown: 'sample'의 소유자 변경: 명령을 허용하지 않음
ubuntu@server:~$
```

3-6 **exit** 명령 입력, root 사용자로 돌아옴

2-4 링크

- 하드 링크(hard link)와 심벌릭 링크(symbolic link)
 - 파일의 링크는 하드 링크(hard link)와 심벌릭 링크(symbolic link) 또는 소프트 링크(soft link)로 구분됨
 - 하드 링크를 생성하는 명령은 **ln 원본파일 링크파일명**
 - 심벌릭 링크를 생성하는 명령은 **ln -s 원본파일 링크파일명**

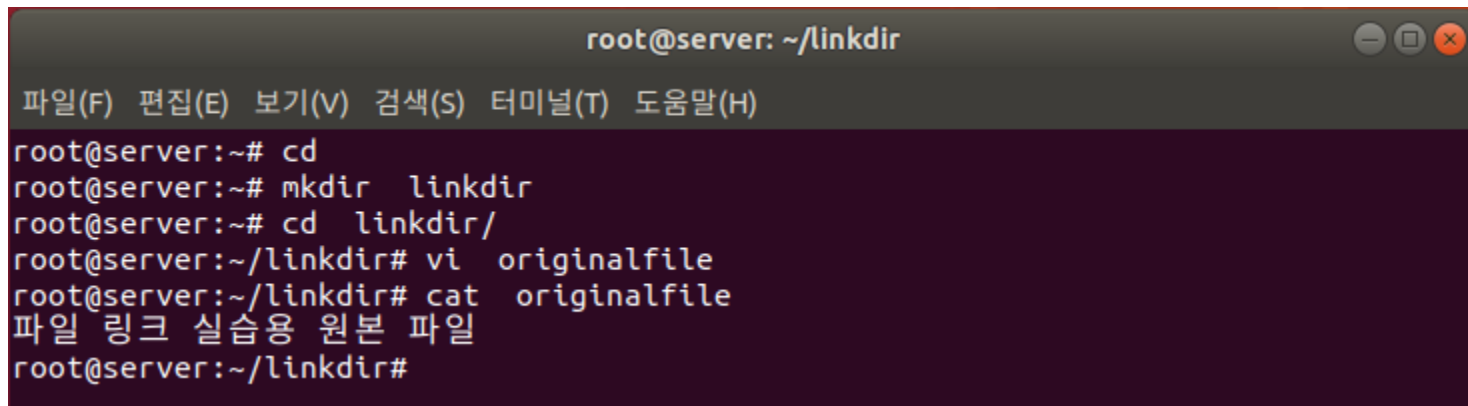


[실습 5-3] 하드 링크와 심벌릭 링크 생성하기

교재 191~193p 참고

1. 파일 생성하고 확인하기

- 1-1 Server 실행, /root/linkdir/ 디렉터리 생성 후, 그 안에 originalfile 파일 생성
vi 에디터나 gedit를 이용하여 '파일 링크 실습용 원본 파일' 입력 후 저장
cat 명령으로 파일 내용 확인



```
root@server: ~/linkdir
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# cd
root@server:~# mkdir linkdir
root@server:~# cd linkdir/
root@server:~/linkdir# vi originalfile
root@server:~/linkdir# cat originalfile
파일 링크 실습용 원본 파일
root@server:~/linkdir#
```

[실습 5-3] 하드 링크와 심벌릭 링크 생성하기

교재 191~193p 참고

2. 하드 링크와 심벌릭 링크 확인하기

2-1 **ln** 명령어와 옵션을 조합하여 하드 링크와 심벌릭 링크 파일 생성

```
ln originalfile hardlink    -- 하드 링크 생성
ln -s originalfile softlink -- 심벌릭 링크(소프트 링크) 생성
ls -il                      -- -il 옵션은 inode 번호를 맨 앞에 출력
cat hardlink                -- 하드 링크의 내용 확인
cat softlink                -- 심벌릭 링크의 내용 확인
```

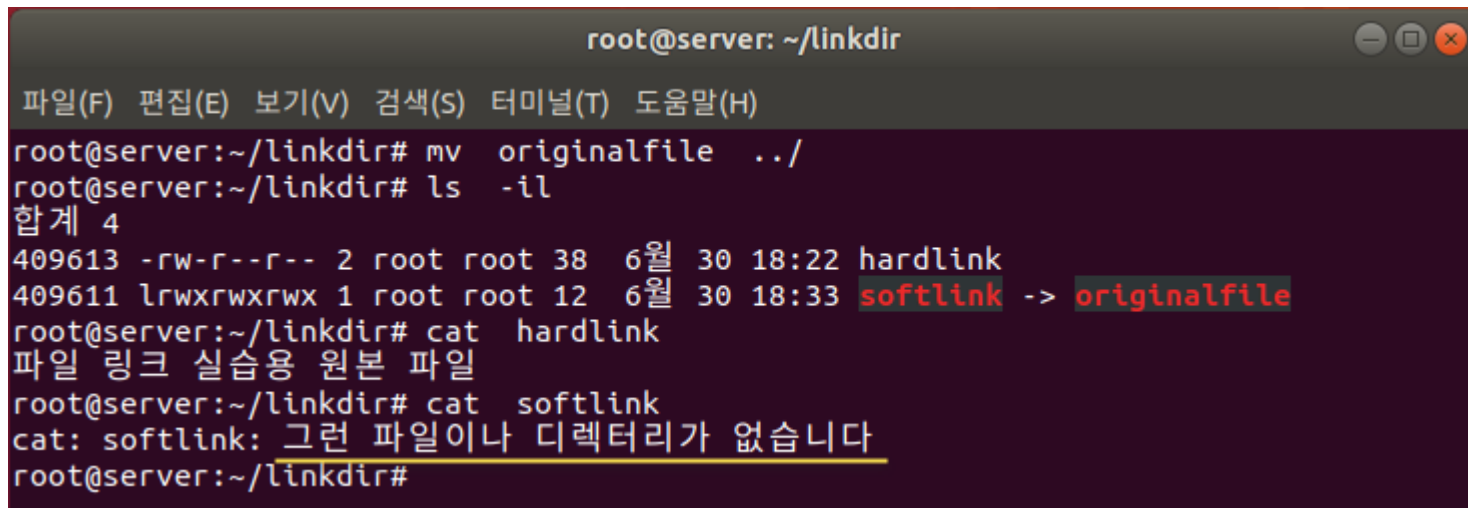
```
root@server: ~/linkdir
