



운영체제 실습

## 리눅스 기본 명령어

2018.3

표월성

[wspyo74@naver.com](mailto:wspyo74@naver.com)

[cherub.sungkyul.ac.kr](mailto:cherub.sungkyul.ac.kr)



### 목 차

#### I. 기본 명령어

1. 시스템 정보
2. 파일 및 디렉토리 명령어

# 시스템 정보

## 1. 시스템 정보 출력

### 시스템 정보 출력

#### ◆ uname

- 시스템 정보 출력

옵션	내용	비고
-a	모든 정보 출력	우분투(Ubuntu) 라즈비안
-m	하드웨어 종류를 출력	
-n	호스트 이름을 출력	
-o	운영체제 출력	
-r	운영체제 릴리즈 정보 출력	
-s	운영체제 이름 출력	
-v	운영체제 버전 출력	

```
pi@raspberrypi:~$ uname
Linux
pi@raspberrypi:~$ uname -a
Linux raspberrypi 4.9.59-v7+ #1047 SMP Sun Oct 29 12:19:23 GMT 2017 armv7l GNU/Linux
pi@raspberrypi:~$ uname -m
armv7l
pi@raspberrypi:~$ uname -n
raspberrypi
pi@raspberrypi:~$ uname -o
GNU/Linux
pi@raspberrypi:~$ uname -r
4.9.59-v7+
pi@raspberrypi:~$ uname -s
Linux
pi@raspberrypi:~$ uname -v
#1047 SMP Sun Oct 29 12:19:23 GMT 2017
pi@raspberrypi:~$
```

# 파일 및 디렉토리 관련 명령어

## 2. 파일 및 디렉토리 관련 명령어

### 파일 및 디렉토리

#### ◆ 파일(File)

- 데이터를 저장하기 위해 사용되는 객체 (텍스트파일, 실행파일(실행가능한 바이너리 파일), 그림파일, 동영상파일 등)

#### ◆ 디렉토리(Directory)

- 윈도우의 '폴더(Folder)' = 디렉토리
- 파일이나 디렉토리를 저장하기 위해 사용되는 객체
- 디렉토리도 파일

#### ◆ 리눅스의 파일 및 디렉토리의 Naming Rules

- 파일과 디렉토리 이름에 / 을 사용할 수 없다. ( 슬래쉬(/)는 경로명 구분자로 사용)
- 알파벳, 숫자, 불임표(-), 밑줄(\_), 마침표(.)만 사용
- 공백문자, \*, |, ", ', @, #, \$, %, ^, & 등 사용불가
- 대소문자 구별
- 마침표(.)로 시작하는 파일과 디렉토리는 숨김파일로 간주

예1)	예2)	예3)
test.txt	&test	myhome/./test
hello.c	my home	bad/name
game11		

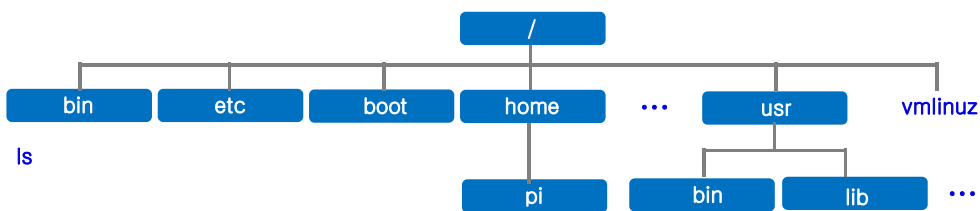
## 2. 파일 및 디렉토리 관련 명령어

## 리눅스 디렉토리 구조

## ◆ 리눅스의 디렉토리 구조

- 역트리 구조
- 최상위 디렉토리 : 루트 디렉토리 (/)
- 디렉토리 구분자 : 슬래쉬(/) 기호 사용 ( windows의 경우 : 역슬래쉬(\) 사용)

```
File Edit Tabs Help
pi@raspberrypi:~$ cd /
pi@raspberrypi:/ $ ls
bin  etc  initrd.img.old  media  proc  sbin  tmp  vmlinuz
boot home  lib  mnt  root  srv  usr  vmlinuz.old
dev  initrd.img  lost+found  opt  run  sys  var
pi@raspberrypi:/ $
```



## 2. 파일 및 디렉토리 관련 명령어

## 리눅스 디렉토리 구조

## ◆ 리눅스의 디렉토리 구조

디렉토리	내용				
/	<ul style="list-style-type: none"> <li>• 최상위 디렉토리 = "루트 디렉토리"라고 함</li> <li>• 절대경로를 표시할 때, 시작점으로 사용 됨</li> </ul>				
/bin	<ul style="list-style-type: none"> <li>• 일반 사용자들이 사용하는 명령어(실행파일)들이 있는 디렉토리</li> <li>• 시스템을 운영하는데 사용되는 기본적인 명령어</li> </ul> <table border="1"> <tr> <td>페도라(Fedora)</td><td>/bin을 /usr/bin 디렉토리의 심볼릭 링크로 유지</td></tr> <tr> <td>우분트(ubuntu)</td><td>/bin과 /usr/bin을 나누어 관리함</td></tr> </table>	페도라(Fedora)	/bin을 /usr/bin 디렉토리의 심볼릭 링크로 유지	우분트(ubuntu)	/bin과 /usr/bin을 나누어 관리함
페도라(Fedora)	/bin을 /usr/bin 디렉토리의 심볼릭 링크로 유지				
우분트(ubuntu)	/bin과 /usr/bin을 나누어 관리함				
/etc	<ul style="list-style-type: none"> <li>• 시스템 환경설정을 위한 각종 파일들이 존재</li> </ul>				
/media	<ul style="list-style-type: none"> <li>• 탈부착이 가능한 외부장치를 연결(마운트)하는데 사용되는 디렉토리 (DVD, CD-ROM, USB 등)</li> </ul>				
/proc	<ul style="list-style-type: none"> <li>• 시스템 정보 디렉토리</li> <li>- 프로세스 및 커널 관련 정보를 가지고 있는 가상 파일 시스템( 대부분 읽기 전용)</li> <li>- <code>cpuinfo</code> : CPU 정보</li> <li>- <code>meminfo</code> : 메모리 정보</li> </ul>				
/sbin	<ul style="list-style-type: none"> <li>• 시스템 관리와 관련된 명령어들이 있는 디렉토리</li> <li>- 시스템 점검 및 복구 명령, 네트워크 인터페이스 설정 명령, 시스템 초기 및 종료 명령, 커널 모듈 등</li> </ul>				

