

04

## CHAPTER

# 리눅스 사용자 관리와 파일 관리



# Contents

---

- 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.

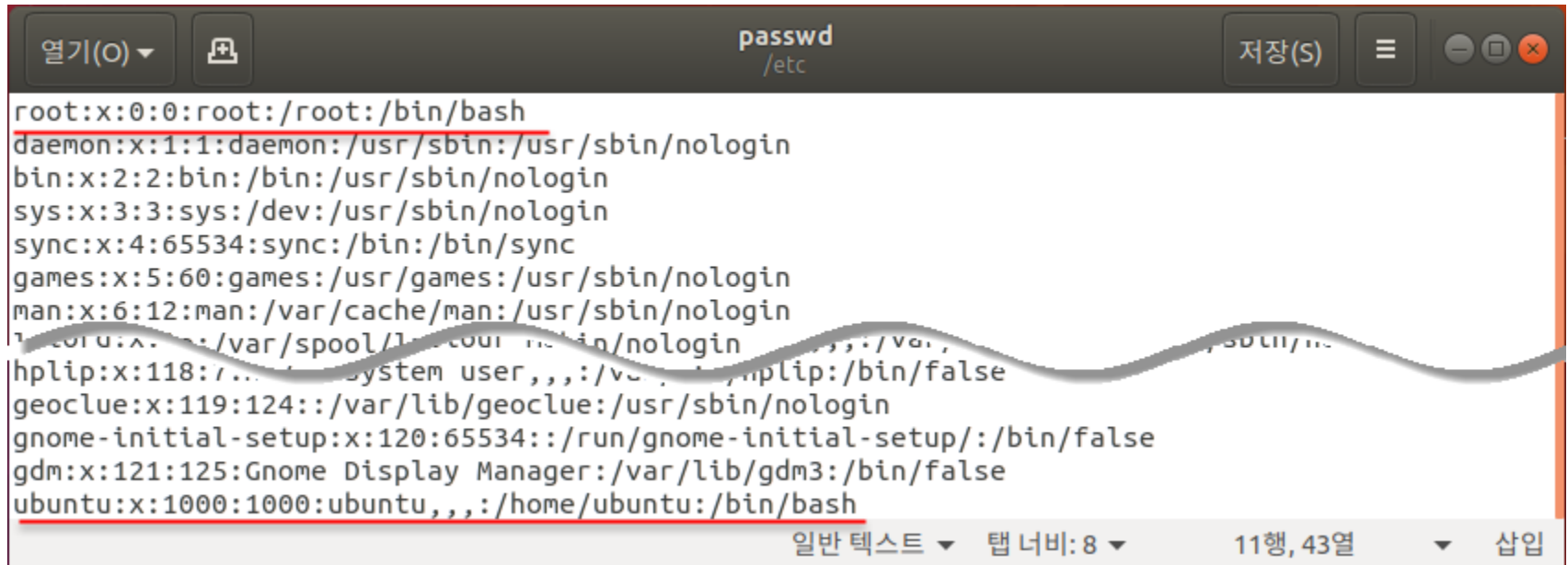
# 1-1 사용자와 그룹의 개념

- 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

# 1-1 사용자와 그룹의 개념

## ■ 사용자 list

- /etc/passwd
- 각 행은 '사용자 이름:비밀번호:사용자 ID:사용자 소속 그룹 ID:추가 정보:홈 디렉터리: 기본 셸'을 의미



```
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
lxd:x:100:100:libvirt:/var/lib/containers:/usr/sbin/nologin
hplip:x:118:lp:HP System user,,,:/var/lib/hplip:/bin/false
geoclue:x:119:124:geoclue:/var/lib/geoclue:/usr/sbin/nologin
gnome-initial-setup:x:120:65534:GNOME Initial Setup:/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
```

# 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

## ■ Group

- /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
  - 그룹의 권한을 공유 받음
- 여러 그룹이면 쉼표(,)로 구분



```
group
/etc

root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,ubuntu
tty:x:5:
disk:x:6:
lxd:x:7:
avahi:x:122:
colord:x:123:
geoclue:x:124:
gdm:x:125:
ubuntu:x:1000:
sambashare:x:126:ubuntu
```

The screenshot shows a text editor window titled 'group /etc'. The file content lists system groups (root, daemon, bin, sys, adm, tty, disk, lxd) and regular users (avahi, colord, geoclue, gdm, ubuntu, sambashare). The lines for root, sys, and ubuntu are underlined in red. The bottom status bar indicates '일반 텍스트', '탭 너비: 8', '1행, 1열', and '삽입' mode.

## 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/deluser

- 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
```

- 특정 사용자를 sudo group에 추가하자

# 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
```

## [실습 5-1] 사용자와 그룹 관리하기

1. 가상머신 접속

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:~#
```

## [실습 5-1] 사용자와 그룹 관리하기

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:~#
```

## [실습 5-1] 사용자와 그룹 관리하기

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:~# 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

## [실습 5-1] 사용자와 그룹 관리하기

### 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:~# 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:~#
```

## [실습 5-1] 사용자와 그룹 관리하기

- 3-2 **adduser --gid 그룹ID 사용자** 명령으로 새로운 hanbit1, hanbit2 사용자 생성,  
ubuntuGroup 그룹(그룹 ID는 1001)으로 지정