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~/linkdir# ln originalfile hardlink
root@server:~/linkdir# ln -s originalfile softlink
root@server:~/linkdir# ls -il
합계 8
409613 -rw-r--r-- 2 root root 38 6월 30 18:22 hardlink
409613 -rw-r--r-- 2 root root 38 6월 30 18:22 originalfile
409611 lrwxrwxrwx 1 root root 12 6월 30 18:33 softlink -> originalfile
root@server:~/linkdir# cat hardlink
파일 링크 실습용 원본 파일
root@server:~/linkdir# cat softlink
파일 링크 실습용 원본 파일
root@server:~/linkdir#
```

[실습 5-3] 하드 링크와 심벌릭 링크 생성하기

교재 191~193p 참고

2-2 원본 파일을 다른 곳으로 이동하고 하드 링크 파일과 심벌릭 링크 파일 확인

```
mv originalfile ../ -- 원본 파일을 상위 디렉터리(..)로 이동
ls -il
cat hardlink
cat softlink
```



```
root@server: ~/linkdir
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~/linkdir# mv originalfile ../
root@server:~/linkdir# ls -il
합계 4
409613 -rw-r--r-- 2 root root 38 6월 30 18:22 hardlink
409611 lrwxrwxrwx 1 root root 12 6월 30 18:33 softlink -> originalfile
root@server:~/linkdir# cat hardlink
파일 링크 실습용 원본 파일
root@server:~/linkdir# cat softlink
cat: softlink: 그런 파일이나 디렉터리가 없습니다
root@server:~/linkdir#
```

[실습 5-3] 하드 링크와 심벌릭 링크 생성하기

교재 191~193p 참고

2-3 원본 파일을 현재 디렉터리에 다시 가져오면 심벌릭 링크가 원상태로 복구됨

```
root@server: ~/linkdir
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~/linkdir# mv ../originalfile .
