

---

# 2장. 리눅스 명령어

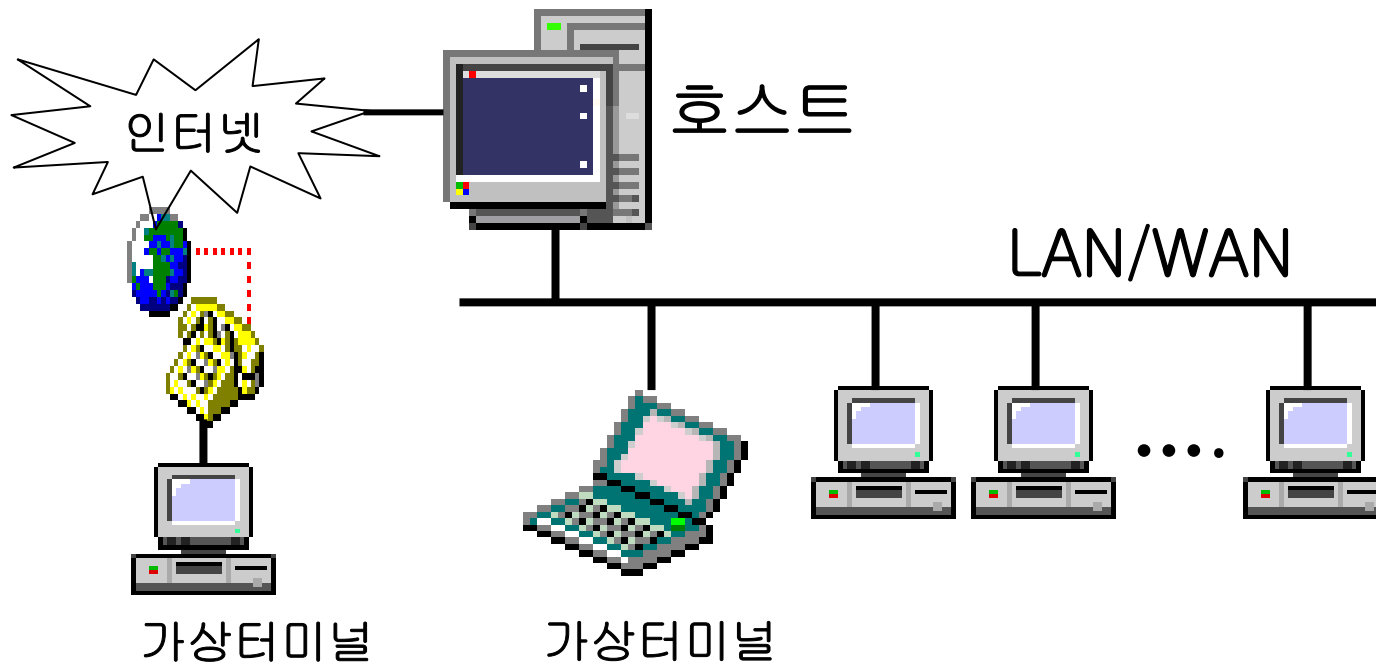
# 리눅스 접속

## ■ 가상 터미널(Virtual terminal) 사용

◆ 넷텀, 세롬 데이터맨 프로, **telnet**

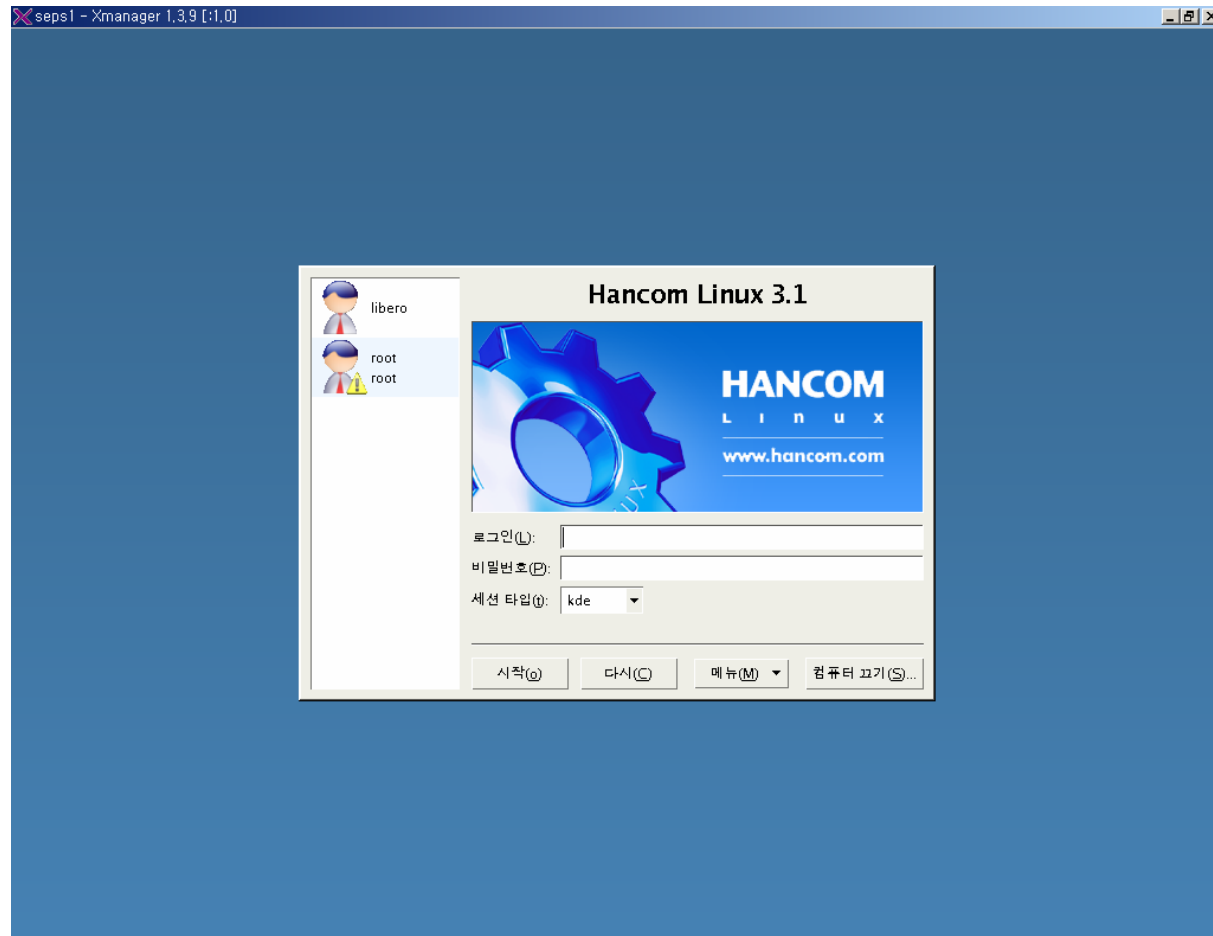
## ■ 윈도우용 접속 프로그램

◆ **X Manager, Exceed**



# 리눅스 접속

## ■ Xmanager (Xbrowser) 접속 화면



# 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) 디렉토리에서 