```
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 ...
'/home/hanbit1'에서 파일들을 복사하는 중 ...
접근이려 함: []:
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 ...
```



## [실습 5-1] 사용자와 그룹 관리하기

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:$6$vlNL76yl$LfSuNmFC8GU3G8i0q51xaLNypRXnQEw0k8E/5EwtZ2fhaz0miQRlgajH2wWI
9vmRlc3drSVR8u6Xhe7SfA5120:18076:0:99999:7:::
hanbit2:$6$0lgU8Tor$hlztc7fqiyIhYphD3g09Vv95mSEh7603kMSJvM5IWHwHLWcyysSeknPRsAvD
s7t2ufEp8woSLR.qkYnwrHnLj1:18076:0:99999:7:::
root@server:~#
```

## [실습 5-2] 사용자 변경

- su (substitute user) – 사용자 변경

#user1로 사용자 변경

su - user1

su user1

#root로 사용자 변경 (root 생략가능)

su root    혹은    su

Su - root    혹은    su -

" - "를 사용하면, 환경변수를 다 삭제

" - " 를 사용하지 않으면,  
그 전 login계정의 환경변수를 가져옴

```
heejinlee@ubuntu:~$ pwd
/home/heejinlee
heejinlee@ubuntu:~$ su - user1
Password:
user1@ubuntu:~$ pwd
/home/user1
user1@ubuntu:~$ su user20
Password:
user20@ubuntu:/home/user1$ pwd
/home/user1
```

## [실습 5-2] Root 계정 동작하기 su root

- Root 계정으로 동작하기 - su root
  - 시스템 설정 변경등과 같은 중요한 모든 권한을 다 가지고 있다.
  - root 비밀번호 설정 필요
    - sudo passwd
    - Passwd 변경시에도 passwd 명령어
  - 한번 설정해주면 su root 가능

```
heejinlee@ubuntu:~$ su root
Password:

su: Authentication failure
heejinlee@ubuntu:~$
heejinlee@ubuntu:~$ sudo passwd
New password:
Retype new password:
passwd: password updated successful
heejinlee@ubuntu:~$ su root
Password:
root@ubuntu:/home/heejinlee# pwd
/home/heejinlee
root@ubuntu:/home/heejinlee# cd
root@ubuntu:~#
```

## [실습 5-3] sudo 는?

### ■ Sudo 는?

- Root 처럼 모든 중요한 관리자 기능을 수행할 수 있다.
- 원래는 root만 관리자 기능을 할 수 있어야 함.
- 다른 사람들은 su root를 통하여 사용자 변경을 해서 관리자 기능을 해야함
- Root의 비밀번호는 처음에 setting이 안되어 있음.
- 그래서 맨 처음 user만 root의 비밀번호를 setting할수 있는 권한을 줌
  - 맨 처음 user만 기본적으로 sudo group에 속하게 함 (sudo group의 권한)
  - Sudo 를 붙이면 root 처럼 모든 권한을 가지고 동작
- Sudo group에 속해져 있는 사용자만 sudo 명령어 가능하다.
  - Sudo adduser *username* sudo
- See "man sudo\_root" for details.

## [실습 5-4] 새로운 사용자에게 기본 setting을 줄때

- /etc/skel 디렉토리
  - 새로 만든 사용자의 home directory(/home/username)는 /etc/skel과 동일함
- 실습
  - /etc/skel에 welcome.txt를 하나 만든다.
  - user1 사용자를 새로 생성한다.
  - /home/user1과 /etc/skel 디렉터리 비교한다.

```
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:~#
```