root@server:~/linkdir# ls -il
합계 8
409613 -rw-r--r-- 2 root root 38 6월 30 18:22 hardlink
409613 -rw-r--r-- 2 root root 38 6월 30 18:22 originalfile
409611 lrwxrwxrwx 1 root root 12 6월 30 18:33 softlink -> originalfile
root@server:~/linkdir# cat hardlink
파일 링크 실습용 원본 파일
root@server:~/linkdir# cat softlink
파일 링크 실습용 원본 파일
root@server:~/linkdir#
```

3-1 프로세스

■ 프로세스 정의

- 하드디스크에 저장된 실행 코드(프로그램)가 메모리에 로딩되어 활성화된 것
- 예를 들어 웹 브라우저 프로그램인 파이어폭스는 하드디스크의 어딘가에 저장되어 있는데 이렇게 하드디스크에 저장된 파일을 '프로그램'이라 부름
- 파이어폭스를 실행하여 화면에 나타난 상태(메모리에 로딩되어 화면에 나타난 상태가 더 정확한 표현)를 '프로세스'라 부름

■ 포그라운드 프로세스

- 포그라운드 프로세스(foreground process)는 파이어폭스와 마찬가지로 실행하면 화면에 나타나서 사용자와 상호 작용을 하는 프로세스

■ 백그라운드 프로세스

- 백그라운드 프로세스(background process)는 화면에 나타나지 않은 채 뒤에서 실행되는 프로세스

3-1 프로세스

- 프로세스 번호
 - 각 프로세스에 할당된 고유 번호
 - 메모리에 로딩되어 활성화된 프로세스를 구분하려면 고유 번호가 필요함
- 작업 번호
 - 현재 실행 중인 백그라운드 프로세스의 순차 번호
- 부모 프로세스와 자식 프로세스
 - 모든 프로세스는 독립적으로 실행되는 것이 아니라 부모 프로세스에 종속되어 실행
 - 예를 들어, 파이어폭스는 X 윈도우 프로세스가 구동된 상태에서 실행되어야 함
 - X 윈도우 : 파이어폭스의 부모 프로세스, 파이어폭스 : X 윈도우의 자식 프로세스
 - 만약 파이어폭스는 그대로 두고 X 윈도우만 강제 종료하면 파이어폭스도 종료됨
 - 부모 프로세스를 종료하면 부모 프로세스에 종속된 자식 프로세스도 종료되는 것

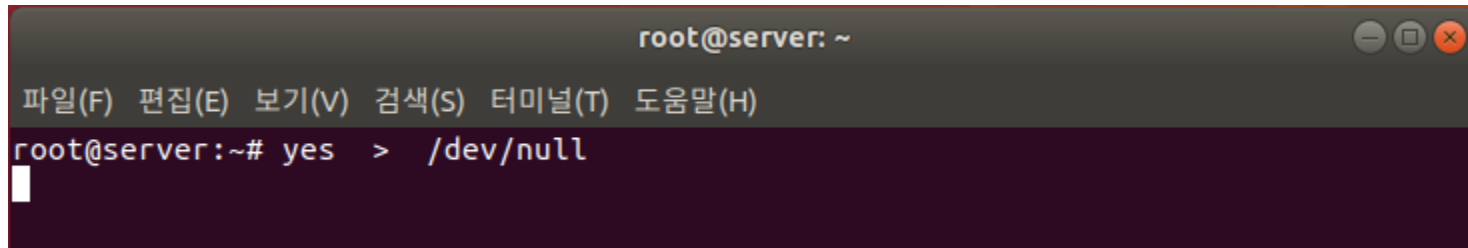
3-1 프로세스

- ps
 - 현재 프로세스의 상태를 확인하는 명령어
 - 많은 옵션과 함께 사용할 수 있음
 - 프로세스 번호와 상태를 확인할 때는 **ps -ef | grep 프로세스명** 명령을 주로 사용
- kill
 - 프로세스를 강제로 종료하는 명령어
 - '-9' 옵션과 함께 사용하면 프로세스가 무조건 종료됨
 - 응답하지 않고 무한 루프를 도는 프로세스는 **kill -9 프로세스번호** 명령으로 강제 종료 가능
- pstree
 - 부모 프로세스와 자식 프로세스의 관계를 트리 형태로 보여주는 명령어

1. 무한 루프를 도는 프로세스 중지시키기

1-1 Server 실행,

터미널에서 **yes > /dev/null** 명령 입력, 무한 루프를 도는 단순한 프로세스 생성



```
root@server: ~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
root@server:~# yes > /dev/null  