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

➤ **[cprog2@seps1 home]\$**

- cprog2 : 로그인한 사용자 계정
- seps1 : 현재 접속하고 있는 호스트 이름
- home : 작업하고 있는 현재 디렉토리의 위치

# 명령어 입력

---

## ■ 명령어 라인 (Command Line)

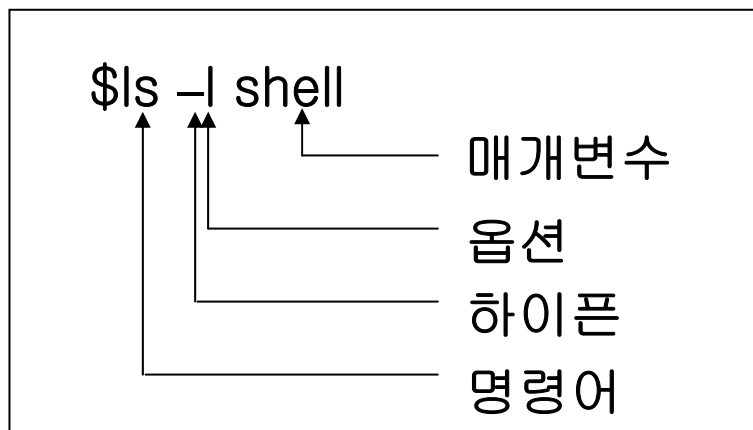
- ◆ 명령어를 입력한 전체 줄
- ◆ 반드시 **Enter** 키를 눌러야 명령어 실행됨
- ◆ 여러 명령어 실행을 위해 각 명령어는 “;” 로 구분