## 2. 파일 및 디렉토리 관련 명령어

## 리눅스 디렉토리 구조

## ◆ 리눅스의 디렉토리 구조

디렉토리	내용
<b>/tmp</b>	<ul style="list-style-type: none"> <li>시스템 사용 중에 발생하는 임시 데이터가 저장</li> <li>이 디렉토리에 있는 파일들은 재시작하면 모두 삭제됨</li> </ul>
<b>/boot</b>	<ul style="list-style-type: none"> <li>리눅스 부트로더(bootloader : GRUB), 커널 이미지 및 부팅 관련 파일과 디렉토리가 저장된 디렉토리</li> </ul>
<b>/home</b>	<ul style="list-style-type: none"> <li>사용자의 홈 디렉토리가 저장되는 디렉토리</li> </ul>
<b>/lib</b>	<ul style="list-style-type: none"> <li>공유 라이브러리 디렉토리</li> <li>- 커널 모듈과 라이브러리 파일들이 존재하는 디렉토리</li> </ul>
<b>/mnt</b>	<ul style="list-style-type: none"> <li>파일 시스템을 임시로 마운트하는 디렉토리</li> </ul>
<b>/root</b>	<ul style="list-style-type: none"> <li>root계정의 홈 디렉토리</li> <li>root = 시스템 관리자 = 루트 관리자 = 최고 권한 관리자 등 (주의) 루트(/) 디렉토리와 혼동하지 말 것</li> </ul>
<b>/srv</b>	<ul style="list-style-type: none"> <li>FTP나 WEB 등 시스템에서 제공하는 서비스의 데이터가 저장</li> </ul>
<b>/dev</b>	디바이스(장치) 파일들이 존재하는 디렉토리 (하드디스크, CD-ROM 등)
<b>/lost+found</b>	<ul style="list-style-type: none"> <li>파일시스템에 문제가 발생하여 복구할 경우, 문제가 되는 파일들이 저장되는 디렉토리</li> <li>보통 비어있음</li> </ul>

## 2. 파일 및 디렉토리 관련 명령어

## 리눅스 디렉토리 구조

## ◆ 리눅스의 디렉토리 구조

디렉토리	내용
<b>/usr</b>	<ul style="list-style-type: none"> <li>사용자 디렉토리(Unix System Resource의 약자)</li> <li>기본 실행파일과 라이브러리 파일, 헤더파일 등</li> <li>일반 사용자들이 사용하는 파일들이 존재</li> </ul>
	<b>/usr/bin</b> 사용자 명령어가 존재하는 디렉토리
	<b>/usr/include</b> 기본 C라이브러리 헤더파일과 라이브러리 헤더파일들이 있는 디렉토리
	<b>/usr/lib</b> 라이브러리 파일들이 존재하는 디렉토리
	<b>/usr/bin</b> 과 <b>/usr/sbin</b> 에 있는 실행파일을 위한 라이브러리
	<b>/usr/local</b> 프로그램이 설치되는 디렉토리
	모든 프로그램이 반드시 여기에 설치되는 것은 아님
	<b>/usr/man</b> 매뉴얼 디렉토리
	<b>/usr/sbin</b> 관리자가 사용하는 명령어가 있는 디렉토리
<b>/usr/share</b>	아키텍처에 의존하지 않는 파일들이 존재하는 디렉토리
	리눅스 소스를 가지고 있는 디렉토리
<b>/opt</b>	추가적인 패키지(프로그램)가 설치되는 디렉토리
<b>/run</b>	실행중인 서비스와 관련된 파일이 저장
<b>/sys</b>	리눅스 커널과 관련된 파일이 있는 디렉토리
<b>/var</b>	<ul style="list-style-type: none"> <li>가변 자료 저장 디렉토리</li> <li>시스템 운영 중에 시스템 자료 데이터가 변경될 때 변경된 자료들이 저장되는 디렉토리</li> <li>- 각 종 로그, 메일을 받아 저장하는 임시 디렉토리 등</li> </ul>

## 2. 파일 및 디렉토리 관련 명령어

## pwd

◆ 현재 작업중인 디렉토리 (Present Working Directory)

- \$pwd

```
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $ cd /
pi@raspberrypi:/ $ pwd
/
pi@raspberrypi:/ $
```

## cd

◆ cd (change directory)

- 디렉토리 변경 (현재 작업디렉토리 위치를 변경하는 것)

- \$cd [이동할 디렉토리명]

예1) \$cd /

예2) \$cd /usr

예3) \$cd /home/pi

## 2. 파일 및 디렉토리 관련 명령어

## ls

◆ ls (List)

✓ 디렉토리 내의 파일 및 디렉토리 목록 출력

✓ Windows의 dir명령과 유사

✓ 사용법

- \$ls [옵션] [디렉토리] // 예) \$ls -al /boot

옵션	설 명	옵션	설 명						
-a	모든 파일을 표시 숨김파일(hidden file) 포함 - '.'으로 시작하는 파일, 현재디렉토리( . )와 상위디렉토리( .. )	-F	파일 형식(속성)을 표시 <table><tr><td>/</td><td>디렉토리</td></tr><tr><td>*</td><td>실행파일</td></tr><tr><td>@</td><td>심볼릭 링크 파일</td></tr></table>	/	디렉토리	*	실행파일	@	심볼릭 링크 파일
/	디렉토리								
*	실행파일								
@	심볼릭 링크 파일								
-l	파일에 대한 자세한 정보를 표시 (파일의 종류, 퍼미션, 소유권, 크기, 최종 변경일자 등)								
-i	아이노드(inode) 번호 표시								
-m	목록을 가로로 표시	-R	현재 디렉토리 및 하위 디렉토리의 파일들을 모두 출력						
-t	파일 시간을 기준으로 정렬하여 표시 (최근 파일이 제일 먼저 출력)	-S	파일 크기가 큰 것부터 정렬하여 출력						
-A	목록에서 현재디렉토리('.')과 상위디렉토리('..')를 제외	-1	한 줄에 1개의 파일을 출력						

## 2. 파일 및 디렉토리 관련 명령어

ls

◆ ls (List)

```

pi@raspberrypi:~ $ ls
Desktop  Downloads  Pictures  Templates  python_games  test.txt
Documents Music      Public    Videos     test

pi@raspberrypi:~ $ ls -a
.         .dbus      .vnc      Public
..        .gconf     .xsession-errors  Templates
.Xauthority .gstreamer-0.10 .xsession-errors.old Videos
.bash_history .local     Desktop   python_games
.bash_logout .pki       Documents test
.bashrc      .profile   Downloads test.txt
.cache       .themes    Music
.config      .viminfo   Pictures

pi@raspberrypi:~ $ ls -li
256016 Desktop  256020 Music    256018 Templates 385164 test
130070 Documents 256021 Pictures 256022 Videos 128188 test.txt
256017 Downloads 256019 Public    130071 python_games

pi@raspberrypi:~ $ ls -lm
Desktop, Documents, Downloads, Music, Pictures, Public, Templates, Videos,
python_games, test, test.txt
pi@raspberrypi:~ $

