

# Chapter 04

## 리눅스 사용자와 파일



## 학습목표

- 리눅스의 슈퍼유저와 사용자에 대해 설명할 수 있다.
- 리눅스 파일 관련 명령어를 사용할 수 있다.
- 리눅스 파일의 특징을 설명할 수 있다.
- 파일 압축 명령어를 사용할 수 있다.



## 학습내용

- ❖ 리눅스 사용자 관리
  - 리눅스의 사용자
- ❖ 리눅스 파일
  - 파일 관련 명령어
  - 리눅스 파일의 특징
  - 파일 압축 및 tar



## 리눅스의 사용자

### 리눅스의 다중 사용자

- 마이크로 소프트사의 윈도우 OS는 여러 사용자를 추가할 수는 있지만 대개 한 사용자가 쓰는 구조로 되어 있음
- 하지만 리눅스와 같은 유닉스 계열 OS는 여러 사용자가 동시에 접속해서 사용할 수 있는 구조로 되어 있음.
- 즉, 리눅스 서버 1대에 여러 사용자가 동시에 접속 가능
- 여러 사용자의 쉬운 관리를 위해 그룹이 있음. 속성이 같은 사용자끼리 하나의 그룹으로 묶음.
- 한 명의 사용자는 자신의 홈 디렉토리를 가지며 위치는 “/home/사용자 이름”로 지정됨.



## 리눅스의 사용자

### 슈퍼 유저 = root

- 리눅스는 여러 사용자중 단 한 명의 슈퍼 유저가 존재함.
- 슈퍼 유저(Super user) 를 root 라고도 함.
- Root는 리눅스에서 신과 같은 존재로 모든 권한을 가지고 있음.
- Root는 모든 것을 할 수 있음.
- Root만이 실행할 수 있는 명령어가 존재함.
- Root의 홈 디렉토리는 /root 에 위치
- Root의 셸의 프롬프트는 # 이고 일반 사용자의 프롬프트는 \$ 임



## 리눅스의 사용자



### 일반 사용자

- 홈 디렉토리는 /home/사용자이름
- 권한이 있는 파일이나 디렉토리만 읽거나 쓰거나 실행가능
- 실행가능한 명령어가 제한됨.
- 일반 사용자의 프롬프트는 \$ (배쉬 셸의 경우)

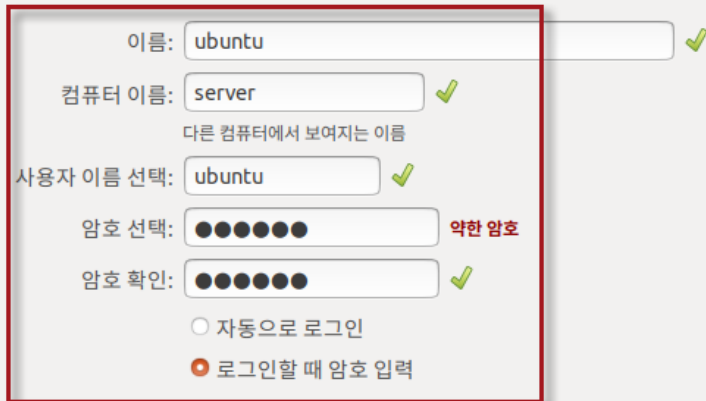


## 리눅스의 사용자

### 일반 사용자

- Server 가상머신에서 현재 일반 사용자는 “ubuntu” 라는 이름의 사용자
- 우분투를 설치할 때 다음과 같은 화면에서 ubuntu를 입력했기 때문

당신은 누구십니까?



이름: ubuntu ✓

컴퓨터 이름: server ✓  
다른 컴퓨터에서 보여지는 이름

사용자 이름 선택: ubuntu ✓

암호 선택: ●●●●●● 약한 암호

암호 확인: ●●●●●● ✓

☐ 자동으로 로그인

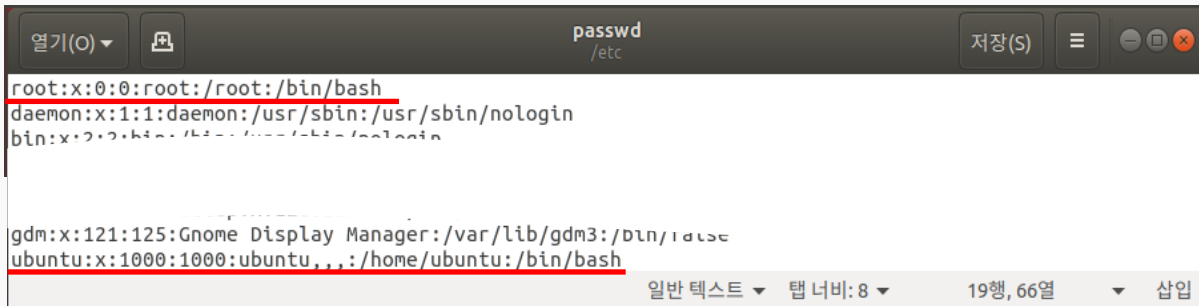
☒ 로그인할 때 암호 입력



## 리눅스의 사용자

### 일반 사용자

- 일반 사용자인 ubuntu 의 홈 디렉토리는 /home/ubuntu
- Gedit를 사용하여 /etc/passwd 파일 열기



```
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

gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
```