## ■ 오타 교정

- ◆ 삭제(erase) 문자
  - 백스페이스 또는 **Ctrl-h** 키
- ◆ 라인킬(linekill) 문자
  - **Ctrl-x, Ctrl-u** 사용
- ◆ **stty** 를 이용한 변경
  - **stty erase [Ctrl-h]**

# 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



# 온라인 매뉴얼

## ■ 매뉴얼 출력(man)

man	
일반형식	man [section] name
주요옵션	section – 아래 구분에서 매뉴얼을 찾아 보여준다

## ■ 편성

구분	주제
1절	<b>Commands (명령어)</b>
2절	<b>System Calls (시스템 호출)</b>
3절	<b>Subroutines (라이브러리 함수)</b>
4절	<b>Special files (특수 파일)</b>
5절	<b>File formats and conventions (파일 형식)</b>
6절	<b>Games (게임)</b>
7절	<b>Macro package and language conventions (기타정보)</b>
8절	<b>Maintenance commands and procedures (보수)</b>

# 온라인 매뉴얼

## ■ 매뉴얼 형식

항목	의미
NAME	명령어의 이름과 기능 기술
SYNOPSIS	명령어의 일반형식 기술
DESCRIPTION	명령어와 옵션의 변화를 더욱더 자세히 설명
OPERANDS	특정연산자 사용방법 기술
ENVIRONMENT VARIABLES	환경변수 기술
EXIT STATUS	실행후 정상종료, 비 정상종료 확인
ATTRIBUTES	속성 정의
SEE ALSO	관련된 명령어, 유용한 정보 제공
NOTES	주의, 또는 특히 알아야 할 사항 제공
USER COMMANDS	사용예제 제공