```

inode 번호

.	현재 디렉토리
..	상위 디렉토리
''으로 시작하는 파일	숨김파일

## 2. 파일 및 디렉토리 관련 명령어

ls

◆ ls (List)

```

pi@raspberrypi:~ $ ls -l
total 36
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Desktop
drwxr-xr-x 5 pi pi 4096 Nov 29 02:22 Documents
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Downloads
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Music
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Pictures
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Public
drwxr-xr-x 2 pi pi 4096 Nov 29 02:22 python_games
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Templates
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Videos
pi@raspberrypi:~ $

pi@raspberrypi:~ $ ls -al
total 96
drwxr-xr-x 17 pi pi 4096 Mar 24 07:14 .
drwxr-xr-x 3 root root 4096 Nov 29 01:22 ..
-rw-r--r-- 1 pi pi 19 Nov 29 02:57 .bash_history
-rw-r--r-- 1 pi pi 220 Nov 29 01:22 .bash_logout
-rw-r--r-- 1 pi pi 3523 Nov 29 01:22 .bashrc
drwxr-xr-x 5 pi pi 4096 Nov 29 02:56 .cache
drwx----- 8 pi pi 4096 Nov 29 02:56 .config
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Desktop
drwxr-xr-x 5 pi pi 4096 Nov 29 02:22 Documents
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Downloads
drwx----- 3 pi pi 4096 Nov 29 02:56 .gnupg
drwxr-xr-x 3 pi pi 4096 Nov 29 02:22 .local
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Music
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Pictures
-rw-r--r-- 1 pi pi 675 Nov 29 01:22 .profile
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Public
drwxr-xr-x 2 pi pi 4096 Nov 29 02:22 python_games
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Templates
drwxr-xr-x 3 pi pi 4096 Nov 29 02:56 .themes
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Videos
drwx----- 3 pi pi 4096 Mar 24 07:14 .vnc
-rw-r--r-- 1 pi pi 56 Nov 29 02:57 .Xauthority
-rw-r--r-- 1 pi pi 3954 Nov 29 02:57 .xsession-errors
-rw-r--r-- 1 pi pi 3954 Nov 29 02:56 .xsession-errors.old
pi@raspberrypi:~ $

```



## 2. 파일 및 디렉토리 관련 명령어

```

ls
pi@raspberrypi:~$ ls -al
total 96
drwxr-xr-x 17 pi pi 4096 Mar 24 07:14 .
drwxr-xr-x 3 root root 4096 Nov 29 01:22 ..
-rw-r--r-- 1 pi pi 19 Nov 29 02:57 .bash_history
-rw-r--r-- 1 pi pi 220 Nov 29 01:22 .bash_logout
-rw-r--r-- 1 pi pi 3523 Nov 29 01:22 .bashrc
drwxr-xr-x 5 pi pi 4096 Nov 29 02:56 .cache
drwxr-xr-x 8 pi pi 4096 Nov 29 02:56 .config
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Desktop
drwxr-xr-x 5 pi pi 4096 Nov 29 02:22 Documents
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Downloads
drwxr-xr-x 3 pi pi 4096 Nov 29 02:56 .gnupg
drwxr-xr-x 3 pi pi 4096 Nov 29 02:22 .local
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Music
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Pictures
-rw-r--r-- 1 pi pi 675 Nov 29 01:22 .profile
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Public
drwxr-xr-x 2 pi pi 4096 Nov 29 02:22 python_games
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Templates
drwxr-xr-x 3 pi pi 4096 Nov 29 02:56 .themes
drwxr-xr-x 2 pi pi 4096 Nov 29 02:56 Videos
drwxr-xr-x 3 pi pi 4096 Mar 24 07:14 .vnc
-rw-r--r-- 1 pi pi 56 Nov 29 02:57 .Xauthority
-rw-r--r-- 1 pi pi 3954 Nov 29 02:57 .xsession-errors
-rw-r--r-- 1 pi pi 3954 Nov 29 02:56 .xsession-errors.old

```

◆ ls (List)  
 소유권 (파일 소유자, 파일소유그룹)  
 링크 수  
 최종 수정된 날짜 및 시간  
 파일 및 디렉토리 이름  
 파일 유형(type)  
 허가권(permission)  
 파일 크기(byte)

## 2. 파일 및 디렉토리 관련 명령어

```

ls
pi@raspberrypi:~$ ls -l /dev
total 0
crw-rw-rw- 1 root root 10, 235 Nov 29 02:57 autofs
drwxr-xr-x 2 root root 580 Nov 29 02:57 block
crw-rw-rw- 1 root root 10, 234 Nov 29 02:57 btrfs-control
drwxr-xr-x 3 root root 60 Jan 1 1970 bus
crw-rw-rw- 1 root root 10, 63 Nov 29 02:57 cachefiles
drwxr-xr-x 2 root root 2660 Nov 29 02:57 char
crw-rw-rw- 1 root root 5, 1 Nov 29 02:57 console
crw-rw-rw- 1 root root 10, 62 Nov 29 02:57 cpu_dma_latency
crw-rw-rw- 1 root root 10, 203 Nov 29 02:57 cuse
drwxr-xr-x 7 root root 140 Nov 29 02:57 disk
crw-rw-rw- 1 root video 29, 0 Nov 29 02:57 fb0
lrwxrwxrwx 1 root root 13 Nov 3 2016 fd -> /proc/self/fd
crw-rw-rw- 1 root root 1, 7 Nov 29 02:57 full
crw-rw-rw- 1 root root 10, 229 Nov 29 02:57 fuse
crw-rw-rw- 1 root gpio 254, 0 Nov 29 02:57 gpiochip0
crw-rw-rw- 1 root gpio 254, 1 Nov 29 02:57 gpiochip1
crw-rw-rw- 1 root gpio 254, 2 Nov 29 02:57 gpiochip2
crw-rw-rw- 1 root gpio 244, 0 Nov 29 02:57 gpiomem
crw-rw-rw- 1 root root 248, 0 Nov 29 02:57 hidraw0
crw-rw-rw- 1 root root 248, 1 Nov 29 02:57 hidraw1
crw-rw-rw- 1 root root 248, 2 Nov 29 02:57 hidraw2
crw-rw-rw- 1 root root 10, 183 Nov 29 02:57 hwrng
lrwxrwxrwx 1 root root 25 Nov 3 2016 initctl -> /run/systemd/initctl/fifo
drwxr-xr-x 4 root root 180 Nov 29 02:57 input
crw-rw-rw- 1 root root 1, 11 Nov 29 02:57 kmsg
lrwxrwxrwx 1 root root 28 Nov 3 2016 log -> /run/systemd/journal/dev-log
brw-rw-rw- 1 root disk 7, 0 Nov 29 02:57 loop0

```

◆ 파일 유형(type)



