

---

# 리눅스 환경

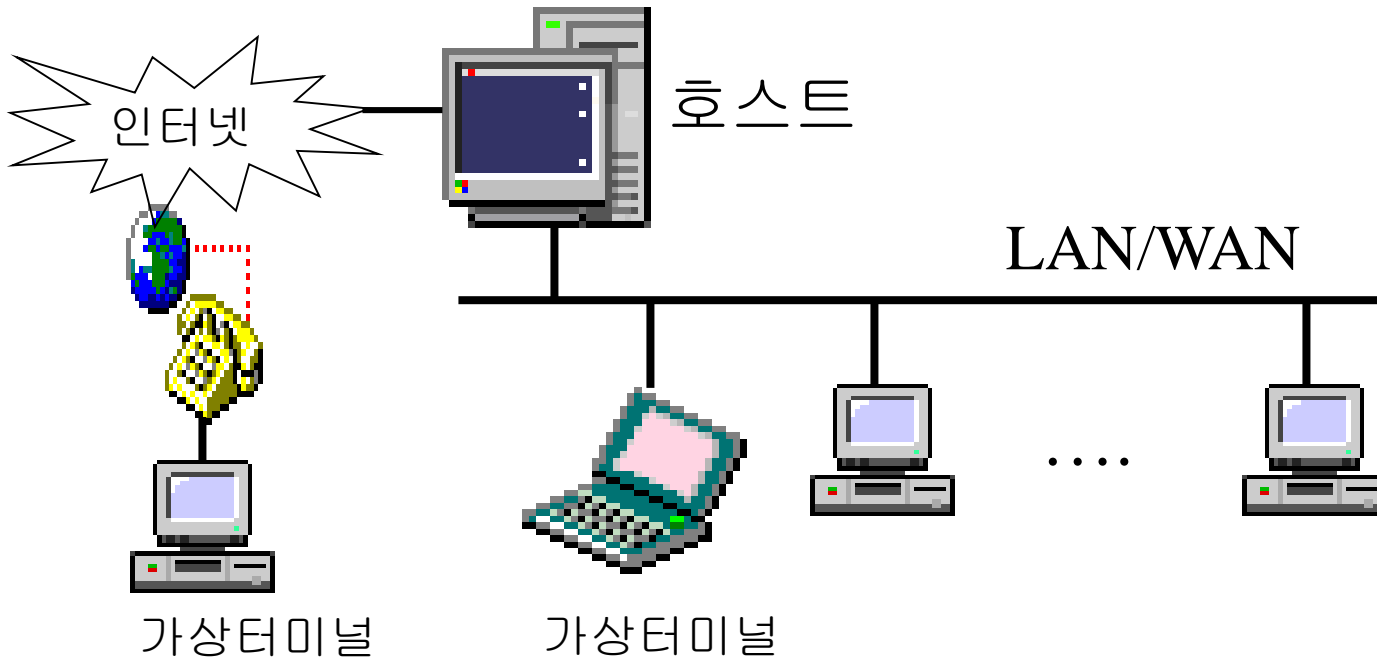
# 목차

---

- 리눅스 접속
- 리눅스 파일 시스템
- 기본 명령어
- 일반 명령어
- 시스템 관리 명령
- 파일 편집

# 리눅스 접속

- 리눅스 부팅 후 로그인
- 가상 터미널(Virtual terminal) 사용
  - ◆ telnet, SSH, zterm, **puTTY**
- 윈도우용 접속 프로그램
  - ◆ **VNC**, NX, RDP, X Manager, X Ming, Exceed



# 텍스트 접속

## ■ 라즈베리파이 설정

◆ **raspi-config** 프로그램을 실행하고 SSH 항목 활성화

## ■ MS 윈도우 환경에서 puTTY 설치

◆ puTTY 홈페이지(<http://www.chiark.greenend.org.uk/~sgtatham/putty/>)에서 puTty 다운로드 및 설치