# 작업종료

---

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

# 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

### ◆ 디렉토리

- 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

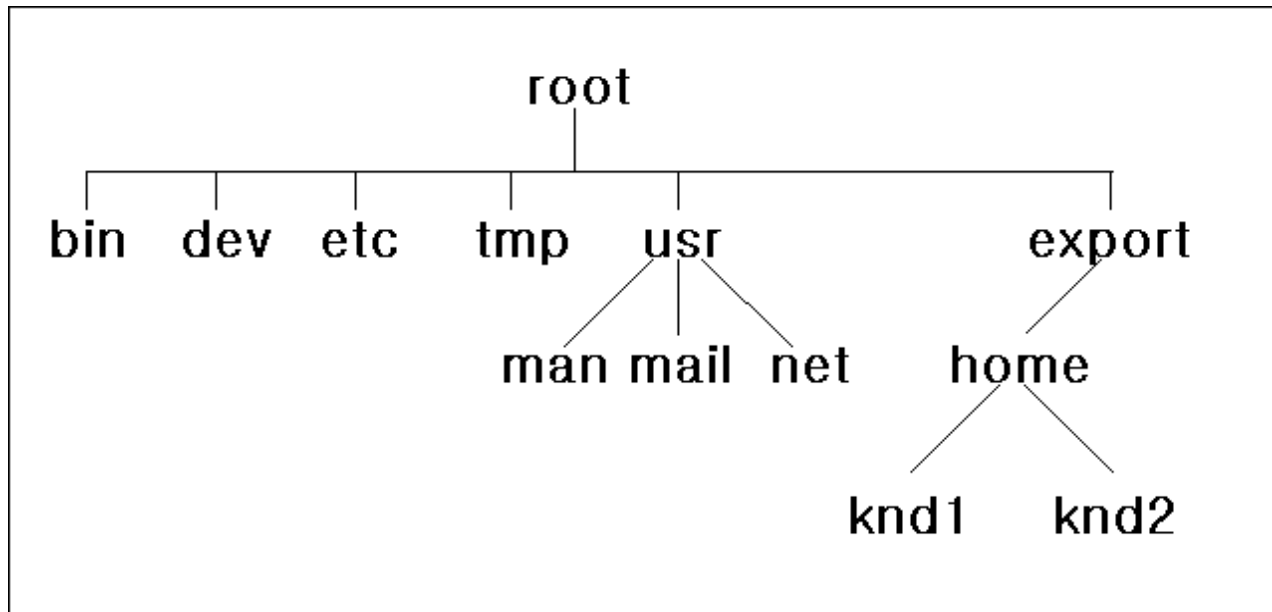
### ◆ 특수화일

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

# Unix/Linux 파일시스템

## ■ 트리(Tree)구조 파일 시스템

- ◆ 파일과 디렉토리의 계층적 구조
- ◆ 최상위는 항상 루트(root) 디렉토리에 기반



# Linux 파일시스템

## ■ 디렉토리의 종류

디렉토리	의미
/bin	각종 기본 프로그램을 포함합니다. 보통 ls, cp 와 같은 명령어 실행 파일이 저장되어 있습니다.
/dev	tty, ram 과 같은 특수 Device 파일을 포함합니다. 또 다른 디바이스를 추가하기 위해서 사용합니다.
/etc	시스템 관리자를 위해 필요한 환경 설정 파일 등을 포함합니다. 가장 대표적인 것이 바로 passwd 파일입니다.
/lib	시스템 프로그램 실행을 위한 각종 라이브러리를 포함합니다.
/mnt	외부 장치인 플로피 디스크, 시디롬, 삼바등을 마운트하기 위해서 제공되는 디렉토리이다.
/proc	시스템 실행 정보를 제공하는 proc 파일 시스템 디렉토리입니다.
/tmp	Unix 시스템이 사용하는 임시파일을 저장합니다.
/usr	시스템에 사용되는 각종 프로그램들이 설치되는 디렉토리이다.

# Unix/Linux 파일시스템

## ■ 홈 디렉토리

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

# 작업 디렉토리 확인

## ■ 현재 작업중인 디렉토리 확인(pwd)

pwd	
일반형식	pwd
주요옵션	

## ■ 사용 예

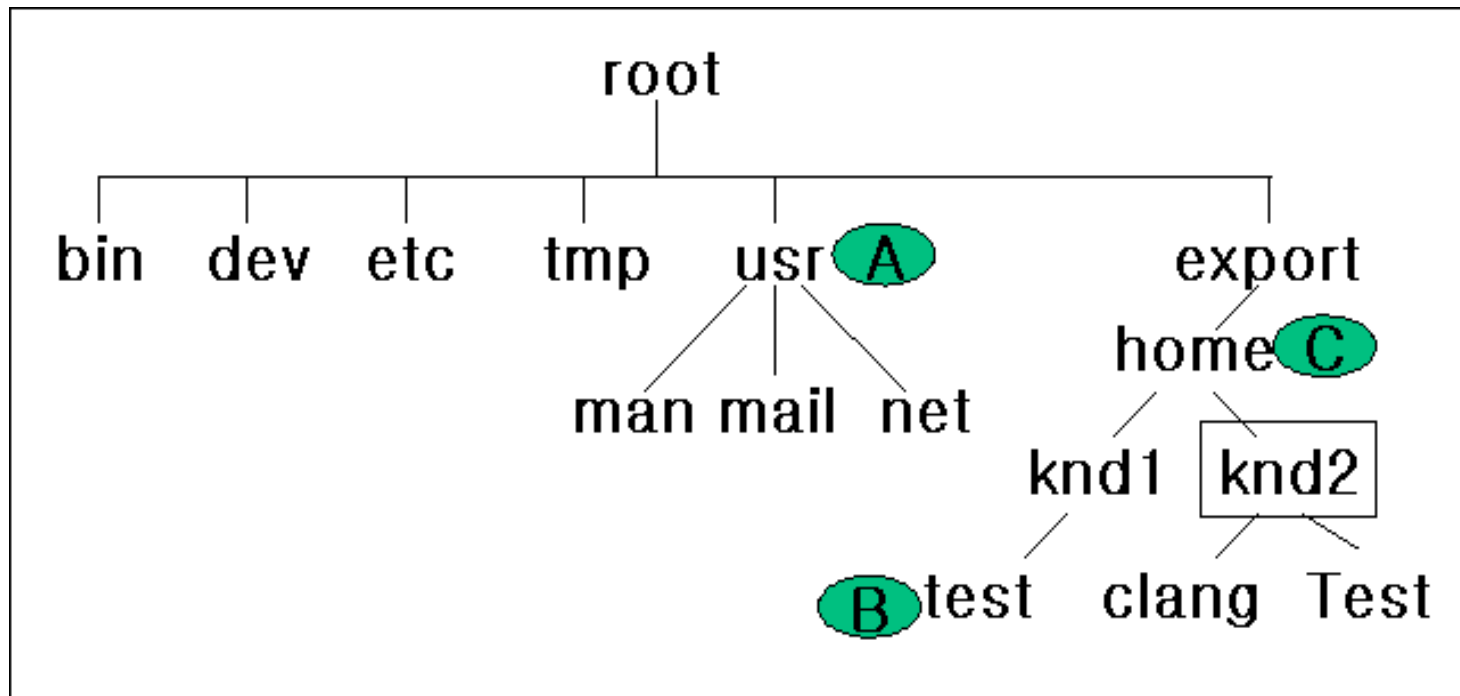
```
[cprog2@seps5 cprog2]$ pwd  
/home/cprog2  
[cprog2@seps5 cprog2]$
```