_
```

[실습 5-4] 포그라운드 프로세스와 백그라운드 프로세스 살펴보기

교재 195~197p 참고

1-2 바탕화면에서 마우스 오른쪽 버튼을 클릭하여 [터미널 열기] 선택

ps -ef | grep yes 명령으로 프로세스 번호 확인

root는 프로세스 소유자, 5526은 프로세스 번호, 4301은 부모 프로세스 번호

프로세스 번호 확인 후, **kill -9 5526** 명령을 입력, 프로세스 종료

kill 명령어로 프로세스를 종료하면 다른 터미널에서 실행 중인 것도 자동으로 종료됨

작동 중인 포그라운드 프로세스만 종료 : 프로세스가 실행되고 있는 터미널에서 Ctrl + C

```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# ps -ef | grep yes
root      5526    4301  98  22:00 pts/1    00:04:47 yes
root      5699    5687   0  22:05 pts/2    00:00:00 grep --color=auto yes
root@server:~# kill -9 5526
root@server:~#
```



```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# yes > /dev/null
죽었음
root@server:~#
```

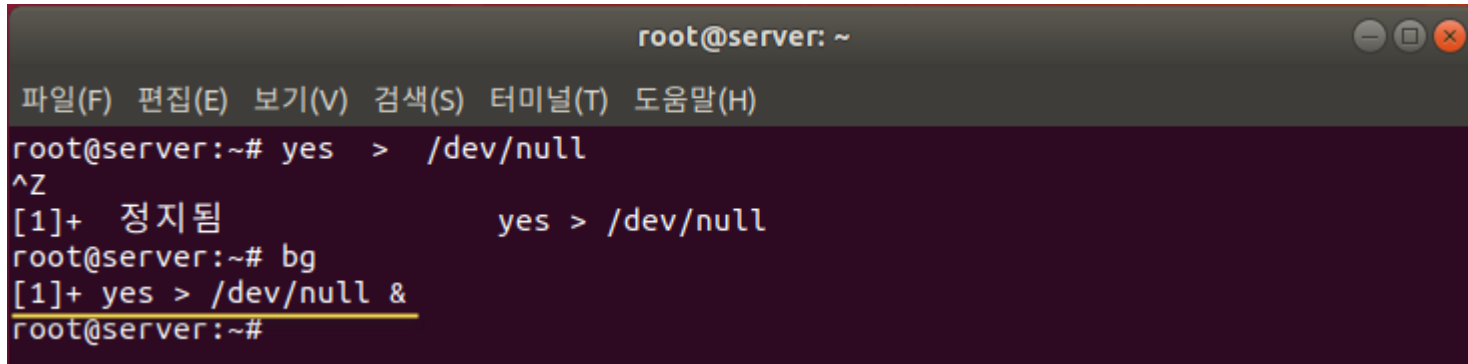
2. 프로세스 상황 바꾸기

2-1 포그라운드 프로세스 → 백그라운드 프로세스 → 포그라운드 프로세스로 바꾸기

다시 **yes > /dev/null** 명령 입력, 포그라운드 프로세스 생성

2-2 Ctrl + Z 를 눌러 프로세스를 일시 중지시키고, **bg** 명령어 입력

잠시 중지된 프로세스를 백그라운드 프로세스로 계속 실행



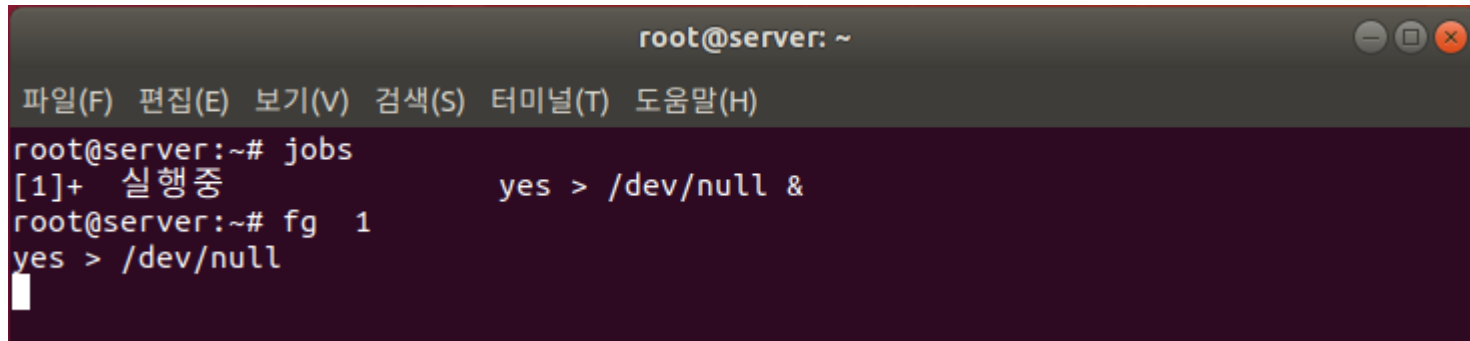
```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# yes > /dev/null
^Z
[1]+  정지됨                  yes > /dev/null
root@server:~# bg
[1]+ yes > /dev/null &
root@server:~#
```

[실습 5-4] 포그라운드 프로세스와 백그라운드 프로세스 살펴보기

교재 195~197p 참고

2-3 **jobs** 명령어를 입력하면 현재 실행 중인 백그라운드 프로세스 확인 가능
맨 앞에 나오는 것이 작업 번호

fg 작업번호 명령을 입력하면 포그라운드 프로세스로 만들 수 있음

A terminal window titled 'root@server: ~' with standard window controls. The menu bar shows '파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)'. The command history shows: 'root@server:~# jobs' followed by '[1]+ 실행중 yes > /dev/null &', then 'root@server:~# fg 1', and finally 'yes > /dev/null' with a cursor on the next line.