◆ 실행한 후에 라즈베리파이 IP 입력하고 접속 시도

## ■ 정상 접속 시

◆ 최초 접속 시 공개 키 설정

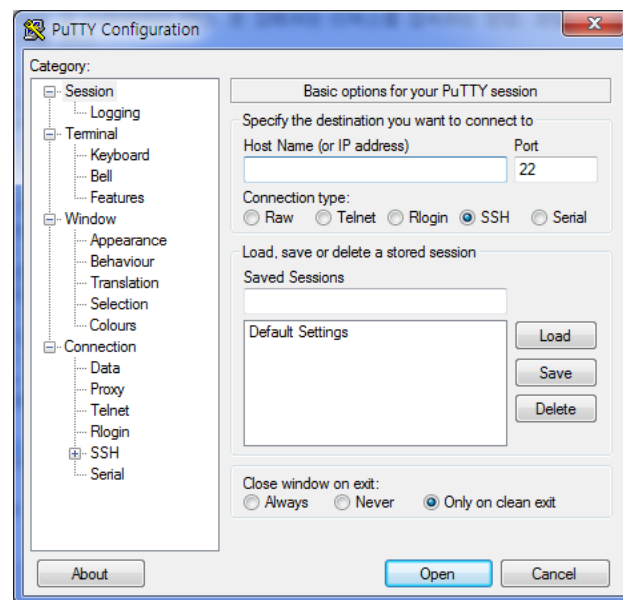
➤ 'yes' 입력

◆ putty 터미널 창 생성 및 로그인

◆ 로그인 아이디 및 패스워드 입력

➤ Login: pi

➤ Password: raspberry



# 그래픽 접속

## ■ VNC 서버 설치

- ◆ VNC (Virtual Network Computing) 는 그래픽 데스크톱 환경으로 원격 컴퓨터를 제어하는 소프트웨어의 일종
- ◆ 라즈베리파이에 VNC 서버 설치

`$ sudo apt-get install tigntvncserver`

### ◆ 서버 실행

- 인자 :1 은 VNC 디스플레이 번호를 1로 설정
- 처음 실행하면 패스워드를 요청하므로 입력

`$ vncserver :1`

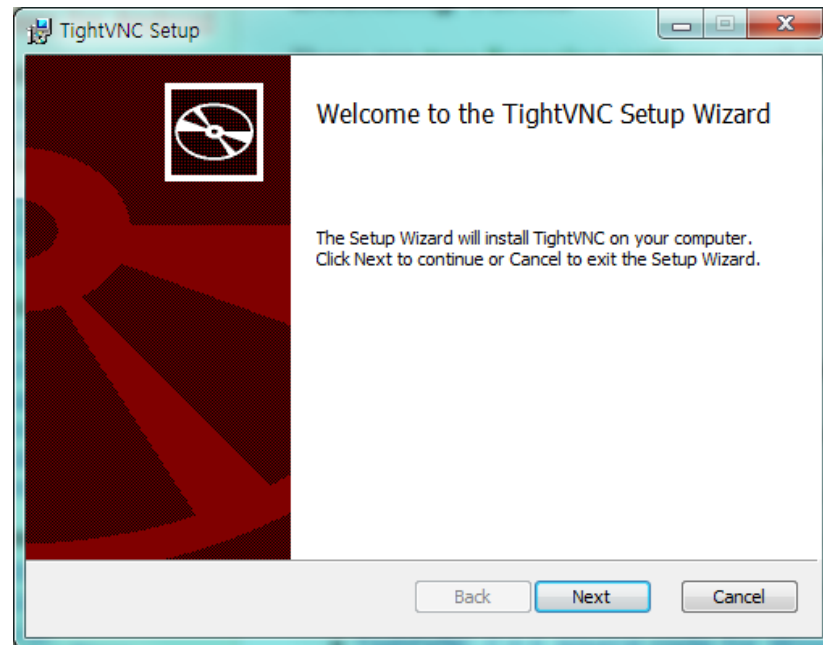
# 그래픽 접속

## ■ VNC 클라이언트 설치

### ◆ 윈도우 등의 다른 컴퓨터에 VNC 뷰어 프로그램을 설치

➤ **RealVNC** 또는 **TightVNC**(<http://www.tightvnc.com>)