# [실습 5-5] 사용자와 그룹 관리하기

## 4. X 윈도우 환경에서 사용자 관리하기

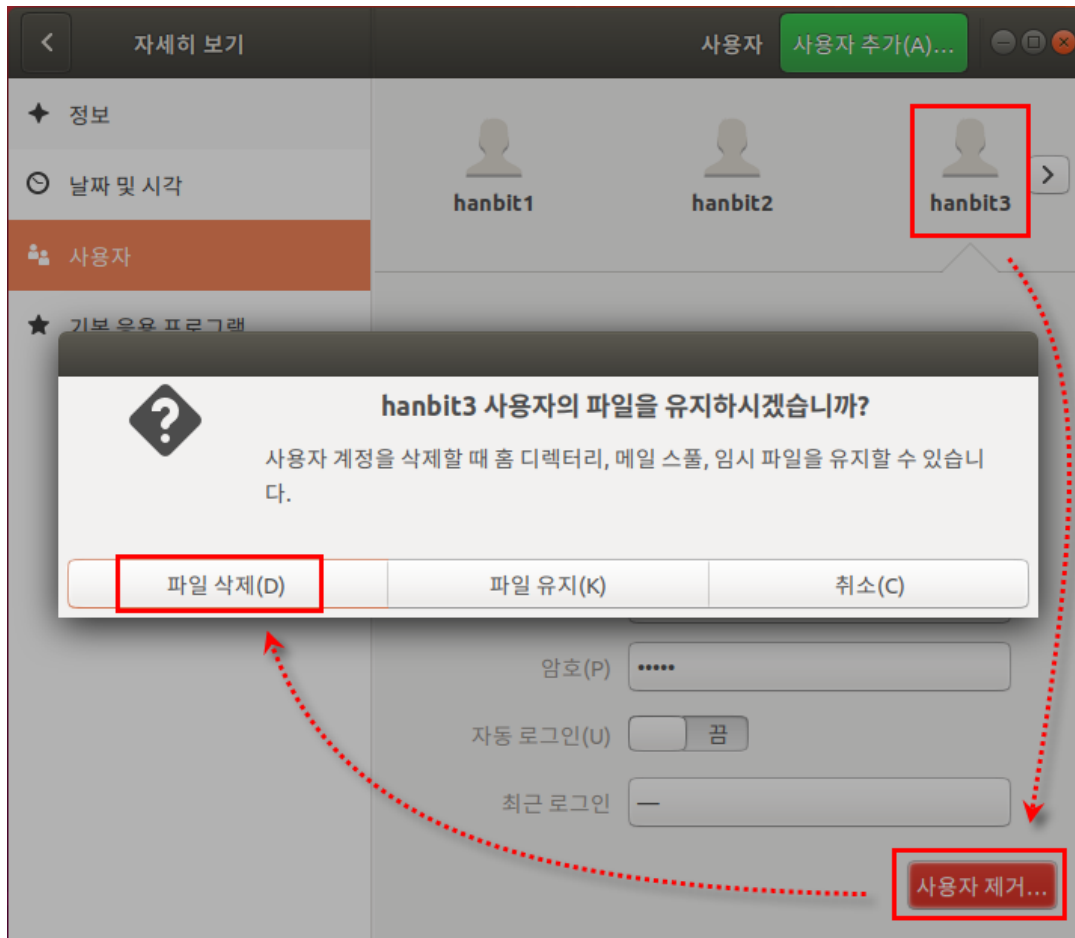
4-1 바탕화면에서 오른쪽 위의 ▼ 아이콘에 이어 ✱ 아이콘 클릭

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

The image shows two windows from a Linux desktop environment. The left window, titled '자세히 보기' (Details View), has a sidebar with '정보' (Info), '날짜 및 시각' (Date and Time), '사용자' (Users), and '기본 응용 프로그램' (Default Applications). The '사용자' section is highlighted. The 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) set to '표준' (Standard), '언어(L)' (Language) set to '한국어' (Korean), '암호(P)' (Password) masked with dots, '자동 로그인(U)' (Auto Login) set to '끔' (Off), and '최근 로그인' (Last Login) set to '—'. A red dashed arrow points from the '사용자' sidebar item to the '사용자 추가(A)...' button in the top right corner. The right window, titled '사용자 추가(A)...', has a '취소(C)' (Cancel) button and a '추가(A)' (Add) button. It contains fields for '계정 종류(T)' (Account Type) with '표준' (Standard) and '관리자' (Administrator) options, '전체 이름(F)' (Full Name) set to 'hanbit3', and '사용자 이름(U)' (Username) set to 'hanbit3'. Below these is a section for '암호' (Password) with options for '다음 로그인에 암호를 설정하도록 허용(L)' (Allow to set password at next login) and '지금 암호 설정(N)' (Set password now). The '지금 암호 설정(N)' option is selected. It includes fields for '암호(P)' (Password) and '확인(C)' (Confirm), both masked with dots. A red dashed arrow points from the '추가(A)' button in the left window to the '추가(A)' button in the right window. Another red dashed arrow points from the '전체 이름(F)' field in the right window to the '사용자 추가(A)...' button in the left window.

## [실습 5-5] 사용자와 그룹 관리하기

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

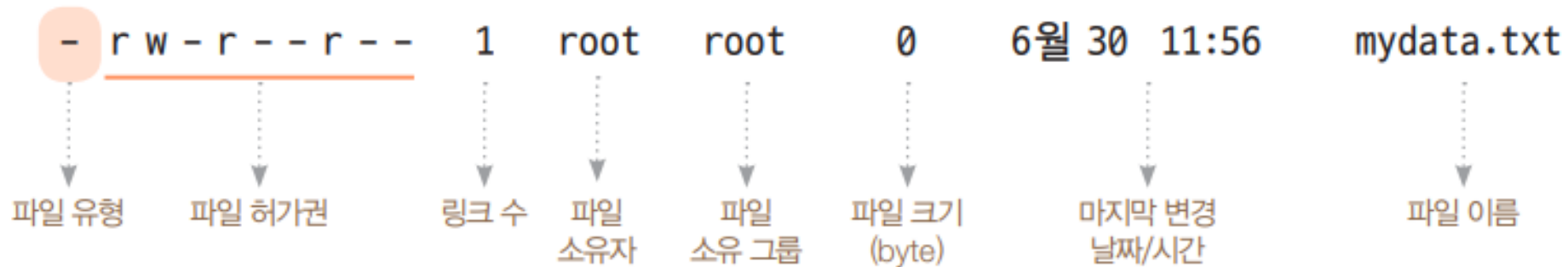


4-4 오른쪽 위의 X를 클릭, 설정 창 닫기

## 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:~# 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 (permission)
    - 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

- 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

### 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:~# ls -l sample
-rw-r--r-- 1 root root 49 6월 30 12:40 sample
root@server:~#
```

## [실습 5-2] 파일 허가권과 소유권 실행하기

### 2. 파일 속성 변경하기

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

```
root@server:~# whoami
root
root@server:~# ./sample
bash: ./sample: 허가 거부
root@server:~#
```

## [실습 5-2] 파일 허가권과 소유권 실행하기

### 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:~# 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: 쿼백: 명령어를 찾을 수 없음
backups crash local log metrics run spool
cache lib lock mail opt snap tmp
root@server:~#
```

## [실습 5-2] 파일 허가권과 소유권 실행하기

### 3. 소유권 변경하기

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

```
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:~#
```