일반 텍스트 ▼ 탭 너비: 8 ▼ 19행, 66열 ▼ 삽입

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



## 리눅스의 사용자

### 일반 사용자 ubuntu

```
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



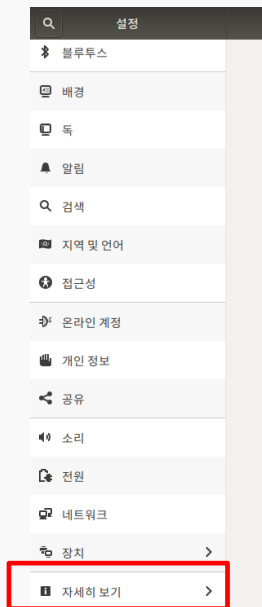
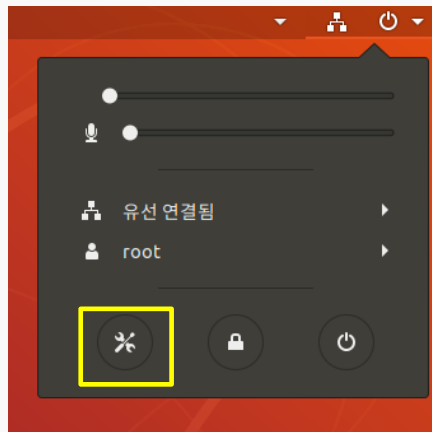
# 리눅스 사용자 관리



## 리눅스의 사용자

### 사용자 추가하기

- ① 바탕화면의 전원 버튼의 설정 버튼 선택
- ② 자세히 보기 선택



# 리눅스 사용자 관리



## 리눅스의 사용자

③ 사용자

④ 사용자 추가

⑤ 전체 이름과 암호를 설정

⑥ 추가

⑦ 터미널에서 /home 밑에 korea1이 생성되었음을 확인

## 사용자 추가하기

자세히 보기

사용자

사용자 추가(A)...

ubuntu

계정 종류(T) 표준 관리자

언어(L) 한국어

암호(P) .....

자동 로그인(U) ☐ 끄

최근 로그인 오늘, 17:14

```
root@server:~# ls /home/  
korea1 ubuntu  
root@server:~#
```

취소(C) 사용자 추가 추가(A)

계정 종류(T) 표준 관리자

전체 이름 korea1 ✓

사용자 이름(U) korea1 ✓

홈 폴더의 이름입니다. 바꿀 수 없습니다.

암호

☐ 다음 로그인에 암호를 설정하도록 허용(L)

☒ 지금 암호 설정(N)

암호(P) ..... ✓

영문자, 숫자, 문장 기호를 더 쓰면 더 강력합니다.

확인(C) ..... ✓



## 파일 관련 명령어

### 리눅스 명령어의 규칙

- 모든 명령어는 영문 소문자이다.
- 일반 사용자등 모두가 사용할 수 있는 명령어와 Root만 사용할 수 있는 명령어가 있다.
- 명령어의 종류
  - ✓ 내부 명령어 : 실행파일이 따로 없고 쉘 프로그램에 포함되어 있는 명령어 (예 cd)
  - ✓ 외부 명령어 : 실행파일이 따로 존재하는 명령어(예: ls는 /bin/ls라는 파일로 존재)



## 파일 관련 명령어



ls

- LiSt의 약자, 해당 디렉터리에 있는 파일 목록 나열

```
# ls -- 현재 디렉터리의 파일 목록을 보여줌
# ls /etc/systemd -- /etc/systemd 디렉터리의 목록을 보여줌
# ls -a -- 현재 디렉터리의 목록(숨김 파일 포함)을 보여줌
# ls -l -- 현재 디렉터리의 목록을 자세히 보여줌(l은 L의 소문자)
# ls *.conf -- 확장자가 conf인 목록을 보여줌
# ls -l /etc/systemd/b* -- /etc/systemd 디렉터리에 있는 목록 중 앞 글자가 b인 것을 자세히 보여줌
```

- \* 기호는 아무것이라는 의미
- 대개 ls -alF 라고 하면 모든 파일의 정보를 볼수 있음.



## 파일 관련 명령어



cd

- Change Directory의 약자, 디렉터리를 이동하는 명령어