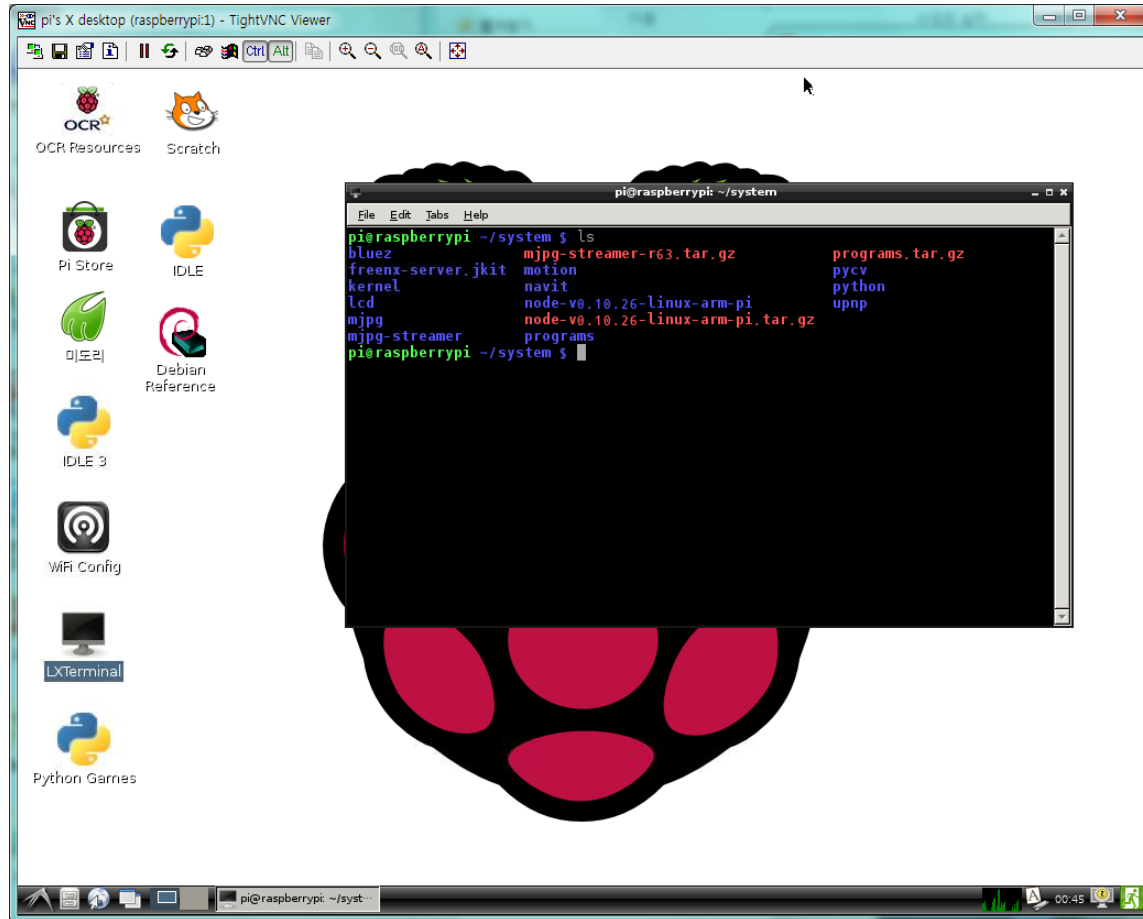
– 다운로드 및 설치



- **TightVNC Viewer** 프로그램 실행
- “Remote Host” 입력 창에 <라즈베리파이 IP>:1 을 입력
- **Connect** 버튼을 클릭하고 패스워드 입력

# 리눅스 접속

## ■ VNC 실행화면



# Login과 Password

## ■ 리눅스 사용자 접속 과정

- ◆ 시스템 관리자가 제공하는 사용자 등록명 (login ID)
- ◆ 일종의 비밀번호인 패스워드(password)
  - 다른 사용자로부터 자신의 정보들을 보호
- ◆ 접속 후 자신의 홈(home) 디렉터리에서 파일 생성/편집/저장 가능

## ■ Login ID 규칙

- ◆ 영문 소문자와 숫자의 조합으로 8자 이내의 단어

## ■ Password 규칙

- ◆ 8자 내외의 문자, 숫자, 특수문자로 구성
- ◆ 2자 문자와 1자 숫자 또는 특수문자가 포함
- ◆ 적어도 3자의 문자는 상이해야 함
- ◆ password 는 Login ID 나 User name 과 상이해야 함



# 프롬프트

## ■ 셸(Shell)

- ◆ 명령어 해석 프로그램
- ◆ 다음 명령을 입력할 준비가 될 때, **프롬프트(prompt)** 출력

## ■ 셸 프롬프트

- ◆ “%”, “\$” 등의 기호 사용
- ◆ Bourne Shell 이나 Korn Shell 은 “%”, C Shell 은 “\$” 사용
- ◆ 사용자의 목적에 따라 프롬프트 변경 가능

➤ **pi@raspberrypi ~ \$**

- **pi** : 로그인한 사용자 계정
- **raspberrypi** : 현재 접속하고 있는 **호스트 이름**
- **~** : 작업하고 있는 현재 디렉터리의 위치 (~ 는 사용자 홈디렉터리)

# Unix/Linux 파일

## ■ 파일의 개념

- ◆ 파일은 사용자가 이용할 수 있는 데이터의 실체를 의미. 여기에는 시스템 그 자체도 하나의 파일로 볼 수 있고, 응용 프로그램도 파일로 볼 수 있음.