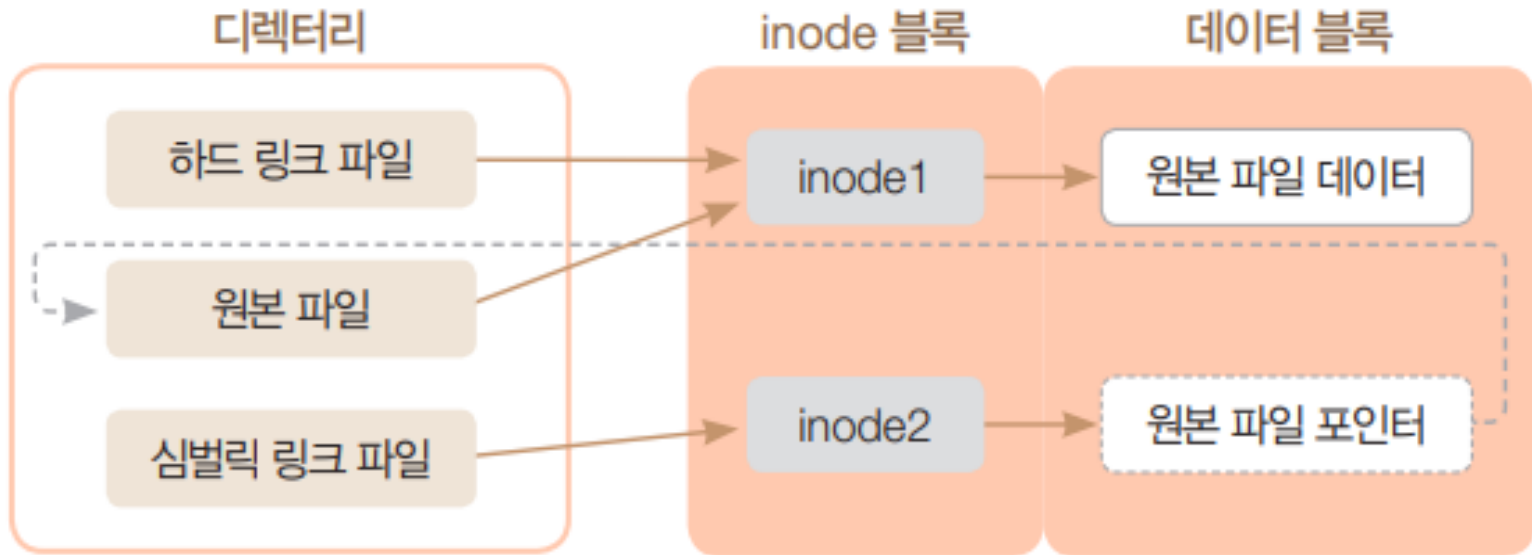
3-2 **chgrp ubuntu sample** 명령 입력, 그룹을 ubuntu 그룹으로 변경

사용자와 그룹을 한꺼번에 바꾸려면 **chown Ubuntu:ubuntu sample** 명령 실행

```
root@server:~# chgrp ubuntu sample
root@server:~# ls -l sample
-rwxr-xr-x 1 ubuntu ubuntu 49  6월 30 12:40 sample
root@server:~#
```

## 2-4 링크

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





## [실습 5-3] 하드 링크와 심벌릭 링크 생성하기

### 1. 파일 생성하고 확인하기

#### 1-1 Server 실행, originalfile 파일 생성

vi 에디터나 gedit를 이용하여 '파일 링크 실습용 원본 파일' 입력 후 저장

**cat** 명령으로 파일 내용 확인

```
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

## 2. 하드 링크와 심벌릭 링크 확인하기

### 2-1 하드 링크와 심벌릭 링크 파일 생성

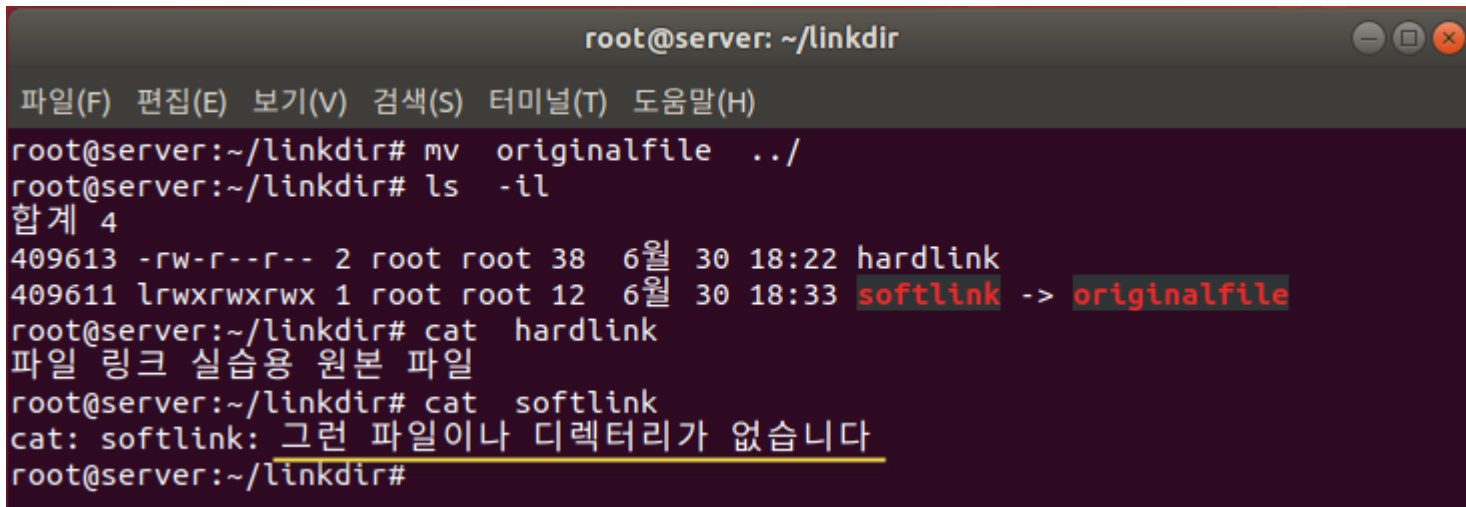
```
ln originalfile hardlink    -- 하드 링크 생성
ln -s originalfile softlink -- 심벌릭 링크(소프트 링크) 생성
ls -il                      -- -il 옵션은 inode 번호를 맨 앞에 출력
cat hardlink                -- 하드 링크의 내용 확인
cat softlink                -- 심벌릭 링크의 내용 확인
```