# 디렉토리 경로

## ■ 절대경로와 상대경로

- ◆ 절대경로 - 모든 경로 명을 반드시 루트디렉토리인 "/"로 시작되어 특정한 파일이나 디렉토리를 표시
- ◆ 상대경로 - 현재 작업 디렉토리에 대해 상대적인 경로이름 이용



# 디렉토리 경로

## ■ 절대경로

디렉토리	절대경로이름
<b>A</b>	/usr
<b>B</b>	/export/home/knd1/test
<b>C</b>	/export/home/home

## ■ 상대경로

디렉토리	상대경로이름
<b>A</b>	../../usr
<b>B</b>	../knd1/test
<b>C</b>	../

# 작업 디렉토리 변경

## ■ 디렉토리 변경(cd)

cd	
일반형식	cd [ directory ]
주요옵션	

## ■ 사용 예

```
[cprog2@seps5 cprog2]$ cd book2
```

```
[cprog2@seps5 book2]$
```

```
[cprog2@seps5 book2]$ cd ~
```

```
[cprog2@seps5 cprog2]$
```

# 디렉토리 파일 목록 보기

## ■ 디렉토리 파일 보기 (ls)

ls	
일반형식	ls [ 옵션 ]
주요옵션	<ul style="list-style-type: none"><li>-a, --all : 디렉토리 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 디렉토리까지 출력</li></ul>

## ■ 사용 예

```
[cprog2@seps5 cprog2]$ ls
```

```
Desktop Mail bashrc book2 packages
```

```
[cprog2@seps5 cprog2]$
```

# 디렉토리 파일 목록 보기

## ■ 사용 예

```
[cprog2@seps5 cprog2]$ ls -l
```

```
합계 20
```

```
drwxr-xr-x    3 cprog2    cprog2    4096 2월    6   12:06 Desktop
drwx-----   7 cprog2    cprog2    4096 6월   16   2003  Mail
-rw-rw-r--    1 cprog2    cprog2    1651 2월    4   17:20  bashrc
drwxrwxr-x   12 cprog2    cprog2    4096 2월    3   11:32  book2
drwxr-xr-x    2 cprog2    cprog2    4096 2월    1   15:22  packages
```

```
[cprog2@seps5 cprog2]$ ls -a
```

```
.                .bash_history  .emacs        kde          .viminfo Mail
..               .bash_logout  .esd_auth     .mcp         .wmrc      bashrc
.ICEauthority    .bash_profile .gdbtkinit    .mcp         .xawtv     book2
.Xauthority      .bashrc       .gimp-1.2     .mozilla.xsession-
errors packages
.ami             .ddd          .gtkrc-kde    .qt          Desktop
[cprog2@seps5 cprog2]$
```

# 파일 복사

## ■ 파일의 복사(cp)

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 디렉토리에 있는 모든 파일을 복사

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

```
[cprog2@seps5 cprog2]$ ls
Desktop  Mail  Test  bashrc  book2  cprog2  packages
[cprog2@seps5 cprog2]$ cp /bin/date Test
[cprog2@seps5 cprog2]$ cd Test
[cprog2@seps5 Test]$ ls
Test.c  date
```