## ■ 파일의 종류

### ◆ 디렉터리

- 다른 파일을 조직하고 액세스하는데 필요한 정보를 가지고 있음

### ◆ 일반화일

- 보통 우리가 평상시 쓰는 파일
- 문서(text) 또는 아스키(ASCII) 파일
  - 키보드로 입력 가능한 ASCII 문자로 구성
  - ASCII 문자 : 대·소문자, 숫자, 공백, 탭, 문장부호, 제어문자 포함 128개 코드
- 2진 파일 - 문자가 아닌 데이터 포함. 예) 실행파일

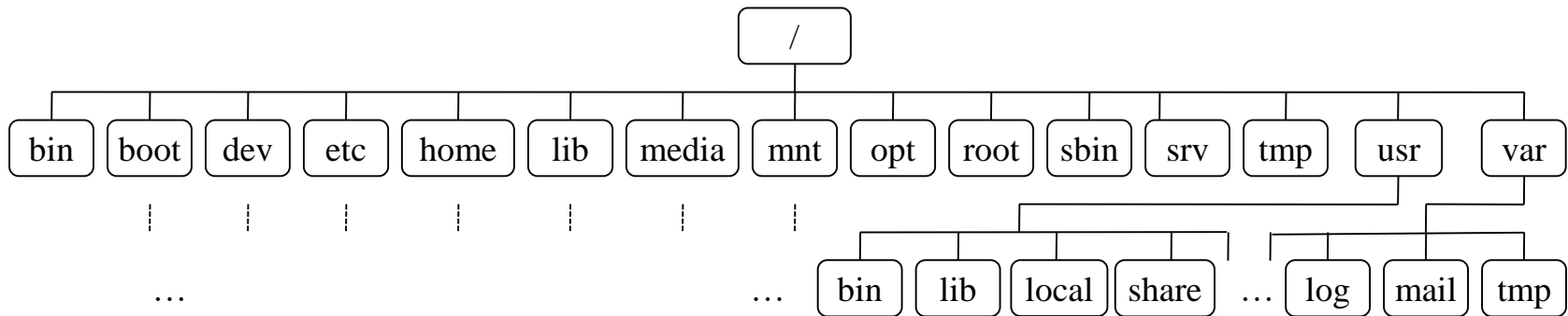
### ◆ 특수화일

- Unix/Linux가 자원을 관리하는 디바이스(Device) 파일 등

# Unix/Linux 파일시스템

## ■ 트리(Tree)구조 파일 시스템

- ◆ 파일과 디렉터리의 계층적 구조
- ◆ 최상위는 항상 루트(root) 디렉터리에 기반



# Linux 파일시스템

## ■ 디렉터리의 종류

디렉터리	의미
/bin	보통 ls, cp 와 같은 기본적인 명령어 실행 파일이 저장
/dev	tty, ram 과 같은 하드웨어 장치 파일을 포함
/etc	시스템 관리자를 위해 필요한 <b>환경 설정 파일</b> 등을 포함
/home	여러 사용자들의 <b>홈 디렉터리들이 위치</b>
/lib	시스템 프로그램 실행을 위한 각종 라이브러리를 포함
/mnt	외부 장치인 플로피 디스크, 시디롬, 삼바등을 마운트하기 위해서 제공되는 디렉터리
/proc	시스템 실행 정보를 제공하는 proc 파일 시스템 디렉터리임
/root	시스템 관리자인 슈퍼유저의 <b>홈 디렉터리</b>
/sbin	슈퍼유저가 사용할 수 있는 중요한 관리 명령어들이 위치
/tmp	시스템이 사용하는 임시파일을 저장
/usr	시스템에 사용되는 각종 프로그램들이 설치되는 디렉터리
/var	빠르게 갱신되거나 변화하는 <b>가변 데이터들이</b> 위치(로그, 메일 등)