```
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

### 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#
```

➔ hardlink는 정상 동작

➔ Symbolic link는 비정상 동작 (원본 파일 찾지 못함)

## [실습 5-3] 하드 링크와 심벌릭 링크 생성하기

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

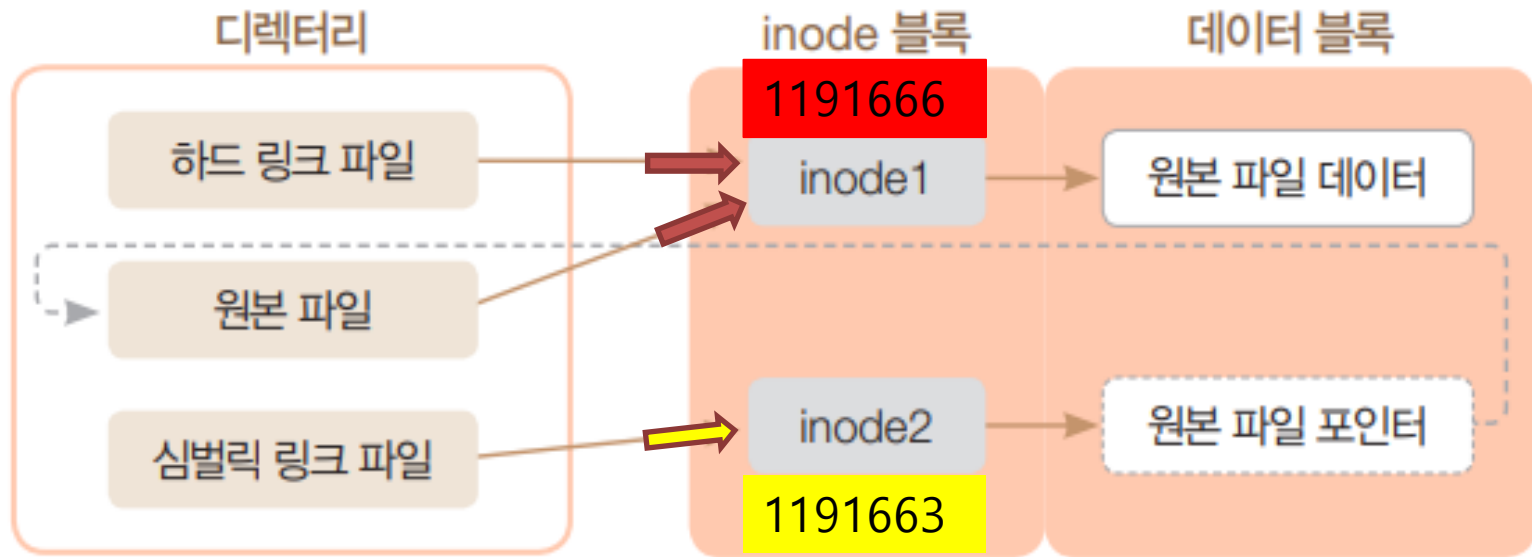
```
root@server: ~/linkdir
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~/linkdir# mv ../originalfile .
root@server:~/linkdir# ls -il
합계 8
409613 -rw-r--r-- 2 root root 38 6월 30 18:22 hardlink
409613 -rw-r--r-- 2 root root 38 6월 30 18:22 originalfile
409611 lrwxrwxrwx 1 root root 12 6월 30 18:33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

## [실습 5-3] 하드 링크와 심벌릭 링크 생성하기

- stat file명
  - File의 Inode, links, file 유형을 확인

```
hjb@ubuntu:~/link$ stat symboliclink original hardlink
  File: symboliclink -> original
  Size: 8              Blocks: 0          IO Block: 4096   symbolic link
Device: 805h/2053d    Inode: 1191663      Links: 1
Access: (0777/lrwxrwxrwx)  Uid: ( 1000/      hj)   Gid: ( 1000/      hj)
Access: 2022-03-27 06:43:08.573722469 -0700
Modify: 2022-03-27 06:43:06.989730581 -0700
Change: 2022-03-27 06:43:06.989730581 -0700
 Birth: -
  File: original
  Size: 17             Blocks: 8          IO Block: 4096   regular file
Device: 805h/2053d    Inode: 1191666      Links: 2
Access: (0664/-rw-rw-r--)  Uid: ( 1000/      hj)   Gid: ( 1000/      hj)
Access: 2022-03-27 06:42:53.897797543 -0700
Modify: 2022-03-27 06:42:53.897797543 -0700
Change: 2022-03-27 06:45:56.208849418 -0700
 Birth: -
  File: hardlink
  Size: 17             Blocks: 8          IO Block: 4096   regular file
Device: 805h/2053d    Inode: 1191666      Links: 2
```

## [실습 5-3] 하드 링크와 심벌릭 링크 생성하기



- Hardlink를 여러 개 더 만들어 Inode는 동일하고, links 변경되는 것을 확인한다.
- Symbolic link를 여러 개 더 만들어 다른 Inode가 생성되는 것을 확인한다.

## 하드 링크와 심벌릭 링크 차이

- 하드링크는 inode를 그대로 가르키고 있기 때문에, 원본이 삭제되더라도 그대로 사용 가능
  - 원본을 유지할 수 있는 장점
  - 파악하지 못한 하드링크가 있는 경우 불필요한 저장소 낭비
  - 원본이 이동되어도 사용 가능
- 심벌릭링크는 원본 이름만 같으면, 다른 파일이어도 동작.