## 2. 파일 및 디렉토리 관련 명령어

ls

## ◆ 파일 유형(type)

문자	파일유형	비고	
-	일반파일	실행파일, 스크립트, 이미지파일, 텍스트파일, 설정파일, 아카이브 파일, 압축파일 등	
d	디렉토리	-	
b	블록 디바이스 (Block Device)	hd?	IDE 하드디스크
		sd?	SCSI, UBS, SATA
		fd0	플로피디스크
		ram0	램디스크
		cdrom sr0	CD-ROM
c	문자 디바이스 (Character Device)	보통 입/출력 장치	
		tty, pts	터미널 장치 (로컬, 원격터미널)
		lp0	프린터 장치
		기타	키보드, 마우스 등
l	링크파일	다른 파일명으로 연결해 주는 파일	
p	파이프 파일	프로세스 간 통신을 위한 특수파일	
s	소켓(socket)	네트워크 통신을 위한 특수파일	

## 2. 파일 및 디렉토리 관련 명령어

ls

## ◆ 와일드카드 문자 (Wildcard Character)

✓ \*: 모든 문자

✓ ?: 하나의 문자

```

pi@raspberrypi:~ $ cd /bin
pi@raspberrypi:/bin $
pi@raspberrypi:/bin $ ls m*
machinectl mkdir mknod mktemp modeline2fb more mount mountpoint mt mt-gnu mv
pi@raspberrypi:/bin $
pi@raspberrypi:/bin $ ls mo*
modeline2fb more mount mountpoint
pi@raspberrypi:/bin $
pi@raspberrypi:/bin $ ls m?
mt mv
pi@raspberrypi:/bin $
pi@raspberrypi:/bin $ ls m??
ls: cannot access m??: 그런 파일이나 디렉터리가 없습니다
pi@raspberrypi:/bin $
pi@raspberrypi:/bin $ ls m???
more
pi@raspberrypi:/bin $

```

## 2. 파일 및 디렉토리 관련 명령어

## 경로(path)

## ◆ 경로(path)

- 디렉토리 계층 구조에 있는 특정 파일이나 디렉토리의 위치
- 경로를 구분하는 구분자 : /

루트 디렉토리

/bin/ls

디렉토리 구분자

## ◆ 절대경로 (Absolute Path)

- ✓ / (루트 디렉토리)를 기준으로 한 경로
- ✓ 반드시 / (루트 디렉토리)로 시작한다.
- ✓ / (루트 디렉토리)로 부터 시작하여, 특정 파일이나 디렉토리 위치까지 이동하면서, 거치게 되는 모든 중간 디렉토리의 이름을 표시
- ✓ 특정 위치를 가리키는 절대 경로명은 항상 동일

```
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $ ls /
bin  dev  home  lost+found  mnt  proc  run  srv  tmp  var
boot  etc  lib  media  opt  root /sbin  sys  usr
pi@raspberrypi:~ $
pi@raspberrypi:~ $ cd /usr/bin
pi@raspberrypi:/usr/bin $ pwd
/usr/bin
pi@raspberrypi:/usr/bin $
```

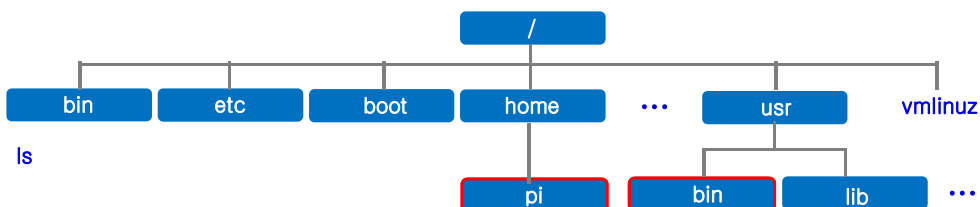
## 2. 파일 및 디렉토리 관련 명령어

## 경로(path)

## ◆ 상대경로 (Relative Path)

- ✓ 현재 작업 디렉토리를 기준으로 한 경로
- ✓ 상대 경로명은 현재 디렉토리가 어디냐에 따라 달라짐(상대경로명은 항상 달라짐)
- ✓ / (루트 디렉토리) 이외의 문자로 시작
- ✓ 현재 디렉토리를 기준으로, 서브 디렉토리로 내려가면서, 서브 디렉토리의 이름을 추가
- ✓ 현재 디렉토리를 기준으로 상위 디렉토리로 가려면, ..(마침표 2개)를 사용

```
pi@raspberrypi:/usr/bin $ pwd
/usr/bin
pi@raspberrypi:/usr/bin $
pi@raspberrypi:/usr/bin $
pi@raspberrypi:/usr/bin $ pwd
/usr/bin
pi@raspberrypi:/usr/bin $ cd ..
pi@raspberrypi:/usr $ pwd
/usr
pi@raspberrypi:/usr $ cd ../home/pi
pi@raspberrypi:~ $
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $
```



## 2. 파일 및 디렉토리 관련 명령어

## 경로(path)

## ◆ 홈 디렉토리(~)

- ✓ 각 사용자에게 할당된 디렉토리
- ✓ 처음 사용자 계정을 만들 때 지정
- \$cd ~pi
- \$cd ~
- \$cd

```
pi@raspberrypi:~ $ cd /usr/bin
pi@raspberrypi:/usr/bin $
pi@raspberrypi:/usr/bin $ pwd
/usr/bin
pi@raspberrypi:/usr/bin $ cd ~
pi@raspberrypi:~ $
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $
pi@raspberrypi:~ $ cd /dev
pi@raspberrypi:/dev $
pi@raspberrypi:/dev $ pwd
/dev
pi@raspberrypi:/dev $ cd
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $
```

## 2. 파일 및 디렉토리 관련 명령어

## mkdir

## ◆ 디렉토리 생성 (make directory)

## ✓ 하나의 Directory 생성

\$mkdir <생성할 디렉토리명>

ex) \$mkdir test

// test 디렉토리 생성

## ✓ 2개 이상 Directory 생성

\$mkdir <디렉토리1> <디렉토리2> ...

ex) \$mkdir dir1 dir2

// dir1디렉토리와 dir2디렉토리 생성

## ✓ 하위 디렉토리까지 한번에 생성

\$mkdir -p <디렉토리명>

ex) \$mkdir -p ./infolinux/201ho/3com

// 한번에 하위 디렉토리까지 한번에 생성

```
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $ ls
Desktop Downloads Pictures Templates cap
Documents Music Public Videos python_games
pi@raspberrypi:~ $
pi@raspberrypi:~ $ mkdir test
pi@raspberrypi:~ $ ls
Desktop Downloads Pictures Templates cap python_games test
Documents Music Public Videos python_games
pi@raspberrypi:~ $
pi@raspberrypi:~ $ mkdir dir1 dir2
pi@raspberrypi:~ $ ls
Desktop Downloads Pictures Templates cap dir2 test
Documents Music Public Videos dir1 python_games
pi@raspberrypi:~ $
pi@raspberrypi:~ $ mkdir -p ./infolinux/201ho/3com
pi@raspberrypi:~ $ ls
Desktop Downloads Pictures Templates cap dir2 python_games
Documents Music Public Videos dir1 infolinux test
pi@raspberrypi:~ $ ls -R ./infolinux
./infolinux:
201ho
./infolinux/201ho:
3com
./infolinux/201ho/3com:
pi@raspberrypi:~ $
```