```
# cd -- 현재 사용자의 홈 디렉터리로 이동. 만약 현재 사용자가 root이면 /root 디렉터리로 이동
# cd ~ubuntu -- ubuntu 사용자의 홈 디렉터리로 이동
# cd .. -- 바로 상위의 디렉터리로 이동. '..'는 현재 디렉터리의 부모 디렉터리를 의미(예: 현재 디렉터리가 /etc/systemd이면 /etc 디렉터리로 이동)
# cd /etc/systemd -- /etc/systemd 디렉터리로 이동(절대 경로)
# cd ../etc/systemd -- 상대 경로로 이동. 현재 디렉터리의 상위(..)로 이동한 후 /etc/systemd로 이동
```

- ~표시는 홈 디렉토리라는 의미로 “cd ~”을 입력하면 자신의 홈 디렉토리로 이동
- . 표시는 현재 디렉토리, .. 표시는 상위 디렉토리라는 의미



## 파일 관련 명령어

### pwd

- Print Working Directory의 약자
- 현재 디렉터리의 전체 경로를 보여줌

# pwd -- 현재 작업 중인 디렉터리의 경로 출력

```
root@server:~# pwd
/root
root@server:~# cd /etc
root@server:/etc# pwd
/etc
root@server:/etc# cd ..
root@server:/# pwd
/
root@server:/# ls
bin      dev      initrd.img      lib64      mnt      root      snap      sys      var
boot     etc      initrd.img.old  lost+found  opt      run      srv      tmp      vmlinuz
cdrom    home    lib              media      proc     sbin     swapfile  usr
```



## 파일 관련 명령어

### touch

- 크기가 0인 새 파일을 생성하거나, 이미 파일이 존재한다면 파일의 최종 수정 시간을 변경

# touch abc.txt -- 파일이 없으면 abc.txt라는 빈 파일을 생성하고, abc.txt가 있으면 파일의 최종 수정 시간을 현재 시간으로 변경

```
root@server:~# touch test.txt
root@server:~# ls -alF
합계 88
drwx----- 17 root root 4096 2월 19 18:57 ./
drwxr-xr-x 24 root root 4096 2월 17 14:37 ../
-rw----- 1 root root 1256 2월 19 16:37 .ICEauthority
-rw----- 1 root root 351 2월 19 17:44 .bash_history
-rw-r--r-- 1 root root 3147 2월 19 13:58 .bashrc
drwx----- 11 root root 4096 2월 18 16:07 .cache/
drwx----- 12 root root 4096 2월 18 17:02 .config/
drwx----- 3 root root 4096 2월 18 16:56 .gnupg/
drwxr-xr-x 3 root root 4096 2월 17 17:35 .local/
drwx----- 4 root root 4096 2월 18 16:07 .mozilla/
-rw-r--r-- 1 root root 149 2월 17 17:37 .profile
drwx----- 2 root root 4096 2월 18 16:56 .ssh/
drwx----- 6 root root 4096 2월 18 16:07 .thunderbird/
-rw-r--r-- 1 root root 73 2월 19 14:11 test.sh
-rw-r--r-- 1 root root 0 2월 19 18:57 test.txt
drwxr-xr-x 2 root root 4096 2월 17 17:38 공개/
```



## 파일 관련 명령어

### mkdir

- MaKe DIRectory의 약자, 새로운 디렉터리를 생성

```
root@server:~# mkdir test_dir
root@server:~# ls
test.sh  test_dir  다운로드  바탕화면  사진  템플릿
test.txt  공개      문서      비디오    음악
root@server:~# cd test_dir/
root@server:~/test_dir# ls
root@server:~/test_dir#
```

### rmdir

- ReMove DIRectory의 약자, 디렉터리를 삭제, 해당 디렉터리가 비어 있고 디렉터리에 대한 삭제 권한이 있어야 함. 파일이 들어 있는 디렉터리를 강제로 삭제하려면 'rm -r' 실행

```
root@server:~/test_dir# cd ..
root@server:~# ls
test.sh  test_dir  다운로드  바탕화면  사진  템플릿
test.txt  공개      문서      비디오    음악
root@server:~# rmdir test_dir/
root@server:~# ls
test.sh  test.txt  공개  다운로드  문서  바탕화면  비
root@server:~#
```





## 파일 관련 명령어

### cp

- CoPy의 약자, 파일이나 디렉터리를 복사

```
root@server:~# ls
test.sh  test.txt  공개  다운로드  문서  바탕화면  비디
root@server:~# cp test.txt test2.txt
root@server:~# ls
test.sh  test2.txt  다운로드  바탕화면  사진  템플릿
test.txt  공개      문서      비디오   음악
```

### rm

- ReMove의 약자, 파일이나 디렉터리를 삭제, 사용자에게 해당 파일이나 디렉터리의 삭제 권한이 있어야 함, root는 모든 권한을 가지고 있기 때문에 이 명령을 사용하는데 제약 없음