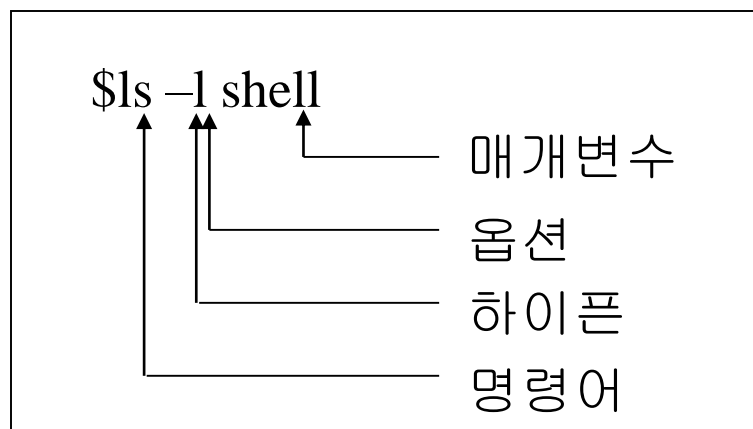
# Unix/Linux 파일시스템

## ■ 홈 디렉터리

- ◆ 시스템관리자가 사용자를 등록할 때 기본적으로 주어지는 전용 공간
- ◆ 기본적으로 `"/home/userID"` 형식으로 생김
- ◆ 사용자의 홈 디렉터리 식별하는 `$HOME`이라는 변수를 지정
- ◆ 홈 디렉터리 환경설정 파일
  - `".profile"` , `".login"` , `".cshrc"`

# 명령어

## ■ 명령어의 구조



## ■ 명령어 종류

### ◆ 비 대화식(NonInteractive) 명령어

- 해당 명령어만 입력하여 사용자가 원하는 결과를 표시
- 매우 간단하며 사용도 편리

### ◆ 대화식(Interactive) 명령어

- 명령어들이 실행하는 동안에 어떤 것을 질문하고 사용자가 대답하고 하는 일련의 과정이 끝나면 결과를 보여줌.

# 셸에서 사용하는 특수문자들

코드	키	의미
init	^C	실행중인 프로그램의 중단
erase	^H	마지막 문자의 삭제
werase	^W	마지막 한 문자의 삭제
kill	^U	한 줄 전체의 삭제
quit	^W	프로그램을 중단시키고 코어(core)파일에 저장
stop	^S	화면표시의 정지
start	^Q	화면표시의 재개
eof	^D	더 이상 자료 없음을 표시

# 작업종료

---

- 사용자가 작업을 마치기 위해서는 반드시 로그아웃 (Logout)을 해야 함.
- 로그아웃 방법
  - ◆ “logout” 명령
  - ◆ "Ctrl+D" 키
  - ◆ "exit" 명령어



# 디렉터리 파일 목록 보기

## ■ 디렉터리 파일 보기 (ls(List))

ls	
일반형식	ls [ 옵션 ]
주요옵션	<ul style="list-style-type: none"><li>-a, --all : 디렉터리 내의 모든 파일을 출력</li><li>-l, --format=long : 파일 종류, 사용권한, 크기 등 출력</li><li>-s, --size : 1K 단위로 파일 크기를 표시</li><li>-t, --sort=time : 최근에 만들어진 파일부터 출력</li><li>-c, --time : 최근에 변경한 파일부터 출력</li><li>-R : 하위 디렉터리까지 출력</li><li>-i 파일의 inode 번호를 보여 준다.</li><li>-u 액세스(access)한 날짜 순서대로 정렬한다.</li><li>-r 정렬된 순서의 역으로 출력한다.</li></ul>

# 디렉터리 파일 목록 보기

## ■ 사용 예

```
pi@raspberrypi ~ $ ls -l
total 12
drwxr-xr-x 2 pi pi 4096 7월 18 16:01 Desktop
drwxr-xr-x 2 pi pi 4096 7월 20 13:47 Downloads
drwxr-xr-x 2 pi pi 4096 1월 27 20:15 python_games
pi@raspberrypi ~ $ ls -a
. .bashrc Downloads .local .thumbnails
.. .cache .gconf .pki .Xauthority
.asoundrc .config .gststreamer-0.10 .profile .xsession-errors
.bash_history .dbus .gvfs python_games
.bash_logout Desktop .lessht .themes
```