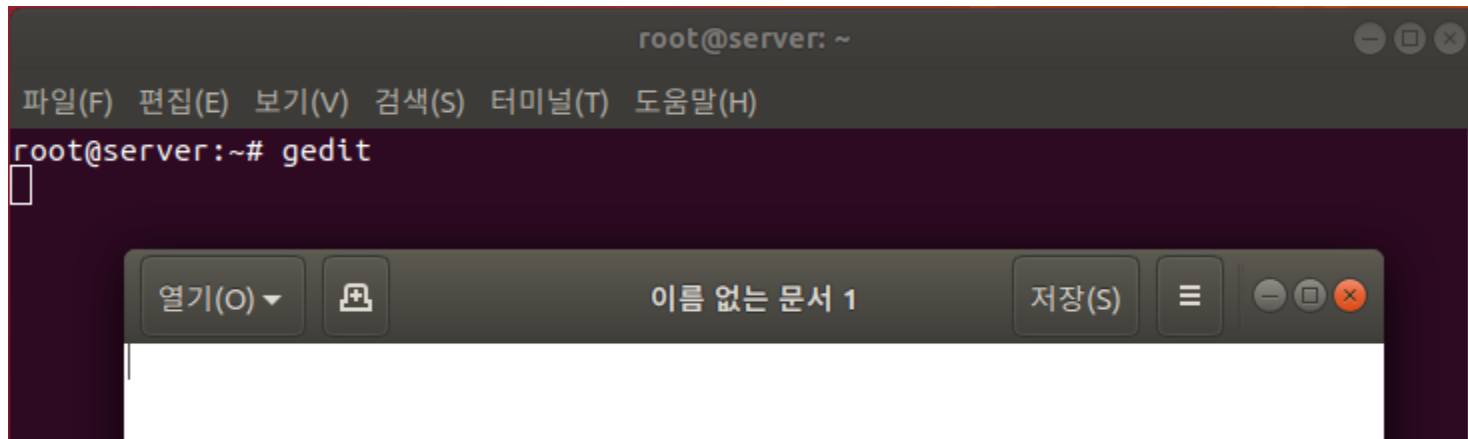
```
root@server: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# jobs
[1]+  실행중          yes > /dev/null &
root@server:~# fg 1
yes > /dev/null
█
```

2-4 Ctrl + C 를 눌러 프로세스 종료

3. 명령 실행 시 처음부터 백그라운드로 실행되도록 설정하기

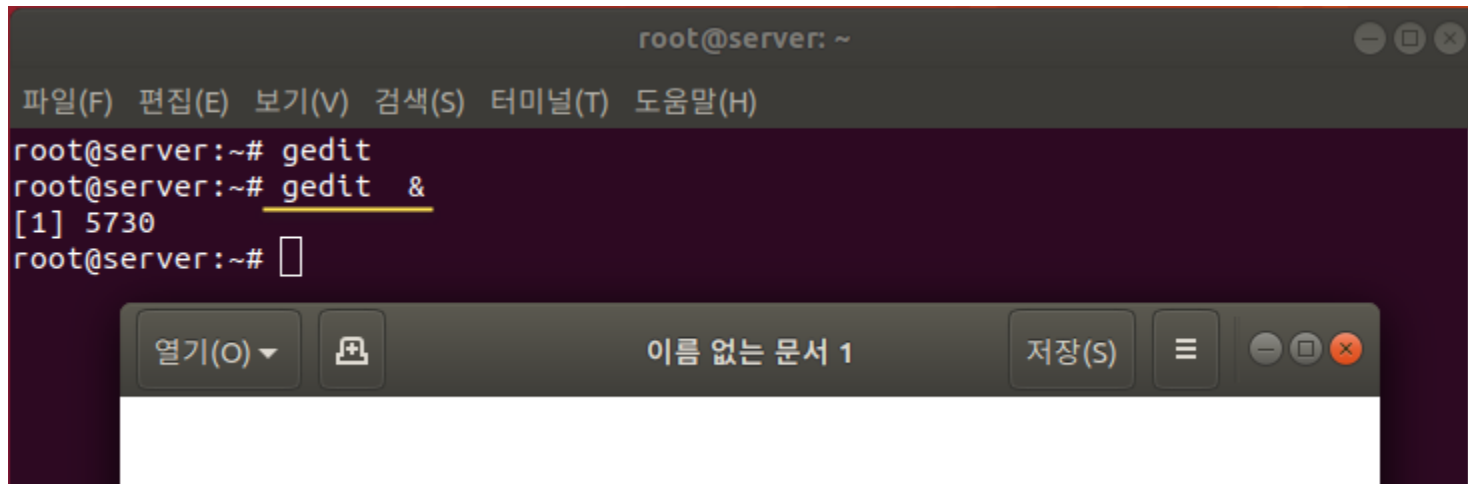
3-1 **gedit** 명령 입력, gedit 실행

그런데 **gedit** 명령을 입력한 터미널을 더 이상 사용할 수 없음



3-2 gedit 종료, **gedit &** 명령을 입력하면 터미널 계속 사용 가능

명령어 뒤에 붙이는 '&'는 백그라운드로 프로세스를 실행하겠다는 의미



3-2 서비스

- 데몬(daemon) = 서비스
 - 눈에 보이지 않지만 현재 시스템에서 동작 중인 프로세스
 - 백그라운드 프로세스의 일종
 - 서버 유형과도 밀접한 관련이 있음, '서비스=데몬=서버 프로세스'라고 이해해도 무방

3-3 서비스와 소켓의 구분

- 서비스와 소켓의 차이
 - 서비스 : 평상시에도 늘 작동하는 서버 프로세스
 - 소켓(socket) : 필요할 때만 작동하는 서버 프로세스
- 서비스의 특징
 - 시스템과 별도로 구동되는 프로세스
 - 웹 서버, DB 서버, FTP 서버 등이 있음
 - 주로 **systemctl start/stop/restart 서비스명** 명령으로 실행 및 종료
 - 디렉터리에 있는 파일은 대부분 **systemctl start/stop/restart 서비스명** 명령으로 실행, 중지, 재시작 가능

```
root@server: /lib/systemd/system
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:/lib/systemd/system# pwd
/lib/systemd/system
root@server:/lib/systemd/system# ls *.service
ModemManager.service          pppd-dns.service