```
# rm abc.txt      -- 해당 파일 삭제(내부적으로 rm -f로 연결됨)
# rm -i abc.txt   -- 삭제 시 정말 삭제할지 확인하는 메시지 출력
# rm -f abc.txt   -- 삭제 시 확인하지 않고 바로 삭제(f: force)
# rm -r abc       -- abc 디렉터리와 그 하위 디렉터리를 강제로 모두 삭제. 편리하지만 주의해서 사용
                  해야 함(r: recursive)
```



## 파일 관련 명령어

### mv

- MoVe의 약자, 파일이나 디렉터리 이름을 변경하거나 다른 디렉터리로 이동할 때 사용

```
# mv abc.txt /etc/systemd/ -- abc.txt를 /etc/systemd/ 디렉터리로 이동
# mv aaa bbb ccc ddd      -- aaa, bbb, ccc 파일을 /ddd 디렉터리로 이동
# mv abc.txt www.txt      -- abc.txt의 파일명을 www.txt로 변경
```

### file

- 해당 파일이 어떤 종류의 파일인지 보여줌

```
# file /etc/systemd/system.conf -- system.conf는 텍스트 파일이므로 아스키 파일(ASCII)로 표시
# file /bin/gzip -- gzip은 실행 파일이므로 ELF 64-bit LSB executable 파일로 표시
```



## 파일 관련 명령어

### cat

- conCATenate의 약자, 파일의 내용을 화면에 출력

```
root@server:~# cat test.sh
#!/bin/sh
echo "user name: " $USER
echo "home directory: " $HOME
exit 0
root@server:~#
```

### more

- 텍스트 형식으로 작성된 파일을 페이지 단위로 화면에 출력
- Space bar를 누르면 다음 페이지로 이동

```
root@server:~# more /etc/systemd/system.conf
# This file is part of systemd.
#
# systemd is free software; you can redistribute it and/or modify it
# under the terms of the GNU Lesser General Public License as published by
# the Free Software Foundation; either version 2.1 of the License, or
# (at your option) any later version.
#
```



## 리눅스 파일의 종류

### 파일의 종류

- 파일: 관련 있는 정보들의 집합
- 리눅스는 파일을 효과적으로 관리하기 위해 디렉터리를 사용
- 리눅스에서 파일은 사용 목적에 따라 일반 파일, 디렉터리, 심볼릭 링크, 장치 파일로 구분

### 일반 파일

- 데이터를 저장하는 데 주로 사용
- 각종 텍스트 파일, 실행 파일, 이미지 파일 등 리눅스에서 사용하는 대부분의 파일은 일반 파일에 해당
- 실행 파일이나 이미지 파일의 경우 바이너리 형태로 데이터가 저장되어 바이너리 파일이라고 함



## 리눅스 파일의 종류

### 디렉터리(Directory)

- 리눅스에서는 디렉터리도 파일로 취급
- 디렉터리 파일에는 해당 디렉터리에 저장된 파일이나 하위 디렉터리에 대한 정보가 저장

### 심볼릭 링크

- 원본 파일을 대신하여 다른 이름으로 파일명을 지정한 것(윈도우의 바로가기 파일과 비슷)

### 장치 파일

- 리눅스에서는 하드디스크나 키보드 같은 각종 장치도 파일로 취급
- 주로 /dev 디렉토리에 위치함.



## 리눅스 파일의 특징

### 숨김 파일

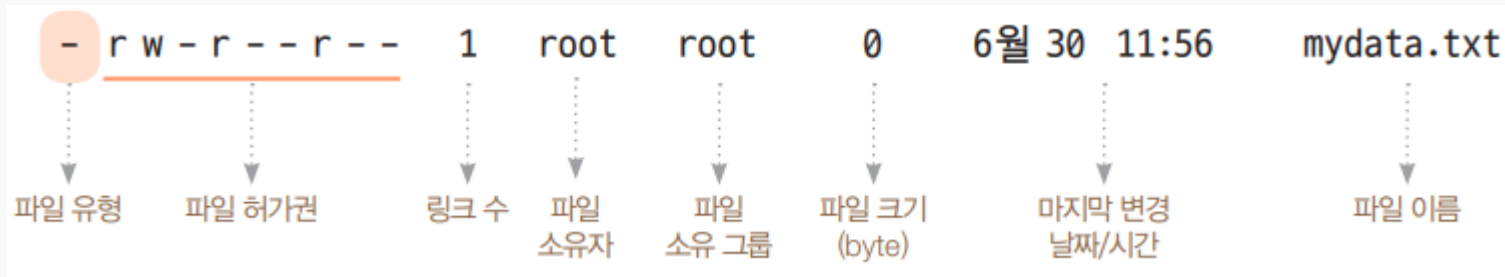
- 점(.)으로 시작하는 파일은 숨김 파일