## 2. 파일 및 디렉토리 관련 명령어

## cp

## ◆ 파일 or 디렉토리 복사 (copy)

## ✓ 파일 복사

\$cp [옵션] &lt;원본파일&gt; &lt;대상파일&gt;

예) \$cp ./exam.c ./temp.c

// 현재 디렉토리에 있는 exam.c파일을  
현재 디렉토리에 temp.c로 복사

\$cp [옵션] &lt;원본파일&gt; &lt;원본파일2&gt; &lt;대상디렉토리&gt;

예) \$cp ./exam.c ./temp.c ./sss

// 현재 디렉토리에 있는 exam.c와 temp.c파일을  
sss 디렉토리로 복사

```
pi@raspberrypi:~/test $ pwd
/home/pi/test
pi@raspberrypi:~/test $ vi exam.c
pi@raspberrypi:~/test $ ls
exam.c
pi@raspberrypi:~/test $ cp ./exam.c ./temp.c
pi@raspberrypi:~/test $ ls
exam.c  temp.c
pi@raspberrypi:~/test $ mkdir sss
pi@raspberrypi:~/test $ ls
exam.c  sss  temp.c
pi@raspberrypi:~/test $ cp ./exam.c ./temp.c ./sss
pi@raspberrypi:~/test $ ls ./sss
exam.c  temp.c
pi@raspberrypi:~/test $ ls
exam.c  sss  temp.c
pi@raspberrypi:~/test $ cp -r sss ttt
pi@raspberrypi:~/test $ ls
exam.c  sss  temp.c  ttt
pi@raspberrypi:~/test $
```

## ✓ 디렉토리 복사

\$cp [옵션] &lt;원본디렉토리&gt; &lt;대상디렉토리&gt;

예) \$cp -r dir1 dir2

// dir1디렉토리와 dir2디렉토리 복사

옵션	설명	비고
-a	원본파일의 속성, 링크정보를 유지하면서 복사	-dpr 과 같은 역할
-b	이미 파일이 존재할 때, 복사할 대상을 덮어쓰거나 지울 때를 대비하여 백업파일을 만든다.	파일 이름
-d	심볼릭 링크를 보존	
-f	대상 디렉토리에 이미 파일이 존재하는 경우, 그냥 덮어쓴다.	
-i	파일이 이미 존재할 경우, 덮어 쓸 것인지 물어봄	
-P	원본파일의 소유권(소유권, 그룹), 퍼미션, 허용시간을 보존한 채로 복사	
-r	서브 디렉토리까지 모두 복사	디렉토리 복사
-s	디렉토리가 아닌 경우, 파일을 복사하지 않고 심볼릭 링크를 생성	
-v	현재 진행되는 작업을 표시해 준다.	

## 2. 파일 및 디렉토리 관련 명령어

## rm

## ◆ 파일 or 디렉토리 삭제 (remove)

## ✓ 파일 및 디렉토리 삭제

\$rm [옵션] &lt;삭제할 파일&gt;

예) \$rm ./exam.c

// 현재 디렉토리에 있는 exam.c파일을 삭제

\$rm [옵션] &lt;삭제할 파일1&gt; &lt;삭제할 파일2&gt; ...

예) \$rm -rf ./exam.c ./temp.c ./sss

// 현재 디렉토리에 있는 exam.c와 temp.c파일과 sss 디렉토리를 강제로 삭제

## ✓ 비어있는 디렉토리 삭제 (rmdir)

\$rmdir &lt;삭제할 빈 디렉토리&gt;

예) \$rmdir dir1

// 비어있는 dir1디렉토리 삭제

옵션	설명	비고
-a	디렉토리 삭제 (디렉토리 내에 파일이 존재하는 경우, 삭제불가)	
-i	삭제 할 것인지 물어봄	
-f	강제로 삭제	삭제여부 물어보지 않음
-r	하위 디렉토리를 포함한 파일들을 모두 삭제	
-v	삭제되기 전에 파일들을 보여줌	

## 2. 파일 및 디렉토리 관련 명령어

## rm

## ◆ 파일 or 디렉토리 삭제 (remove)

```

pi@raspberrypi:~/test $ ls
exam.c  sss  temp.c  ttt
pi@raspberrypi:~/test $ rm ./exam.c
pi@raspberrypi:~/test $ ls
sss  temp.c  ttt
pi@raspberrypi:~/test $ rm -r ttt
pi@raspberrypi:~/test $ ls
sss  temp.c
pi@raspberrypi:~/test $ rm sss
rm: cannot remove `sss': 디렉터리입니다
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ rm -rf temp.c sss
pi@raspberrypi:~/test $ ls
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ cd ..
pi@raspberrypi:~ $ ls
Desktop  Downloads  Pictures  Templates  cap  dir2  python_games
Documents Music    Public    Videos    dir1 infolinux test
pi@raspberrypi:~ $ rmdir test
pi@raspberrypi:~ $ ls
Desktop  Downloads  Pictures  Templates  cap  dir2  python_games
Documents Music    Public    Videos    dir1 infolinux
pi@raspberrypi:~ $

```

## 2. 파일 및 디렉토리 관련 명령어

## mv

## ◆ 파일 or 디렉토리 이동 (move) / 이름변경(rename)

## ✓ 이름변경(rename)

\$mv [옵션] <원본파일> <이름변경파일>

예) \$mv ./exam.c ./temp.c // 현재 디렉토리에 있는 exam.c파일의 이름을 temp.c파일로 이름변경

\$rm [옵션] <원본디렉토리> <변경디렉토리>

예) \$mv dir1 dir2 // 현재 디렉토리에 있는 dir1디렉토리의 이름을 dir2로 변경

## ✓ 이동(move)

\$mv [옵션] <원본파일> <디렉토리>

예) \$mv ./exam.c ./dir2 // 현재 디렉토리에 있는 exam.c파일을 dir2디렉토리로 이동

옵션	설명	비고
-f	(대상파일이 있어도) 강제로 삭제하고 이동	삭제여부 물어보지 않음
-i	대상파일이 있는 경우, 덮어 쓸 것인가를 물어봄	
-v	파일 옮기기 전의 파일들을 보여줌	

## 2. 파일 및 디렉토리 관련 명령어

## touch

## ◆ 크기가 0인 파일 생성

\$touch &lt;파일명&gt;

예) \$touch tempfile

```
pi@raspberrypi:~/test $ ls
pi@raspberrypi:~/test $ touch tempfile
pi@raspberrypi:~/test $ ls -l
합계 0
-rw-r--r-- 1 pi pi 0 4월 20 16:42 tempfile
pi@raspberrypi:~/test $
```