# 파일 삭제

## ■ 파일의 삭제(**rm**)

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

## ■ 사용 예

```
[cprog2@seps5 cprog2]$ cd Test
```

```
[cprog2@seps5 Test]$ ls
```

```
Test.c  date
```

```
[cprog2@seps5 Test]$ rm date
```

```
[cprog2@seps5 Test]$ ls
```

```
Test.c
```

# 디렉토리 생성

## ■ 디렉토리 생성 (mkdir)

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

## ■ 사용 예

```
[cprog2@seps5 cprog2]$ ls
Desktop Mail Test bashrc book2 packages
[cprog2@seps5 cprog2]$ mkdir temp
[cprog2@seps5 cprog2]$ ls
Desktop Mail Test bashrc book2 packages temp
[cprog2@seps5 cprog2]$ mkdir -p Test3/Test5
[cprog2@seps5 cprog2]$ ls
Desktop Mail Test Test3 bashrc book2 packages temp
[cprog2@seps5 cprog2]$ ls Test3
Test5
```



# 디렉토리 삭제

## ■ 디렉토리 삭제(**rmdir**)

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

## ■ 사용 예

```
[cprog2@seps5 Test3]$ ls
```

```
Test5
```

```
[cprog2@seps5 Test3]$ rmdir Test5
```

```
[cprog2@seps5 Test3]$ ls
```

```
[cprog2@seps5 Test3]$
```

# 파일 이름 변경과 옮기기

## ■ 파일 이름 변경과 옮기기(mv)

mv	
일반형식	<code>mv [ -fi ] source dest</code> <code>mv [ -fi] source ... dest_dir</code>
주요옵션	<ul style="list-style-type: none"><li>-b : 대상 파일이 지워지기 전에 백업 파일을 만들</li><li>-f : 대상 파일의 접근허가와 관계없이 무조건 파일을 이동</li><li>-i : 대상 파일이 기존 파일이면, 덮어쓸 것인지 물어봄</li><li>-u : 대상 파일보다 원본 파일이 최근의 것일 때 업그레이드</li><li>-v : 파일 옮기는 과정을 자세하게 보여준다</li></ul>

# 파일 이름 변경과 옮기기

## ■ 사용 예

```
[cprog2@seps5 cprog2]$ ls
Desktop Mail Test bashrc book2 mvTest.c packages
[cprog2@seps5 cprog2]$ mv mvTest.c Test
[cprog2@seps5 cprog2]$ ls
Desktop Mail Test bashrc book2 packages
[cprog2@seps5 cprog2]$ cd Test
[cprog2@seps5 Test]$ ls
Test.c Test2 mvTest.c
[cprog2@seps5 Test]$ cp mvTest.c ../; cd ..
[cprog2@seps5 cprog2]$ mv -i mvTest.c Test
mv: overwrite `Test/mvTest.c'? Y
[cprog2@seps5 cprog2]$ ls
Desktop Mail Test bashrc book2 packages
[cprog2@seps5 cprog2]$
```

# 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

```
[cprog2@seps5 temp]$ ls
[cprog2@seps5 cprog2]$ touch out
[cprog2@seps5 cprog2]$ ls
-rw-rw-r-- 1 cprog2          cprog2  0   9월   3 19:24  out
[cprog2@seps5 cprog2]$ touch out
[cprog2@seps5 cprog2]$ ls
-rw-rw-r-- 1 cprog2          cprog2  0   9월   3 19:30  out
```

# 파일 만들기

## ■ 파일 만들기(cat)

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

## ■ 사용 예

```
[cprog2@seps5 Test]$ cat > catTest.txt
```

```
Hello!
```

```
Nice to meet you. ← Ctrl-D 입력
```

```
[cprog2@seps5 Test]$ ls
```

```
Test.c  Test2  catTest.txt  mvTest.c
```

```
[cprog2@seps5 Test]$
```