```
root@server:~# ls -a
.          .bashrc   .local    .thunderbird  공개      비디오
..         .cache    .mozilla  test.sh        다운로드  사진
.ICEauthority .config  .profile  test.txt       문서      음악
.bash_history .gnupg   .ssh      test2.txt      바탕화면  템플릿
root@server:~#
```

- ls -a를 실행하면 그림과 같이 .bashrc, .cache등의 숨김 파일까지 출력됨.
- 숨김 파일은 주로 어떤 프로그램의 설정값이나 로그등을 저장하는데 사용됨.
- .bashrc는 배쉬셸의 설정값등을 저장.

- 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:~#
```





## 리눅스 파일의 특징

### 파일 허가권

- 윈도우의 실행파일은 \*.exe와 같은 확장자로 구분하지만 리눅스는 확장자로 구분하지 않고 파일 허가권(=권한)으로 지정
- 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		





## 리눅스 파일의 특징

### 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



## 리눅스 파일의 특징

### chmod

- 파일 허가권을 변경하는 명령어
- root 사용자 또는 해당 파일의 소유자만 실행 가능
- `chmod 777 test.sh` 명령을 실행하면 모든 사용자가 `test.sh` 파일은 읽고 쓰고 실행할 수 있음
- `chmod u+x 파일명` 명령은 소유자(User)에게 실행(eXecute) 권한을 허가하라(+)는 의미

```
root@server:~# ls -alF test.sh
-rw-r--r-- 1 root root 73  2월 19 14:11 test.sh
root@server:~# chmod 777 test.sh
root@server:~# ls -alF test.sh
-rwxrwxrwx 1 root root 73  2월 19 14:11 test.sh*
root@server:~# chmod 644 test.sh
root@server:~# ls -alF test.sh
-rw-r--r-- 1 root root 73  2월 19 14:11 test.sh
root@server:~# chmod u+x test.sh
root@server:~# ls -alF test.sh
-rwxr--r-- 1 root root 73  2월 19 14:11 test.sh*
```



## 리눅스 파일의 특징

### 파일 소유권

- 파일 소유권은 특정 사용자와 그룹이 파일에 대한 소유 권한을 가지는 것을 의미
- 그림과 같이 test.sh파일은 소유자가 root 사용자이고 소유 그룹도 root

```
root@server:~# ls -alF test.sh
-rw-r--r-- 1 root root 73  2월 19 14:11 test.sh
root@server:~# chmod 777 test.sh
root@server:~# ls -alF test.sh
-rwxrwxrwx 1 root root 73  2월 19 14:11 test.sh*
root@server:~# chmod 644 test.sh
root@server:~# ls -alF test.sh
-rw-r--r-- 1 root root 73  2월 19 14:11 test.sh
root@server:~# chmod u+x test.sh
root@server:~# ls -alF test.sh
-rwxr--r-- 1 root root 73  2월 19 14:11 test.sh*
```



## 리눅스 파일의 특징

### chown

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



## 리눅스 파일의 특징

### 파일 소유권 실습

- ① 일반 사용자인 ubuntu로 로그인, Root에서 ubuntu로 로그인하려면 login ubuntu 입력
- ② mydata.txt파일을 만든다.
- ③ 그러면 mydata.txt 파일은 소유권이 Ubuntu 사용자에게 있음.
- ④ mydata.txt파일을 /root로 복사
- ⑤ /root 디렉토리는 일반 사용자인 ubuntu 사용자에게 권한이 없으므로 복사 거부됨.

```
root@server:~# login ubuntu
암호:
마지막 로그인: 수  2월 19 19:53:56 KST 2020 일시 pts/0
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-15-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

패키지 0개를 업데이트할 수 있습니다.
0 업데이트는 보안 업데이트입니다.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
ubuntu@server:~$ touch mydata.txt
ubuntu@server:~$ ls -alF mydata.txt
-rw-rw-r-- 1 ubuntu ubuntu 0  2월 19 20:06 mydata.txt
ubuntu@server:~$ cp mydata.txt /root/
cp: '/root/mydata.txt'를 설명할 수 없음: 허가 거부
ubuntu@server:~$
```