## cat

## ◆ 텍스트(txt)파일의 내용 출력

\$cat &lt;txt파일&gt;

- 예) \$cat ./test.txt

// test.txt파일의 내용 보기

## ◆ 텍스트(txt)파일 생성

1. \$cat &gt; txt파일

- 예) \$cat &gt; test.txt

// test.txt파일 생성

2. \$cat &gt;&gt; txt파일

- 예) \$cat &gt;&gt; test.txt

// test.txt파일 생성

```
pi@raspberrypi:~ $ cat > test.txt
12345
67890
pi@raspberrypi:~ $ cat test.txt
12345
67890
pi@raspberrypi:~ $
```

&lt;Ctrl&gt;+d

## 2. 파일 및 디렉토리 관련 명령어

## 리다이렉션(Redirection)

## ◆ 입력 리다이렉션(&gt;)

&gt; : 파일 생성 / 덮어쓰기

- 파일 생성

- 파일 내용 덮어쓰기

예) \$cat &gt; test.txt (test.txt파일이 존재하는 경우, 내용을 덮어쓰)

&gt;&gt; : 파일 생성 / 내용 추가

- 파일 생성 : 파일이 존재하는 경우

- 내용 추가 : 파일이 존재하지 않는 경우

예) \$cat &gt;&gt; test.txt

## ◆ 출력 리다이렉션(&lt;)

- 텍스트 파일 내용 출력

예) \$cat &lt; test.txt

## 2. 파일 및 디렉토리 관련 명령어

## 리다이렉션(Redirection)

```

pi@raspberrypi:~/test $ vi test.txt
pi@raspberrypi:~/test $ cat test.txt
Hi! Nice to meet you^^
My name is PYO WOL SEONG

pi@raspberrypi:~/test $ cat >> test.txt
E-mail : wspyo74@naver.com
pi@raspberrypi:~/test $ cat test.txt
Hi! Nice to meet you^^
My name is PYO WOL SEONG

E-mail : wspyo74@naver.com
pi@raspberrypi:~/test $ cat -b test.txt
 1 Hi! Nice to meet you^^
 2 My name is PYO WOL SEONG

 3 E-mail : wspyo74@naver.com
pi@raspberrypi:~/test $ cat -n test.txt
 1 Hi! Nice to meet you^^
 2 My name is PYO WOL SEONG
 3
 4
 5 E-mail : wspyo74@naver.com
pi@raspberrypi:~/test $

```

1. vi로 test.txt파일 생성

2. cat명령으로 test.txt파일 내용 보기

3. cat명령과 리다이렉션(>>)을 활용하여 test.txt파일에 내용 추가  
- 저장 : <Ctrl> + d

4. cat명령으로 test.txt파일에서 추가한 내용 보기

-b : 라인 번호 표시 (빈 줄 제외)

-n : 라인 번호 표시 (빈 줄 까지)

## 2. 파일 및 디렉토리 관련 명령어

## more

◆ 파일의 내용을 화면 단위로 분할 출력

✓ 한 줄씩 출력 : &lt;enter&gt; 키

✓ 한 화면 단위 출력 : &lt;space bar&gt; 키

✓ \$more &lt;파일명&gt;

- 예) \$more test.txt

- test.txt파일의 내용을 화면 단위로 출력

✓ \$more -n &lt;파일명&gt;

- 예) \$more -10 test.txt

- test.txt파일의 내용을 10줄씩 출력

✓ \$more +n &lt;파일명&gt;

- 예) \$more +10 test.txt

- test.txt파일의 내용을 10번째 줄 부터 출력

## less

◆ 파일의 내용을 화면 단위로 분할 출력

◆ More명령어를 확장한 명령어

✓ 화살표 키, PageUP/Down키 등을 사용가능

✓ 종료 : 'q'

\$less &lt;파일명&gt;

예) \$less test.txt

- test.txt파일의 내용을 화면 단위로 출력

\$less +n &lt;파일명&gt;

예) \$less +10 test.txt

- test.txt파일의 내용을 10번째 줄 부터 출력



## 2. 파일 및 디렉토리 관련 명령어

### head

#### ◆ 파일의 내용 중 처음 10줄만 화면에 출력

- ✓ \$head <파일이름> // default : 10줄
- ✓ \$head -3 <파일이름> // 앞에서 3줄만 출력

```
pi@raspberrypi:~$ head /etc/dhcp/dhclient.conf
# Configuration file for /sbin/dhclient.
#
# This is a sample configuration file for dhclient. See dhclient.conf's
# man page for more information about the syntax of this file
# and a more comprehensive list of the parameters understood by
# dhclient.
#
# Normally, if the DHCP server provides reasonable information and does
# not leave anything out (like the domain name, for example), then
# few changes must be made to this file, if any.
pi@raspberrypi:~$ head -3 /etc/dhcp/dhclient.conf
# Configuration file for /sbin/dhclient.
#
# This is a sample configuration file for dhclient. See dhclient.conf's
pi@raspberrypi:~$
```

### tail

#### ◆ 파일의 내용 중, 마지막 10줄을 화면에 출력

- ✓ \$tail <파일이름> // default : 10cfn
- ✓ \$tail -5 <파일이름> // 마지막 5줄 출력

```
pi@raspberrypi:~$ tail /etc/dhcp/dhclient.conf
# medium "link0 link1";
# option host-name "andare.swiftmedia.com";
# option subnet-mask 255.255.255.0;
# option broadcast-address 192.33.137.255;
# option routers 192.33.137.250;
# option domain-name-servers 127.0.0.1;
# renew 2 2000/1/12 00:00:01;
# rebind 2 2000/1/12 00:00:01;
# expire 2 2000/1/12 00:00:01;
#}
pi@raspberrypi:~$ tail -5 /etc/dhcp/dhclient.conf
# option domain-name-servers 127.0.0.1;
# renew 2 2000/1/12 00:00:01;
# rebind 2 2000/1/12 00:00:01;
# expire 2 2000/1/12 00:00:01;
#}
pi@raspberrypi:~$
```

## 2. 파일 및 디렉토리 관련 명령어

### find

#### ◆ 파일 찾기

- ✓ \$find <경로명> <검색조건> [동작]

#### ✓ 검색조건(옵션)

- ① - name <파일이름> : 파일이름으로 검색 // \$find /home/pi -name 'test.txt'
- ② - type <파일종류> : 파일 종류로 검색 // \$sudo find / -name 'etc\*' -type d : etc로 시작하는 디렉토리 검색
- ③ - user <id> : 사용자(ID)가 소유한 파일 검색
- ④ - perm <접근권한> : 접근 권한과 일치하는 파일 검색 //

