페도라 리눅스

디렉터리와 파일 사용하기

01 리눅스 파일의 종류와 특징

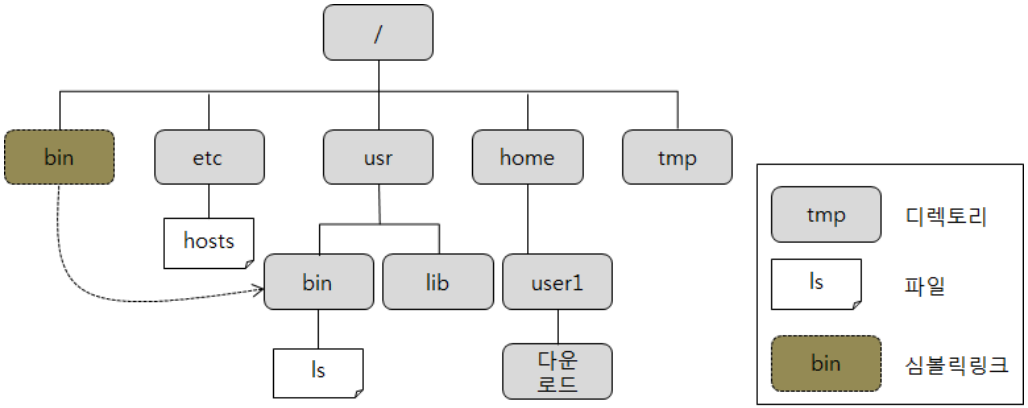
- **파일의 종류 확인: file 명령**

```
[user1@localhost ~]$ file .bash_profile
.bash_profile: ASCII text
[user1@localhost ~]$ file 다운로드
다운로드: directory
[user1@localhost ~]$ file /usr/bin/ls
/usr/bin/ls: ELF 64bit LSB executable,
(생략)
```

01 리눅스 파일의 종류와 특징

디렉터리 계층 구조

- 리눅스에서는 파일을 효율적으로 관리하기 위해 디렉터리를 계층적으로 구성 -> 트리(tree) 구조
- 모든 디렉터리의 출발점은 루트(root, 뿌리) 디렉터리이며, 빗금(/, 슬래시)으로 표시



[그림 2-2] 디렉터리 계층구조 예

- 하위 디렉터리(서브 디렉터리): 디렉터리 아래에 있는 디렉터리 (etc, usr, home, tmp)
- 상위 디렉터리(부모 디렉터리): '..'으로 표시
- 루트 디렉터리를 제외하고 모든 디렉터리에는 부모 디렉터리가 있음

01 리눅스 파일의 종류와 특징

루트 디렉터리의 서브 디렉터리

```
[user1@localhost ~]$ ls -F /
bin@ dev/ home/ lib64@ media/ opt/ root/ sbin@ sys/ usr/
boot/ etc/ lib@ lost+found/ mnt/ proc/ run/ srv/ tmp/ var/
[user1@localhost ~]$
```

- / : 해당 파일이 디렉터리임을 표시
- @: 심볼릭 링크

작업 디렉터리

- 현재 사용 중인 디렉터리를 작업 디렉터리(working directory) 또는 현재 디렉터리(current directory)라고 함
- 현재 디렉터리는 '.' 기호로 표시
- 현재 디렉터리의 위치는 pwd 명령으로 확인

홈 디렉터리

- 각 사용자에게 할당된 디렉터리로 처음 사용자 계정을 만들 때 지정
- 사용자는 자신의 홈 디렉터리 아래에 파일이나 서브 디렉터리를 생성하며 작업 가능
- 홈 디렉터리는 '~' 기호로 표시 : ~user1