NetworkManager-dispatcher.service  procps.service
NetworkManager-wait-online.service  quotaon.service
NetworkManager.service         rc-local.service
accounts-daemon.service        rc.local.service
```

3-3 서비스와 소켓의 구분

■ 소켓의 특징

- 서비스는 항상 구동 중이지만 소켓은 외부에서 특정 서비스를 요청하는 경우에만 systemd가 구동, 요청이 끝나면 소켓은 종료됨
- 소켓으로 설정된 서비스를 요청하는 경우 처음 연결되는 데 걸리는 시간이 서비스보다 더 길 수도 있음
- 소켓과 관련된 스크립트 파일은 `/lib/systemd/system/` 디렉터리에 있는 '소켓명.socket'

```
root@server: /lib/systemd/system
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:/lib/systemd/system# ls *.socket
acpid.socket                systemd-initctl.socket
apport-forward.socket       systemd-journald-audit.socket
avahi-daemon.socket         systemd-journald-dev-log.socket
cups.socket                 systemd-journald.socket
dbus.socket                 systemd-networkd.socket
saned.socket                systemd-rfkill.socket
snapd.socket                systemd-udevd-control.socket
syslog.socket               systemd-udevd-kernel.socket
systemd-fsckd.socket        uidd.socket
root@server:/lib/systemd/system#
```




Thank You