# 디렉터리 경로

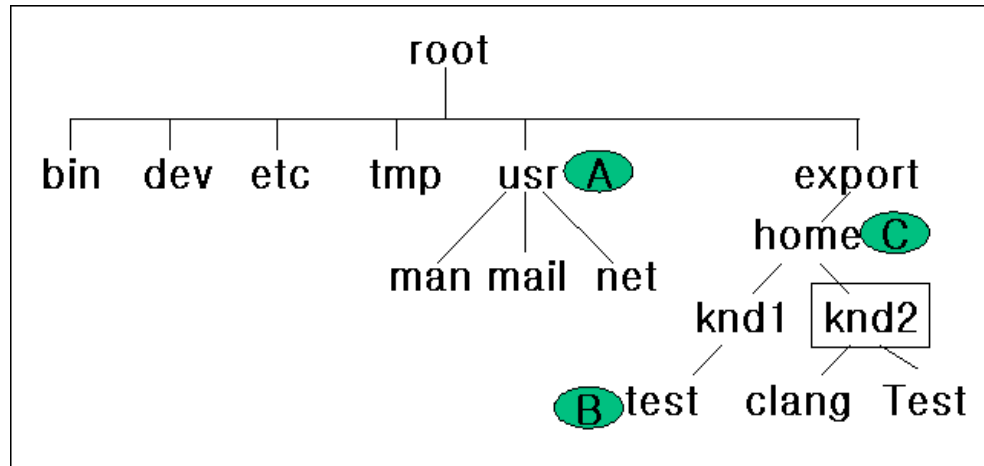
## ■ 절대경로와 상대경로

◆ 절대경로 - 모든 경로 명을 반드시 루트 디렉터리인 "/"로 시작되어 특정한 파일이나 디렉터를 표시

- A - /usr
- B - /export/home/knd1/test
- C - /export/home/home

◆ 상대경로 - 현재 작업 디렉터리에 대해 상대적인 경로이름 이용

- A - ../../../../usr
- B - ../knd1/test
- C - ../



# 작업 디렉터리 확인

## ■ 현재 작업중인 디렉터리 확인(pwd(Print Working Directory))

pwd	
일반형식	pwd
주요옵션	

## ■ 사용 예

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

# 작업 디렉터리 변경

## ■ 디렉터리 변경(cd (Change Directory))

cd	
일반형식	cd [ directory ]
주요옵션	

## ■ 사용 예

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

# 파일 만들기

## ■ 파일 만들기(touch,cat,vi)

touch	
일반형식	touch [ -acm ] [-r ref_file   -t time ] file ... touch [ -acm ] [ date_time ] file ...
주요옵션	-a : 접근시간만 갱신 -c : 지정된 파일이 없는 경우 새로 작성하지 않음 -m : 수정시간만 갱신 -r 또는 -t : 현재 시간 대신 지정 시간(ref_file 또는 time) 사용

## ■ 사용 예

```
pi@raspberrypi ~ $ mkdir Test
pi@raspberrypi ~ $ cd Test
pi@raspberrypi ~/Test $ touch catTest.txt
pi@raspberrypi ~/Test $ ls -la catTest.txt
-rw-r--r-- 1 pi pi 0 2월 11일 16:13 catTest.txt
```

# 파일 만들기

## ■ 파일 만들기(cat (Concatenate, 사슬처럼 잇다))

cat	
일반형식	cat > filename
기능	화면으로부터 입력받아 파일 생성 종료 시 ctrl-D 입력하면 쉘로 빠져나옴

## ■ 사용 예

```
pi@raspberrypi ~ $ cat > catTest.txt
```

```
Hello!
```

```
Nice to meet you. ← Ctrl-D 입력
```

```
pi@raspberrypi ~ $ cat catTest.txt
```

```
Hello!
```

```
Nice to meet you.
```

```
pi@raspberrypi ~ $
```

# 파일 복사

## ■ 파일의 복사(cp(copy))

cp	
일반형식	cp [ -fip ] source dest cp [ -fipr ] source.. dest_dir
주요옵션	-f: 복사할 파일이 있을 경우 삭제하고 복사 -i: 복사할 파일이 있을 경우 복사할 것인지 물어봄 -p: 원본 파일의 모든 정보를 보존한 채 복사 -r: 하위 디렉터리에 있는 모든 파일을 복사

## ■ 사용 예 - /bin/date 를 Test 디렉터리 안으로 복사

```
pi@raspberrypi ~ $ ls
Desktop Downloads python_games Test
pi@raspberrypi ~ $ cp /bin/date Test
pi@raspberrypi ~ $ cd Test
pi@raspberrypi ~/Test $ ls
catTest.txt date
```