# 파일 내용 보기

## ■ 파일의 내용보기(cat, more, 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. \t 로 TAB 문자를 표시함

## ■ 사용 예

```
[cprog2@seps5 Test]$ cat catTest.txt
```

```
Hello!
```

```
Nice to meet you.
```

```
[cprog2@seps5 Test]$ 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>

# 파일 내용 보기

## ■ 파일의 내용보기(head, tail)

head, tail	
일반형식	head [ -number   -n number ] [ file... ] tail [ + number [ lbcf ] ] [ file... ] tail [ -f   -r ] [ -c name   -n number ] [ file... ]
주요옵션	-n : 출력할 라인의 개수이며 기본은 10개의 라인임 -l : 출력크기를 라인단위로 지정 -b : 출력크기를 블록단위로 지정 -c : 출력크기를 문자단위로 지정 -f : 파일 뒤에 추가되는 라인을 계속적으로 보여줌

## ■ 사용 예

```
[cprog2@seps5 Test]$ cat catTest.txt
Hello!
Nice to meet you.
[cprog2@seps5 Test]$ head -1 catTest.txt // 첫번째 줄
Hello!
[cprog2@seps5 Test]$ tail -1 catTest.txt // 마지막 줄
Nice to meet you.
```



# 소유권과 보호

## ■ 파일 접근을 위해 각 파일에 대해 사용권한 설정

◆ 파일/디렉토리에 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 디렉토리를 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 디렉토리를 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

## ■ 사용 예

```
[cprog2@seps5 Test]$ ls -l
```

```
합계 8
```

```
-r-xr-xr-x 1 cprog2          cprog2    0  2월 10 23:01 Test.c
drwxrwxr-x  2 cprog2          cprog24096 2월 10 23:31 Test2
-rw-rw-r--  1 cprog2          cprog2   26  2월 11 00:44 catTest.txt
-rw-rw-r--  1 cprog2          cprog2    0  2월 11 00:17 mvTest.c
```

```
[cprog2@seps5 Test]$
```

```
[cprog2@seps5 Test]$ chmod g+rw Test.c
```

```
[cprog2@seps5 Test]$ ls -l
```

```
합계 8
```

```
-r-xrwxr-x 1 cprog2          cprog2    0  2월 10 23:01 Test.c
drwxrwxr-x  2 cprog2          cprog24096 2월 10 23:31 Test2
-rw-rw-r--  1 cprog2          cprog2   26  2월 11 00:44 catTest.txt
-rw-rw-r--  1 cprog2          cprog2    0  2월 11 00:17 mvTest.c
```

# 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

```
[cprog2@seps5 Test]$ ls -l
```

합계 8

```
-r-xrwxr-x 1 cprog2          cprog2    0  2월 10 23:01 Test.c  
drwxrwxr-x  2 cprog2          cprog24096 2월 10 23:31 Test2  
-rw-rw-r--  1 cprog2          cprog2    26  2월 11 00:44 catTest.txt  
-rw-rw-r--  1 cprog2          cprog2     0  2월 11 00:17 mvTest.c
```

```
[cprog2@seps5 Test]$
```

```
[cprog2@seps5 Test]$ chmod 725 Test.c
```

```
[cprog2@seps5 Test]$ ls -l
```

합계 8

```
-rwx-w-r-x  1 cprog2          cprog2    0  2월 10 23:01 Test.c  
drwxrwxr-x  2 cprog2          cprog24096 2월 10 23:31 Test2  
-rw-rw-r--  1 cprog2          cprog2    26  2월 11 00:44 catTest.txt  
-rw-rw-r--  1 cprog2          cprog2     0  2월 11 00:17 mvTest.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 디렉토리를 포함하여 모드를 변경

## ■ 파일 그룹명 변경(chgrp)

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

# 파일 링크 만들기

## ■ 파일 연결관계(Link) 만들기(ln)

◆ **Unix/Linux** 파일시스템에서 매우 긴 파일이름을 참조하거나, 다른 디렉토리 경로로 연결하여 쉽게 접근할 필요가 있을 때 사용

ln	
일반형식	ln [ -fns ] source_file [ target ] ln [ -fns ] source_file ... target
주요옵션	-f: 접근할 수 없는 사용권한을 가졌을 때도 링크가 가능 -s: 심볼릭 링크 생성