## 3-1 프로세스

### ■ 프로세스 정의

- 하드디스크에 저장된 실행 코드(프로그램)가 메모리에 로딩되어 활성화된 것
- Ex) 웹 브라우저 프로그램인 파이어폭스는 '프로그램', 파이어폭스를 실행하여 화면에 나타난 상태(메모리에 로딩되어 화면에 나타난 상태가 더 정확한 표현)를 '프로세스'라 부름

### ■ Foreground process

- 실행하면 화면에 나타나서 사용자와 상호 작용을 하는 프로세스
- 사용자와 상호작용을 해야만 하는 작업

### ■ Background process

- 화면에 나타나지 않은 채 뒤에서 실행되는 프로세스
- 오래 걸리는 작업, 상호작용을 하지 않아도 되는 작업 실행



## 3-1 프로세스

- 프로세스 번호
  - 각 프로세스에 할당된 고유 번호
  - 메모리에 로딩되어 활성화된 프로세스를 구분하려면 고유 번호가 필요함
- 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

```
heejinlee@ubuntu:~$ ps -ef >>a
heejinlee@ubuntu:~$ head -10 a
UID          PID     PPID  C  STIME TTY          TIME CMD
root           1         0  3   19:32 ?        00:00:02 /sbin/init auto noprompt
root           2         0  0   19:32 ?        00:00:00 [kthreadd]
root           3         2  0   19:32 ?        00:00:00 [rcu_gp]
root           4         2  0   19:32 ?        00:00:00 [rcu_par_gp]
root           5         2  0   19:32 ?        00:00:00 [kworker/0:0-events]
root           6         2  0   19:32 ?        00:00:00 [kworker/0:0H-kblockd]
root           7         2  0   19:32 ?        00:00:00 [kworker/0:1-cgroup_destroy]
root           8         2  0   19:32 ?        00:00:00 [kworker/u256:0-events_unbound]
```

## 3-1 프로세스

### ■ pstree

- 부모 프로세스와 자식 프로세스의 관계를 트리 형태로 보여주는 명령어

```
heejinlee@ubuntu:~$ pstree
systemd--ModemManager--2*[{ModemManager}]
--NetworkManager--2*[{NetworkManager}]
--VGAAuthService
--accounts-daemon--2*[{accounts-daemon}]
--acpid
--anacron
--apache2--2*[apache2--26*[{apache2}]]
--apt.systemd.dai--apt.systemd.dai--unattended-upgr--http
--2*[{unattended-upgr}]
--avahi-daemon--avahi-daemon
--colord--2*[{colord}]
--cron
--cups-browsed--2*[{cups-browsed}]
--cupsd--dbus
--dbus-daemon
--gdm3--gdm-session-wor--gdm-x-session--Xorg--{Xorg}
--gnome-session-b--ssh-agent
--2*[{gnome-session-b}]
--2*[{gdm-x-session}]
--2*[{gdm-session-wor}]
--2*[{gdm3}]
--gnome-keyring-d--3*[{gnome-keyring-d}]
```

## 3-1 프로세스

- 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
- jps
  - 실행 중인 Java process를 보여주는 명령어

# [실습] Firefox, Rhythmbox 를 실행하고, 프로세스 명령어로 확인

- 우분투에서 프로세스들을 실행한다.
  - firefox, rhythmbox등을 실행
- Ps -ef, pstree등의 명령어로 rhythmbox가 process에 존재하는지 확인한다.
  - ps -ef | grep rhythmbox
  - ps -ef | grep firefox
  - pstree | grep firefox

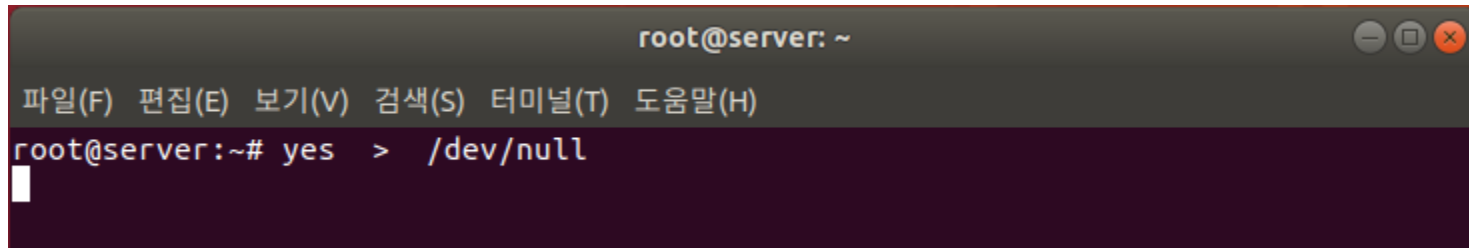
```
heejinlee@ubuntu:/home/user11$ ps -ef | grep firefox
heejinl+  97422      1603  60  15:20 ?                00:00:03 /usr/lib/firefox/firefox -new-window
heejinl+  97579      97422  13  15:20 ?                00:00:00 /usr/lib/firefox/firefox -contentproc -ch
heejinl+  97636      97422   8  15:20 ?                00:00:00 /usr/lib/firefox/firefox -contentproc -ch
heejinl+  97682      97422   2  15:20 ?                00:00:00 /usr/lib/firefox/firefox -contentproc -ch
heejinl+  97698      97422   2  15:20 ?                00:00:00 /usr/lib/firefox/firefox -contentproc -ch
heejinl+  97706      97422   2  15:20 ?                00:00:00 /usr/lib/firefox/firefox -contentproc -ch
heejinl+  97791       2669   0  15:20 pts/0          00:00:00 grep --color=auto firefox
```

```
heejinlee@ubuntu:~$ pstree | grep firefox
      |      | -firefox+-Privileged Cont---14*[{Privileged Cont}]
      |      |      `--65*[{firefox}]
heejinlee@ubuntu:~$ pstree | grep rhythmbox
      |      | -gnome-shell+-rhythmbox---11*[{rhythmbox}]
```