## 리눅스 파일의 특징

### 파일 소유권 실습

- ⑥ 이 파일의 소유권을 root로 변경하기 위해 root로 로그인, root로 로그인하기 위해 login root를 하면 안되고 su 로 로그인
- ⑦ Chown명령을 통해 mydata.txt의 소유권을 root로 변경
- ⑧ /root로 mydata.txt를 복사
- ⑨ 복사됨을 확인

```
ubuntu@server:~$ login root
login: Cannot possibly work without effective root
ubuntu@server:~$ su
암호:
root@server:/home/ubuntu# ls -alF mydata.txt
-rw-rw-r-- 1 ubuntu ubuntu 0  2월 19 20:06 mydata.txt
root@server:/home/ubuntu# chown root.root mydata.txt
root@server:/home/ubuntu# ls -alF mydata.txt
-rw-rw-r-- 1 root root 0  2월 19 20:06 mydata.txt
root@server:/home/ubuntu# cp mydata.txt /root/
root@server:/home/ubuntu# ls -alF /root/mydata.txt
-rw-r--r-- 1 root root 0  2월 19 20:13 /root/mydata.txt
root@server:/home/ubuntu#
```



## 링크

### 하드 링크(hard link)와 심볼릭 링크(symbolic link)

- 파일의 링크는 하드 링크(hard link)와 심볼릭 링크(symbolic link) 또는 소프트 링크(soft link)로 구분됨
- 심볼릭 링크
  - ✓ 단순히 원본파일을 가리키도록 링크만 시켜둔 것으로 윈도우에서 흔히 사용하는 '바로 가기' 같은 것
  - ✓ 원본파일을 가리키고만 있으므로 원본파일의 크기와는 무관
  - ✓ 심볼릭링크에서는 원본파일이 삭제되어 존재하지 않을 경우에 링크파일은 깜박거리면서 링크파일의 원본파일이 없다는 것을 알려줌
  - ✓ 심볼릭 링크를 생성하는 명령은 `ln -s 원본파일 링크파일명`



## 링크

### 하드 링크(hard link)와 심볼릭 링크(symbolic link)

- 하드 링크

- ✓ 원본파일과 다른 이름으로 존재하는 동일한 파일이며 원본파일과 동일한 내용의 다른 파일
- ✓ 하드링크에서는 원본파일과 링크파일 두개가 서로 다른 파일이기 때문에 둘 중 하나를 삭제하더라도 나머지 하나는 그대로 남아 있음
- ✓ 하드링크에서는 원본파일의 내용이 변경될 경우에는 링크파일의 내용 또한 자동으로 변경됨.
- ✓ 하드 링크를 생성하는 명령은 `ln` 원본파일 링크파일명





## 링크

### 파일 링크 실습

- ① /root/test\_dir/ 디렉터리 생성
- ② test\_dir 에 들어가 gedit를 이용하여 org\_file에 hello word 입력후 저장
- ③ cat 명령으로 파일 내용 확인
- ④ 원본 org\_file의 hardlink 생성
- ⑤ 원본 org\_file의 softlink 생성
- ⑥ ls 명령으로 파일 크기 확인
- ⑦ org\_file을 삭제
- ⑧ cat 명령으로 hardlink 파일 확인
- ⑨ cat 명령으로 softlink 파일 확인



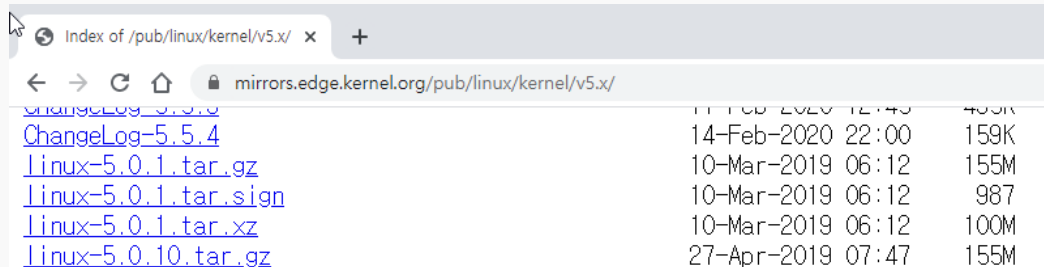
```
root@server:~# mkdir test_dir
root@server:~# cd test_dir/
root@server:~/test_dir# gedit org_file
root@server:~/test_dir# cat org_file
hello world
root@server:~/test_dir# ln org_file hardlink
root@server:~/test_dir# ln -s org_file softlink
root@server:~/test_dir# ls -alF
합계 16
drwxr-xr-x  2 root root 4096  2월 19 20:55 ./
drwx----- 18 root root 4096  2월 19 20:54 ../
-rw-r--r--  2 root root  12  2월 19 20:54 hardlink
-rw-r--r--  2 root root  12  2월 19 20:54 org_file
lrwxrwxrwx  1 root root   8  2월 19 20:55 softlink -> org_file
root@server:~/test_dir# rm org_file
root@server:~/test_dir# cat hardlink
hello world
root@server:~/test_dir# cat softlink
cat: softlink: 그런 파일이나 디렉터리가 없습니다
root@server:~/test_dir#
```