# 파일 삭제

## ■ 파일의 삭제(**rm**(**R**emove))

rm	
일반형식	rm [ -firv ] source dest
주요옵션	-f : 지울 파일이 있을 경우 강제로 삭제 -i : 지울 파일이 있을 경우 지울 것인지 물어봄 -r : 하위 디렉터리에 있는 모든 파일을 삭제 -v : 지우는 파일 정보를 출력

## ■ 사용 예

```
pi@raspberrypi ~/Test $ ls
catTest.txt date
pi@raspberrypi ~/Test $ rm date
pi@raspberrypi ~/Test $ ls
catTest.txt
```

# 파일 이름 변경과 옮기기

## ■ 파일 이름 변경과 옮기기(mv(Move))

mv	
일반형식	mv [ -fi ] source dest mv [ -fi] source ... dest_dir
주요옵션	-b : 대상 파일이 지워지기 전에 백업 파일을 만듦 -f : 대상 파일의 접근허가와 관계없이 무조건 파일을 이동 -i : 대상 파일이 기존 파일이면, 덮어쓸 것인지 물어봄 -u : 대상 파일보다 원본 파일이 최근의 것일 때 업그레이드 -v : 파일 옮기는 과정을 자세하게 보여준다

# 파일 이름 변경과 옮기기

## ■ 사용 예

```
pi@raspberrypi ~/Test $ cd
pi@raspberrypi ~ $ touch mvTest.c
pi@raspberrypi ~ $ ls
Desktop Downloads mvTest.c python_games Test
pi@raspberrypi ~ $ mv mvTest.c Test
pi@raspberrypi ~ $ ls
Desktop Downloads python_games Test
pi@raspberrypi ~ $ cd Test
pi@raspberrypi ~/Test $ ls
catTest.txt mvTest.c
pi@raspberrypi ~/Test $ cp mvTest.c ../; cd ..
pi@raspberrypi ~ $ mv -i mvTest.c Test
mv: overwrite `Test/mvTest.c'? y
pi@raspberrypi ~ $ ls
Desktop Downloads python_games Test
```

# 디렉터리 생성

## ■ 디렉터리 생성 (mkdir(**M**ake **D**irectory))

mkdir	
일반형식	mkdir [ -m mode ] [ -p ] dir ...
주요옵션	-m : 새로운 디렉터리의 허가모드를 지정한 모드로 설정 -p : 하위 디렉터리가 존재하지 않는 경우 함께 생성

## ■ 사용 예

```
pi@raspberrypi ~ $ ls
Desktop Downloads python_games Test
pi@raspberrypi ~ $ mkdir temp
pi@raspberrypi ~ $ ls
Desktop Downloads python_games temp Test
pi@raspberrypi ~ $ mkdir -p Test/Test2
pi@raspberrypi ~ $ mkdir -p Test3/Test5
pi@raspberrypi ~ $ ls
Desktop Downloads python_games temp Test Test3
pi@raspberrypi ~ $ ls Test3
Test5
```

# 디렉터리 삭제

## ■ 디렉터리 삭제(rmdir)

rmdir	
일반형식	rmdir [ -p ] dir ...
주요옵션	-p : 지정한 하위 디렉터리까지 삭제

## ■ 사용 예

```
pi@raspberrypi ~ $ ls Test3
```

```
Test5
```

```
pi@raspberrypi ~ $ rmdir -p Test3/Test5
```

```
pi@raspberrypi ~ $ ls Test3
```

```
ls: cannot access Test3: No such file or directory
```

```
pi@raspberrypi ~ $
```

# 파일 내용 보기

## ■ 파일의 내용보기(cat, more, less, page, head, tail)

cat	
일반형식	cat [ -bnsuvet ] [ file ... ]
주요옵션	-b, -n : 모든 라인에 행 번호를 붙임 (-b 는 nonblank 라인에) -u : 출력이 버퍼에 기록되지 않게 함 -v : 출력할 수 없는 문자를 출력. -e : -vE 옵션과 같음. 라인의 끝에 "\$"를 표시함 -t : -vT 옵션과 같음. \로 TAB 문자를 표시함

## ■ 사용 예

```
pi@raspberrypi ~ $ cat catTest.txt
```

```
Hello!
```

```
Nice to meet you.
```

```
pi@raspberrypi ~ $ cat -b catTest.txt
```

```
1 Hello!
```

```
2 Nice to meet you.
```