## [실습 5-4] 포그라운드 프로세스와 백그라운드 프로세스 살펴보기

### 1. 무한 루프를 도는 프로세스 중지시키기

1-1 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

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

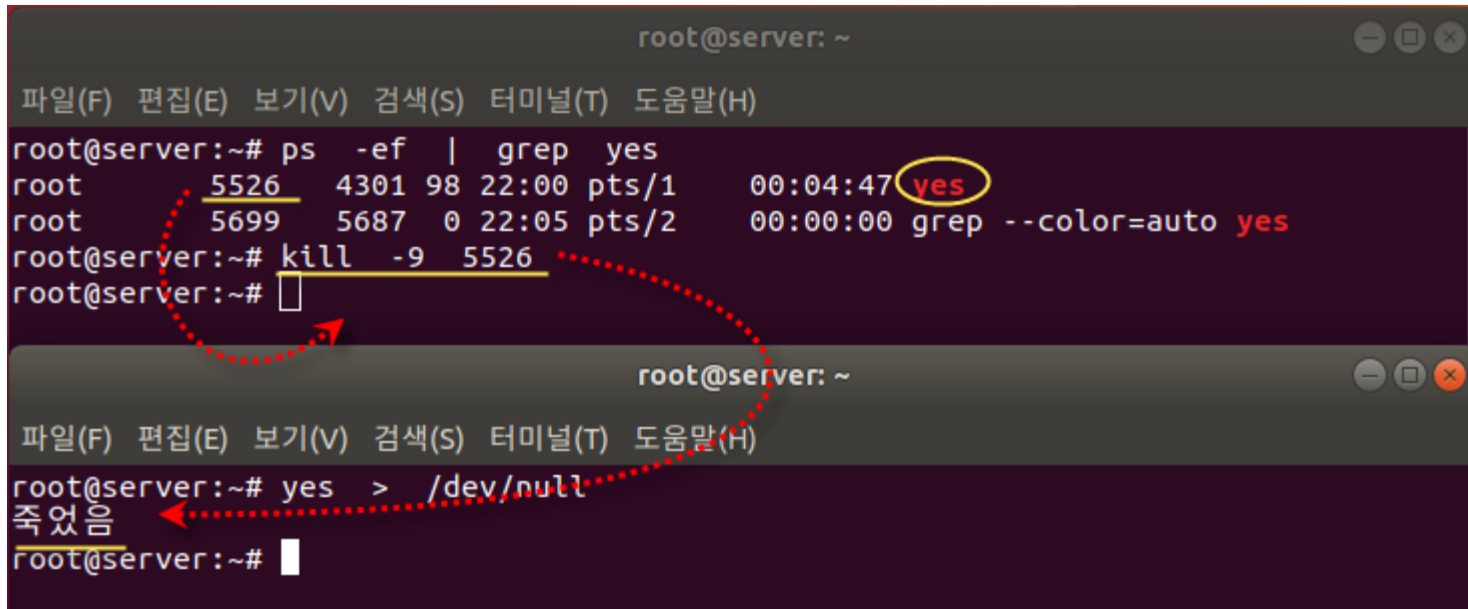
**ps -ef | grep yes** 명령으로 프로세스 번호 확인

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:~#
```

## [실습 5-4] 포그라운드 프로세스와 백그라운드 프로세스 살펴보기

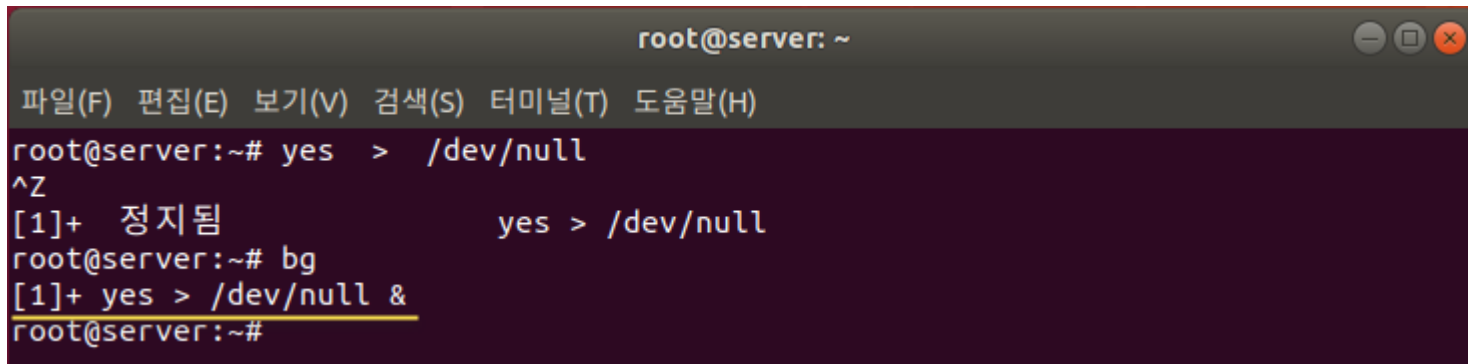
### 2. 프로세스 상황 바꾸기

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

2-3 **jobs** 명령어 - 현재 실행 중인 백그라운드 프로세스 확인 가능

맨 앞에 나오는 것이 작업 번호 (현재 실행 중인 백그라운드 프로세스의 순차 번호)

**fg** **작업번호** 명령을 입력하면 포그라운드 프로세스로 만들 수 있음