## 파일 압축 및 tar

### 리눅스 압축 파일 포맷

- 리눅스는 윈도우에서 많이 사용하는 zip 뿐만 아니라 gzip, bzip2등을 지원
- 각 압축 포맷은 그대로 명령어로 사용됨. 즉, zip파일 포맷으로 압축하는 명령어는 zip
- 리눅스에서는 파일 압축과 더불어 여러 파일을 한번에 묶는 것을 자주 사용하는데 이를 tar라고 함.
- 예를 들어 리눅스 커널 소스는 방대한 량의 파일로 구성되는데 이를 관리하기 어려우므로 tar로 묶고 압축하여 인터넷에 공개.
- Linux-5.0.1.tar.gz은 커널 버전 5.0.1로 tar로 묶고 gzip으로 압축한 파일



Index of /pub/linux/kernel/v5.x/ x +		
mirrors.edge.kernel.org/pub/linux/kernel/v5.x/		
<a href="#">ChangeLog-5.5.0</a>	11-Feb-2020 12:45	455K
<a href="#">ChangeLog-5.5.4</a>	14-Feb-2020 22:00	159K
<a href="#">linux-5.0.1.tar.gz</a>	10-Mar-2019 06:12	155M
<a href="#">linux-5.0.1.tar.sign</a>	10-Mar-2019 06:12	987
<a href="#">linux-5.0.1.tar.xz</a>	10-Mar-2019 06:12	100M
<a href="#">linux-5.0.10.tar.gz</a>	27-Apr-2019 07:47	155M



## 파일 압축 및 tar



tar

- 파일을 묶는 명령어
- 묶을 파일 이름은 보통 tar 확장자를 사용하여 구분하기 좋게 함.
- 사용법
  - ✓ `$ tar [동작] [옵션] [묶음파일이름] [filelist or dir]`
  - ✓ 파일을 묶을 때 : `tar cvf [만들 파일 이름] [묶을 파일들]`
  - ✓ 묶음을 풀 때 : `tar xvf [풀 파일 이름]`

동작	동작 내용
c	tar 파일을 생성할 때 사용 (묶음 파일 생성)
x	묶인 tar 파일을 해제할 때 사용 (묶음 파일 해제)
t	묶인 파일을 확인할 때 사용
옵션	옵션 내용
f (필수)	사용할 tar 파일을 지정
v	파일을 묶거나 해제할 때 그 과정을 자세하게 출력
z	tar 옵션과 gzip을 함께 사용
j	tar 옵션과 bzip2를 함께 사용



## 파일 압축 및 tar

### gzip

- 리눅스 표준 압축 포맷
- 파일을 gzip포맷으로 압축하거나 압축을 풀 때 사용하는 명령어
- 압축을 풀때는 옵션 -d를 사용하거나 gunzip 명령을 사용
- 압축하면 \*.gz이라는 확장자를 가지게 됨
- 사용법

✓ \$ gzip [옵션] [파일명]

옵션	동작 내용
-d	압축 해제
-c	원본 파일을 그대로 냅두고 압축
-n	1~9 까지 숫자를 지정하며 숫자가 적을수록 압축 속도는 빠르지만 압축률이 낮음
-l	압축 파일의 정보를 나타냄
-r	디렉토리를 지정시 디렉토리에 포함된 모든 파일 압축
-v	압축 또는 해제시 자세한 정보 출력



## 파일 압축 및 tar

### bzip2

- 파일을 bzip2포맷으로 압축하거나 압축을 풀 때 사용하는 명령어
- Bzip2는 gzip에 비해 높은 압축률을 가짐
- 압축을 풀때는 옵션 -d를 사용하거나 bunzip2 명령을 사용
- 압축하면 \*.bz2 라는 확장자를 가지게 됨
- 사용법

✓ \$ bzip2 [옵션] [파일명]

옵션	동작 내용
-d	압축 해제
-k	파일 압축이나 해제할 때 원본파일을 삭제하지 않음
-v	압축 또는 해제시 자세한 정보 출력
-s	속도는 느려지지만 사용 메모리 양을 줄임
-f	압축 또는 해제시 같은 이름의 파일이 있으면 덮어쓰기



## 파일 압축 및 tar



### zip

- 파일을 zip포맷으로 압축하거나 압축을 풀 때 사용하는 명령어
- Zip은 다른 압축 포맷과 다르게 암호를 설정할 수 있음.
- 압축을 풀때는 unzip 명령을 사용
- 압축하면 \*.zip 라는 확장자를 가지게 됨
- 사용법

✓ \$ zip [옵션] [파일명]