# 파일 내용 보기

## ■ 파일의 내용보기(more, page)

more, page	
일반형식	<code>more [ -cdfllrsuw ] [ -lines ] [ +linenumber ] [ +/pattern ] [ file... ]</code> <code>page [ -cdfllrsuw ] [ -lines ] [ +linenumber ] [ +/pattern ] [ file... ]</code>
주요옵션	<ul style="list-style-type: none"><li>-c : 화면 스크롤 대신에 화면의 상단에서부터 나타내게 함</li><li>-d : 계속하기 위한 프롬프트 출력</li><li>+linenumber : 지정된 linenumber부터 출력함</li><li>+pattern : 처음 패턴이 발견된 라인의 앞 두 라인부터 출력</li></ul>

# 소유권과 보호

## ■ 파일 접근을 위해 각 파일에 대해 사용권한 설정

◆ 파일/디렉터리에 대한 사용자 범위 제한 및 발생 가능한 사용자 실수로부터의 보호

## ■ 사용권한

모드	종류	의미
4	Read	그 파일에 대해 읽기 가능
2	Write	그 파일에 대해 쓰기 가능
1	Execute	그 파일에 대해 실행 가능(명령어로 사용 가능)
0		그 파일에 대해 어떤 사용권한도 없음

## ■ 소유권한

◆ Unix에서 제공되는 모든 파일들은 3가지 권한의 조합으로 표시

종류	의미
소유자(Owner)	파일이나 디렉터리를 처음 만든 사람
그룹(Group)	사용자는 어느 특정한 그룹에 속함
다른사람들(Public)	현재 사용자 계정을 가진 모든 사람



# 소유권과 보호

## ■ 사용권한 조합

사용권한			값
읽기	쓰기	실행	
-	-	-	0
-	-	Y	1
-	Y	-	2
-	Y	Y	3
Y	-	-	4
Y	-	Y	5
Y	Y	-	6
Y	Y	Y	7

# 파일 사용권한 변경

## ■ 파일 사용권한 변경(chmod)

chmod	
일반형식	chmod [ -fR ] <absolute-mode> file ... chmod [ -fR ] <symbolic-mode-list> file ...
주요옵션	-f : chmod가 에러가 발생하여도 강제로 모드를 변경 -R: 모든 하위 디렉터리를 포함하여 모드를 변경

### ◆ 기호 모드

기호	의미
+	사용권한 부여
-	사용권한 제거
=	사용권한 유지
s	소유자와 그룹만 허용
u	소유자 권한
g	그룹 권한
o	다른 사용자 권한
a	소유자, 그룹, 다른 사용자 모두 사용권한 허가

# 파일 사용권한 변경

## ■ 절대 모드

사용자 범위	사용권한	기호	절대모드
소유자	읽기	r	400
	쓰기	w	200
	실행	x	100
그룹	읽기	r	40
	쓰기	w	20
	실행	x	10
다른 사용자	읽기	r	4
	쓰기	w	2
	실행	x	1

# 파일 사용권한 변경

## ■ 사용 예

```
pi@raspberrypi ~/Test $ ls -l
```

합계 8

```
-rw-r--r-- 1 pi pi 25 8월 27 14:38 catTest.txt
-rw-r--r-- 1 pi pi 0 8월 27 14:03 mvTest.c
drwxr-xr-x 2 pi pi 4096 8월 27 14:24 Test2
-rwxrwxr-x 1 pi pi 0 8월 27 14:23 Test.c
```



```
pi@raspberrypi ~/Test $ chmod 725 Test.c
```

```
pi@raspberrypi ~/Test $ chmod g+w Test2
```

```
pi@raspberrypi ~/Test $ ls -l
```

합계 8

```
-rw-r--r-- 1 pi pi 25 8월 27 14:38 catTest.txt
-rw-r--r-- 1 pi pi 0 8월 27 14:03 mvTest.c
drwxrwxr-x 2 pi pi 4096 8월 27 14:24 Test2
-rwx-w-r-x 1 pi pi 0 8월 27 14:23 Test.c
```

# 파일 소유자 및 그룹 변경

## ■ 파일 소유자 변경(chown)

chown	
일반형식	chown [ -fR ] owner [ : group ] file ...
주요옵션	-f : chown이 에러가 발생하여도 강제로 모드를 변경 -R: 모든 하위 디렉터리를 포함하여 모드를 변경

## ■ 파일 그룹명 변경(chgrp)

chgrp	
일반형식	chgrp [ -fR ] group file ...
주요옵션	-f : chgrp가 에러가 발생하여도 강제로 모드를 변경 -R: 모든 하위 디렉터리를 포함하여 모드를 변경