#### ✓ 동작

- ① - exec 명령 {} \; // 해당 명령 실행  
\$find /home/pi/test -user pi -exec rm {} \;  
→ find 명령으로 찾은 파일의 절대경로가 {}가 있는 위치에 삽입
- ② - ok 명령 {} \; // 해당 명령을 실행하기 전에 사용자에게 확인  
\$find /home/pi/test -user pi -ok rm {} \;
- ③ - ls // 검색결과를 ls형식으로 출력  
\$find /home/pi -name '.bash\*' -ls

## 2. 파일 및 디렉토리 관련 명령어

## find

✓ 검색조건(옵션)

① - name &lt;파일이름&gt; : 파일이름으로 검색 // \$find /home/pi -name 'test.txt'

```

pi@raspberrypi:~/test $ sudo find / -name passwd
/etc/passwd
/etc/cron.daily/passwd
/etc/pam.d/passwd
/usr/share/linian/overrides/passwd
/usr/share/doc/passwd
/usr/share/bash-completion/completions/passwd
/usr/bin/passwd
find: '/run/user/1000/gvfs': Permission denied
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ ls
abc.c  sss  temp  test.txt
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ find /home/pi -name test.txt
/home/pi/test/test.txt
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ find ~ -name test.txt
/home/pi/test/test.txt
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ find ~ -name test.txt -ls
257559      0 -rw-r--r--  1 pi      pi          0 Apr 14 20:19 /home/pi/test/test.txt
pi@raspberrypi:~/test $

```

## 2. 파일 및 디렉토리 관련 명령어

## find

✓ 검색조건(옵션)

② - type &lt;파일종류&gt; : 파일 종류로 검색 // \$sudo find / -name 'etc\*' -type d : etc로 시작하는 디렉토리 검색

```

pi@raspberrypi:~/test $ sudo find / -name 'etc*' -type d
/etc
/opt/sonic-pi/etc
/usr/local/etc
/usr/lib/node_modules/node-red/node_modules/dashdash/etc

```

③ - user &lt;id&gt; : 사용자(ID)가 소유한 파일 검색

④ - perm &lt;접근권한&gt; : 접근 권한과 일치하는 파일 검색

```

pi@raspberrypi:~/test $ ls -l
total 16
-rw-r--r-- 1 pi pi    66 Apr 14 20:20 abc.c
-rwxr-xr-x 1 pi pi 8136 Apr 14 20:21 sss
drwxr-xr-x 2 pi pi 4096 Apr 14 20:34 temp
-rw-r--r-- 1 pi pi    0 Apr 14 20:19 test.txt
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ find /home/pi/test -user pi
/home/pi/test
/home/pi/test/abc.c
/home/pi/test/sss
/home/pi/test/temp
/home/pi/test/test.txt
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ find /home/pi/test -perm 644
/home/pi/test/abc.c
/home/pi/test/test.txt
pi@raspberrypi:~/test $

```

## 2. 파일 및 디렉토리 관련 명령어

## find

## ✓ 동작

① -exec 명령 {} \; // 해당 명령 실행

\$find /home/pi/test -user pi -exec rm {} \;

→ find 명령으로 찾은 파일의 절대경로가 {}가 있는 위치에 삽입

```
pi@raspberrypi:~/test $ ls -l /home/pi/test
total 16
-rw-r--r-- 1 pi pi 66 Apr 14 20:20 abc.c
-rwxr-xr-x 1 pi pi 8136 Apr 14 20:21 sss
drwxr-xr-x 2 pi pi 4096 Apr 14 20:34 temp
-rw-r--r-- 1 pi pi 0 Apr 14 20:19 test.txt
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ find /home/pi/test -user pi -exec rm {} \;
rm: cannot remove '/home/pi/test': Is a directory
rm: cannot remove '/home/pi/test/temp': Is a directory
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ ls
temp
pi@raspberrypi:~/test $ find /home/pi/test -user pi -exec rm -rf {} \;
find: '/home/pi/test': No such file or directory
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ ls
pi@raspberrypi:~/test $
```

## 2. 파일 및 디렉토리 관련 명령어

## find

## ◆ 파일 찾기

✓ \$find <경로명> <검색조건> [동작]

옵션	설명
-name <filename>	파일이름이 filename인 파일을 찾는다
-amin n	n분 전에 마지막으로 access한 파일 찾기
-atime n	n일 전에 마지막으로 access한 파일 찾기
-ctime n	n분/일 전에 마지막으로 변경된 파일 찾기
-newer	지정한 파일보다 최신의 파일 찾기

```
pi@raspberrypi:~/test $ find /home/pi/test -amin 10
pi@raspberrypi:~/test $ find /home/pi/test -amin -10
/home/pi/test/abc.c
/home/pi/test/sss
/home/pi/test/test.txt
pi@raspberrypi:~/test $ find /home/pi/test -cmin -10
/home/pi/test
/home/pi/test/abc.c
/home/pi/test/sss
/home/pi/test/test.txt
pi@raspberrypi:~/test $ find /home/pi/test -empty
/home/pi/test/test.txt
pi@raspberrypi:~/test $ find /home/pi/test -newer test.txt
/home/pi/test
/home/pi/test/abc.c
/home/pi/test/sss
pi@raspberrypi:~/test $
```

## 2. 파일 및 디렉토리 관련 명령어

## grep

◆ 파일 내용 중 원하는 문자열 검색

\$grep [옵션] "문자열" &lt;파일명&gt;

```
pi@raspberrypi:~/test $ ls
test.txt
pi@raspberrypi:~/test $ more test.txt
Hi! Nice to meet you^^
My name is PYO WOL SEONG
```

E-mail : wspyo74@naver.com

```
pi@raspberrypi:~/test $ grep -n "PYO" ./test.txt
```

2:My name is PYO WOL SEONG

pi@raspberrypi:~/test \$

```
pi@raspberrypi:~/test $ grep -c "PYO" ./test.txt
```

1

```
pi@raspberrypi:~/test $ grep -c "google" ./test.txt
```

0

pi@raspberrypi:~/test \$

```
pi@raspberrypi:~/test $ grep -v "PYO" ./test.txt
```

Hi! Nice to meet you^^

E-mail : wspyo74@naver.com

옵션	설명
-v	찾는 문자열이 없는 라인을 표시
-c	찾는 문자열이 있는 행의 개수를 표시
-i	찾는 문자열의 대소문자를 구별하지 않고 찾기
-n	찾는 문자열이 있는 행의 번호를 함께 표시

```
pi@raspberrypi:~/test $ grep -i pyo ./test.txt
```

My name is PYO WOL SEONG

E-mail : wspyo74@naver.com

pi@raspberrypi:~/test \$

```
pi@raspberrypi:~/test $ grep -i PYO ./test.txt
```

My name is PYO WOL SEONG

E-mail : wspyo74@naver.com

pi@raspberrypi:~/test \$

## 2. 파일 및 디렉토리 관련 명령어

## grep

◆ 명령어 결과 중 원하는 문자열 검색

```
pi@raspberrypi:~/test $ grep -n "pi" /etc/passwd
24:pi:x:1000:1000:,,,:/home/pi:/bin/bash
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ cat /etc/passwd | grep -n "pi"
24:pi:x:1000:1000:,,,:/home/pi:/bin/bash
pi@raspberrypi:~/test $ cat /etc/passwd | grep "pi"
pi:x:1000:1000:,,,:/home/pi:/bin/bash
pi@raspberrypi:~/test $
```

/etc/service 파일에서 'NETBIOS'를 포함한 행을 행번호와 함께 출력하라?