옵션	동작 내용
-r	디렉토리를 지정시 디렉토리에 포함된 모든 파일 압축
-P	압축시 암호 지정 가능



## 파일 압축 및 tar

### 실제 압축 파일 풀기

- 파일의 확장자가 .tar.gz이나 .tgz이면 tar로 먼저 묶고 gzip으로 압축되어 있는 파일을 의미함
- 이와 같이 tar+gzip을 한번에 푸는 명령어는 `$ tar xf *.tgz`
- 비슷한 방법으로 tar+bzip2는 \*.tar.bz2라는 파일명을 가짐
- tar+bzip2를 한번에 푸는 명령어는 `$ tar xf *.tar.bzp`



## 파일 압축 및 tar

### 압축 파일 실습

- ① /root/test/ 디렉터리 생성
- ② test 에 들어가 touch명령으로 3개의 파일 file1, file2, file3를 생성
- ③ 3개의 파일을 file.tar 파일로 묶음
- ④ File.tar를 gzip포맷으로 압축

```
root@server:~# mkdir test
root@server:~# cd test
root@server:~/test# touch file1 file2 file3
root@server:~/test# ls -alF
합계 8
drwxr-xr-x  2 root root 4096  2월  19 21:42 ./
drwx----- 19 root root 4096  2월  19 21:42 ../
-rw-r--r--  1 root root    0  2월  19 21:42 file1
-rw-r--r--  1 root root    0  2월  19 21:42 file2
-rw-r--r--  1 root root    0  2월  19 21:42 file3
root@server:~/test# tar -cvf file.tar file*
file1
file2
file3
root@server:~/test# ls -alF
합계 20
drwxr-xr-x  2 root root  4096  2월  19 21:42 ./
drwx----- 19 root root  4096  2월  19 21:42 ../
-rw-r--r--  1 root root 10240  2월  19 21:42 file.tar
-rw-r--r--  1 root root     0  2월  19 21:42 file1
-rw-r--r--  1 root root     0  2월  19 21:42 file2
-rw-r--r--  1 root root     0  2월  19 21:42 file3
root@server:~/test# gzip file.tar
root@server:~/test# ls
file.tar.gz file1 file2 file3
```





## 파일 압축 및 tar

### 압축 파일 해제 실습

- ① 앞에서 생성된 file.tar.gz파일을 상위 디렉토리로 복사
- ② 상위 디렉토리로 이동
- ③ File.tar.gz파일의 압축 해제

```
root@server:~/test# ls
file.tar.gz file1 file2 file3
root@server:~/test# cp file.tar.gz ..
root@server:~/test# cd ..
root@server:~# ls
file.tar.gz org_file test.sh test_dir 문서 사진
hardlink softlink test.txt 공개 바탕화면 음악
mydata.txt test test2.txt 다운로드 비디오 템플릿
root@server:~# tar xf file.tar.gz
root@server:~# ls
file.tar.gz file3 org_file test.sh test_dir 문서 사진
file1 hardlink softlink test.txt 공개 바탕화면 음악
file2 mydata.txt test test2.txt 다운로드 비디오 템플릿
```



## 파일 압축 및 tar

### 실전 문제

- ① Linux 커널의 홈페이지는 [www.kernel.org](http://www.kernel.org)이다. <https://www.kernel.org/pub/linux/kernel/v4.x/>에 들어가 `linux-4.0.1.tar.gz`을 다운로드 받으시오.
- ② 자신의 홈 디렉토리에 `kernel`이라는 디렉토리를 만들고 `kernel` 디렉토리에 `linux-4.0.1.tar.gz`을 복사하고 압축을 푸시오.



## 파일 내용 검색

### grep

- 지정된 파일내의 특정 문자나 단어를 검색해주는 명령어
- 사용법
  - ✓ `$ grep [옵션] [찾을 문자열] [파일이름]`

옵션	동작내용
-c	검색할 문자열이 속한 행이 개수를 출력한다.
-n	찾으려는 문자가 속해있는 행의 번호와 같이 출력 한다.
-r	현재 경로부터 하위경로까지 검색해서 출력을 한다.
-v	찾으려고 하는 문자가 없는 행을 출력 한다.
-w	패턴 표현식을 하나의 단어로 취급하여 검색

- ✓ 예) 현재 폴더에 있는 test.log 파일내부에 'apache' 라는 문자열 찾기
- ✓ `$ grep -r 'apache' ./test.log`



## 파일명으로 검색

### find

- 원하는 이름의 파일, 디렉토리를 검색
- 사용법
  - ✓ `$ find [찾을 위치] -name [파일이름]`
  - ✓ 예) 최상위 폴더부터 하위로 'apache' 로 시작하는 파일 찾기
  - ✓ `$ find / -name 'apache*'`
  - ✓ 예) usr 폴더부터 하위로 'apa'로 시작하는 파일 찾기
  - ✓ `$ find /usr/ -name 'apa*'`

Q & A

Thank You