```
root@server:~# jobs
[1]+  실행중                yes > /dev/null &
root@server:~# fg 1
yes > /dev/null
^Z
[1]+  Stopped                yes > /dev/null
heejinlee@ubuntu:~$ jobs
[1]+  Stopped                yes > /dev/null
heejinlee@ubuntu:~$ bg
[1]+ yes > /dev/null &
heejinlee@ubuntu:~$ jobs
[1]+  Running                yes > /dev/null &
```

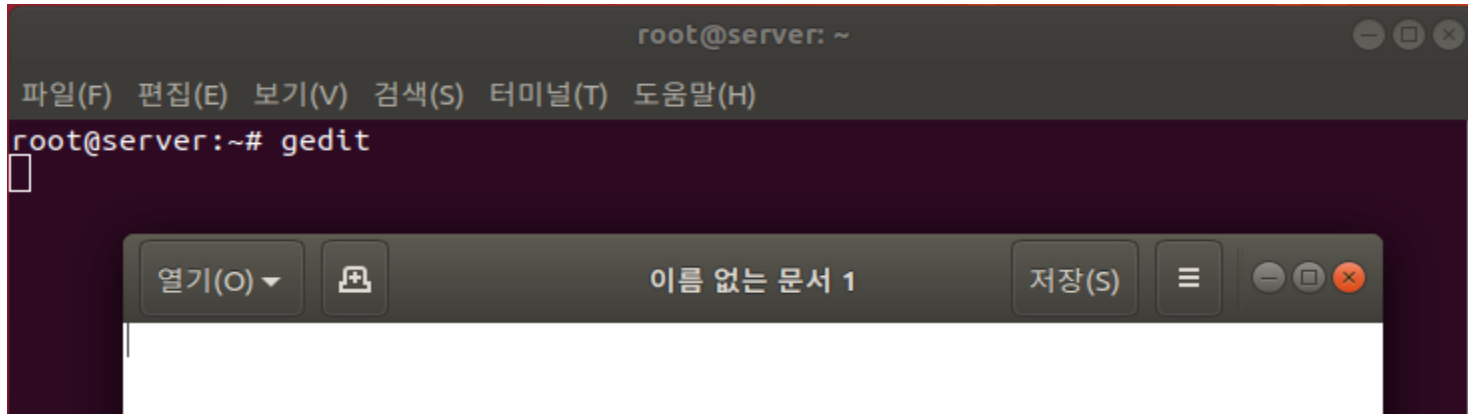
2-4 Ctrl + C 를 눌러 프로세스 종료

## [실습 5-4] 포그라운드 프로세스와 백그라운드 프로세스 살펴보기

3. 명령 실행 시 처음부터 백그라운드로 실행되도록 설정하기

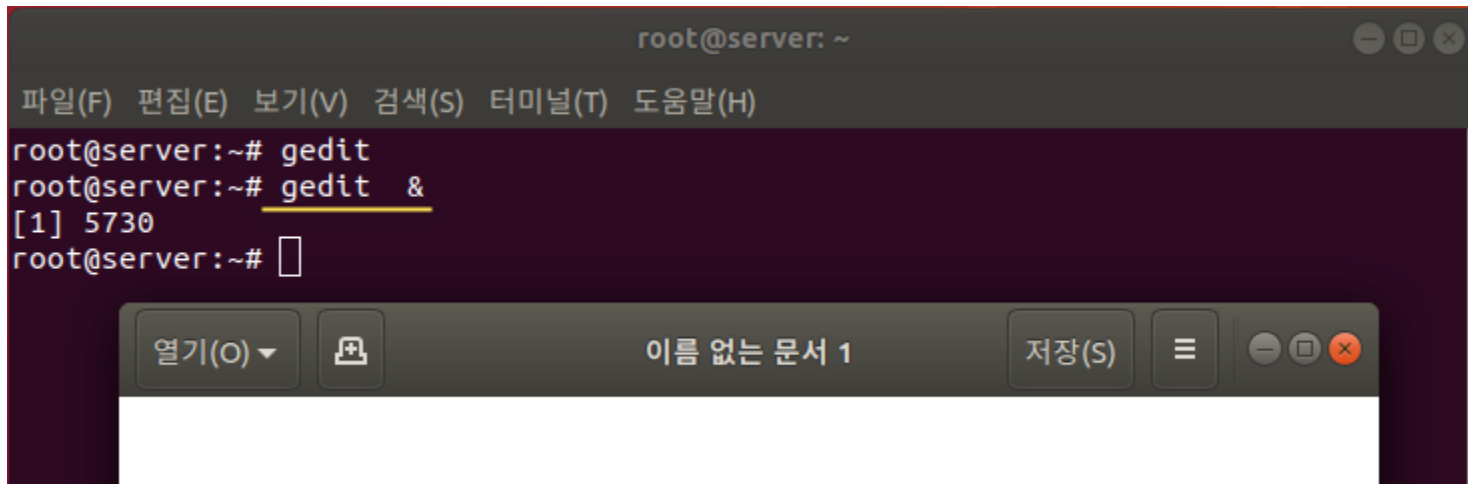
3-1 **gedit** 명령 입력, gedit 실행

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



## [ 과제 및 시험 ]

- 시험범위
  - 리눅스프로그래밍 교재 chap1,2 모두
  - PPT 처음 ~ chap 4 (특히 chap4)
- 시험 준비
  - 교재에 있는 것들을 직접 실습해 볼것



Thank You

---