## clear

◆ 화면의 내용을 지워 줌

\$clear

## 2. 파일 및 디렉토리 관련 명령어

### file

#### ◆ 파일의 종류를 출력

- windows는 확장자를 보고 파일의 종류를 확인
- 리눅스는 확장자가 없는 경우도 많음
- 예) \$file /bin/gzip

```
pi@raspberrypi:~/test $ ls
pi@raspberrypi:~/test $ touch test.txt
pi@raspberrypi:~/test $ file test.txt
test.txt: empty
pi@raspberrypi:~/test $ nano abc.c
pi@raspberrypi:~/test $ gcc -o sss abc.c
pi@raspberrypi:~/test $ ls
abc.c  sss  test.txt
pi@raspberrypi:~/test $ file abc.c
abc.c: C source, ASCII text
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ file sss
sss: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-armhf.so.3, for GNU/Linux 3.2.0, BuildID[sha1]=9af9771941dee10124131c43f4b0f2324231134b, not stripped
pi@raspberrypi:~/test $
```

## 2. 파일 및 디렉토리 관련 명령어

### 파일 링크

#### ◆ 링크

- 기존 파일에 새로운 파일명을 붙이는 것
- 경로가 복잡하거나, 파일명이 복잡한 경우, 짧게 줄여서 사용

#### ◆ 링크의 목적

- 긴 파일명이나 디렉토리명을 짧게 줄여 사용하기 위해 사용
- 파일의 위치가 비 실행 경로에 있을 경우, 해당 파일을 실행하기 위해 사용
- 파일의 경로의 길이가 긴 경우, 간단하게 이동하기 위해 사용

#### ◆ 링크의 종류

- 하드 링크 (Hard Link)
- 심볼릭 링크 (Symbolic Link)

## 2. 파일 및 디렉토리 관련 명령어

## 하드 링크

## ◆ 하드 링크 파일

- 기존 파일에 새로운 파일명을 붙이는 것
- 원본 파일의 또 다른 이름

## ◆ 하드링크의 특징

- 링크 파일을 수정하면, 원본파일도 수정 됨
- 링크 시, 원본파일, 링크파일 모두 Link수 1 증가
- 원본 삭제 시, 링크파일은 정상
- 각각 같은 inode 를 갖는다. (\$ls -li)

## ◆ 사용법

\$ln <원본파일> <링크파일>

예) \$ln srcfile hlink

→ srcfile파일을 hlink파일로 하드링크

srcfile	inode 100
	inode 101
	inode 102
	inode 103

hlink

링크 수 :  
1 증가 (1 → 2)

```
pi@raspberrypi:~/test $ ls -l
total 4
-rw-r--r-- 1 pi pi 50 Apr 15 00:25 srcfile.txt
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ more ./srcfile.txt
Hi! Nice to meet you^^
E-mail : wspyo74@naver.com
pi@raspberrypi:~/test $ ln srcfile.txt hardlink.txt
pi@raspberrypi:~/test $ ls -li
total 8
257589 -rw-r--r-- 2 pi pi 50 Apr 15 00:25 hardlink.txt
257589 -rw-r--r-- 2 pi pi 50 Apr 15 00:25 srcfile.txt
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ vi hardlink.txt
pi@raspberrypi:~/test $ more hardlink.txt
Hi! Nice to meet you^^
E-mail : wspyo74@naver.com
TEL : 010-1234-5678
pi@raspberrypi:~/test $ more srcfile.txt
Hi! Nice to meet you^^
E-mail : wspyo74@naver.com
TEL : 010-1234-5678
pi@raspberrypi:~/test $ rm -rf ./srcfile.txt
pi@raspberrypi:~/test $ ls -li
total 4
-rw-r--r-- 1 pi pi 71 Apr 15 00:28 hardlink.txt
pi@raspberrypi:~/test $
```

링크파일 수정 시, 원본파일도 수정

원본파일 삭제 시, 링크파일 문제 없음

## 2. 파일 및 디렉토리 관련 명령어

## 심볼릭 링크

## ◆ 심볼릭 링크 파일

- 기존 파일을 가리키는 새로운 파일 생성하는 것
- 원본 파일을 가리키는 포인터의 개념(원도우 바로가기)

## ◆ 하드링크의 특징

- 링크 파일을 수정하면, 원본파일도 수정 됨
- 링크 시, 원본파일, 링크파일 모두 Link수 증가하지 않음
- 원본 삭제 시, 링크 파일은 문제발생(링크 깨짐)
- 각각 다른 inode 를 갖는다. (\$ls -li)

## ◆ 사용법

\$ln -s <원본파일> <링크파일>

예) \$ln -s srcfile hlink

→ srcfile파일을 hlink파일로 심볼릭 링크

srcfile	inode 100
	inode 101
	inode 102
	inode 103

symlink

링크 수 :  
1로 변화 없음

```
pi@raspberrypi:~/test $ ls -l
total 4
-rw-r--r-- 1 pi pi 71 Apr 15 00:28 srcfile.txt
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ ln -s srcfile.txt ./symlink.txt
pi@raspberrypi:~/test $ ls -li
total 4
257589 -rw-r--r-- 1 pi pi 71 Apr 15 00:28 srcfile.txt
257574 lrwxrwxrwx 1 pi pi 11 Apr 15 00:46 symlink.txt -> srcfile.txt
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ vi symlink.txt
pi@raspberrypi:~/test $ more symlink.txt
Hi! Nice to meet you^^
E-mail : wspyo74@naver.com
www.sungkyul.ac.kr
pi@raspberrypi:~/test $ more srcfile.txt
Hi! Nice to meet you^^
E-mail : wspyo74@naver.com
www.sungkyul.ac.kr
pi@raspberrypi:~/test $ rm -rf ./srcfile.txt
pi@raspberrypi:~/test $ ls -li
total 0
257574 lrwxrwxrwx 1 pi pi 11 Apr 15 00:46 symlink.txt -> srcfile.txt
pi@raspberrypi:~/test $
pi@raspberrypi:~/test $ more symlink.txt
more: stat of symlink.txt failed: No such file or directory
pi@raspberrypi:~/test $
```

링크파일 수정 시, 원본파일도 수정

원본파일 삭제 시, 링크파일 문제 발생