## ■ 사용 예

```
[cprog2@seps5 cprog2]$ ln -s catTest.txt Test.txt
```

```
[cprog2@seps5 cprog2]$ ls -l
```

합계 8

```
-rw-rw-r--      1 cprog2  cprog2      0  9월   6  21:55 Test.c
lrwxrwxrwx      1 cprog2  cprog2     11  9월   6  22:25 Test.txt -> catTest.txt
-rw-rw-r--      1 cprog2  cprog2      0  9월   6  21:54 Test1.c
drwxrwxr-x      2 cprog2  cprog2   4096  9월   6  21:55 Test2
-rw-rw-r--      1 cprog2  cprog2     26  9월   6  21:55 catTest.txt
```



# 디스크 용량 확인

## ■ 디스크 용량 확인 (df)

df	
일반형식	df [ -aikT ] [ -t fstype ]
주요옵션	-a : 디렉토리로 생략된 블록크기의 파일 시스템 출력 -i : inode 로 디스크 정보를 보여준다 -k : 파일의 크기를 kb(킬로바이트) 단위로 보여준다 -T : 파일 시스템의 정보와 디스크 정보를 보여준다 -t : fstype 값을 가진 파일 시스템에 대한 디스크 정보를 보여줌

## ■ 사용 예

```
[cprog2@seps5 cprog2]$ df
Filesystem      K-blocks    Used   Available   Use%    Mounted on
/dev/sdb1        33973700  8099384   24148536    26%     /
/dev/sda1         101089    16290     79580      17%     /boot
none              256620         0    256620      0%     /dev/shm
[cprog2@seps5 cprog2]$ df -a
Filesystem      1K-blocks    Used   Available   Use%    Mounted on
/dev/sdb1        33973700  8099384   24148536    26%     /
none              0           0         0          -       /proc
usbdevfs          0           0         0          -       /proc/bus/usb
/dev/sda1         101089    16290     79580      17%     /boot
none              0           0         0          -       /dev/pts
none              256620         0    256620      0%     /dev/shm
```

# 디스크 사용 공간 확인

## ■ 디스크 사용 공간 확인 (du)

du	
일반형식	du [ -abckx ] filename...
주요옵션	<ul style="list-style-type: none"><li>-a : 모든 파일들의 정보를 보여준다</li><li>-b : 바이트 단위로 보여준다</li><li>-c : 파일의 사용 정보를 보여주고 난 뒤 총 합계를 보여준다</li><li>-k : 킬로바이트 단위로 보여준다</li><li>-x : 심볼릭 링크이면 그 원본 크기를 보여준다</li></ul>

## ■ 사용 예

```
[cprog2@seps5 cprog2]$ du test
4      test/Test2
12     test
[cprog2@seps5 cprog2]$
```

# 프로세스 상태 확인

## ■ 프로세스 상태 확인 (ps)

ps	
일반형식	ps [-] [ options ]
주요옵션	l: 긴 형식으로 출력 u: 실행 유저와 실행 시간 표시 j: 작업 제어(job control) 형식으로 출력 s: 시그널 형식으로 출력 m: 메모리 정보를 보여준다 a: 다른 사용자의 프로세스 현황을 보여준다 x: 터미널 제어 없이 프로세스 현황 표시

## ■ 사용 예

```
[cprog2@seps5 Test]$ ps ax
```

```
PID TTY          STAT       TIME       COMMAND
  1 ?                S            0:07        init
  2 ?                SW            0:00        [migration/0]
```

.....

```
[cprog2@seps5 Test]$ ps aux
```

```
USER  PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root     1   0.0  0.0  1228  464 ?        S    Feb04   0:07   init
root     2   0.0  0.0     0     0 ?        SW    Feb04   0:00 [migration/0]
```

.....

# 프로세스 종료시키기

## ■ 프로세스 종료시키기 (kill)

kill	
일반형식	kill [-s signal ] process_id
주요옵션	-s 또는 -number : 프로세스에 시그널 번호를 보낸다 -9 를 사용할 경우 프로세스를 강제로 종료시킨다

## ■ 사용 예

```
[cprog2@seps5 cprog2]$ sleep 1000 &
[1] 22001
[cprog2@seps5 cprog2]$ kill -9 22001
[cprog2@seps5 cprog2]$
[1]+  죽었음                sleep 1000
[cprog2@seps5 cprog2]$
```