01 리눅스 파일의 종류와 특징

[표 2-1] 디렉터리의 주요 기능

디렉터리	기능
dev	장치 파일이 담긴 디렉터리이다.
home	사용자 홈 디렉터리가 생성되는 디렉터리이다.
media	시디롬이나 USB 같은 외부 장치를 연결(마운트라고 함)하는 디렉터리이다.
opt	추가 패키지가 설치되는 디렉터리이다.
root	root 계정의 홈 디렉터리이다. 루트(/) 디렉터리와 다른 것이므로 혼동하지 않도록 한다.
sys	리눅스 커널과 관련된 파일이 있는 디렉터리이다.
usr	기본 실행 파일과 라이브러리 파일, 헤더 파일 등 많은 파일이 있다. 참고로 usr은 Unix System Resource의 약자이다.
boot	부팅에 필요한 커널 파일을 가지고 있다.
etc	리눅스 설정을 위한 각종 파일을 가지고 있다.
lost+found	파일 시스템에 문제가 발생하여 복구할 경우, 문제가 되는 파일이 저장되는 디렉터리로 보통은 비어있다.
mnt	파일 시스템을 임시로 마운팅 하는 디렉터리이다.
proc	프로세스 정보 등 커널 관련 정보가 저장되는 디렉터리이다.
run	실행 중인 서비스와 관련된 파일이 저장된다.
srv	FTP나 Web 등 시스템에서 제공하는 서비스의 데이터가 저장된다.
tmp	시스템 사용 중에 발생하는 임시 데이터가 저장된다. 이 디렉터리에 있는 파일들은 재부팅 하면 모두 삭제된다.

01 리눅스 파일의 종류와 특징

경로명

- 파일 시스템에서 디렉터리 계층 구조에 있는 특정 파일이나 디렉터리의 위치 표시
- 경로명에서 각 경로를 구분하는 구분자로 슬래시(/)를 사용
- 경로명에서 가장 앞에 있는 /는 루트 디렉터리를 뜻하지만 경로명 중간에 있는 /는 구분자
- 예: /usr/bin/ls에서 맨 앞의 /는 루트 디렉터리를 의미하고, 중간에 있는 / 두 개는 디렉터리 이름과 파일 이름을 구분하는 구분자

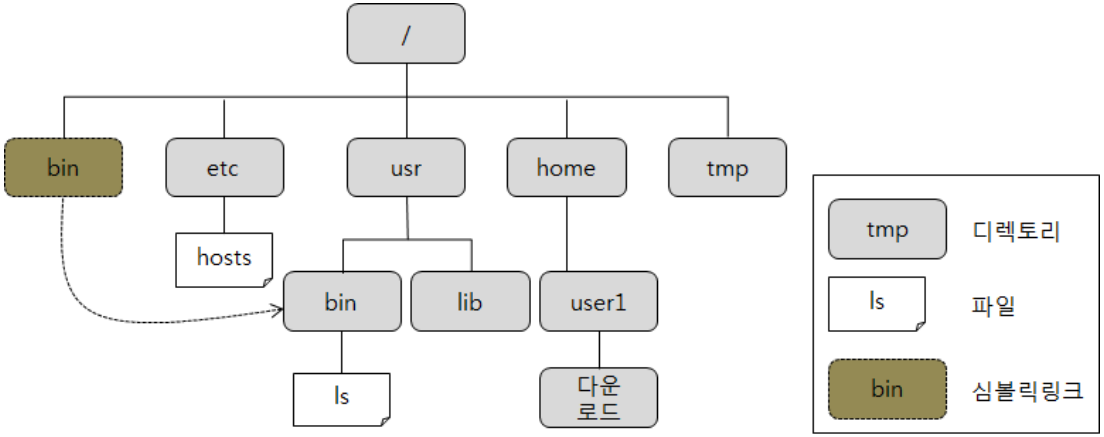
절대 경로명

- 항상 루트(/) 디렉터리부터 시작
- 반드시 /로 시작한다.
- / 디렉터리부터 시작하여 특정 파일이나 디렉터리의 위치까지 이동하면서 거치게 되는 모든 중간 디렉터리의 이름을 표시
- 특정 위치를 가리키는 절대 경로명은 항상 동일

상대 경로명

- 현재 디렉터리를 기준으로 시작
- / 이외의 문자로 시작
- 현재 디렉터리를 기준으로 서브 디렉터리로 내려가면 그냥 서브 디렉터리의 이름을 추가
- 현재 디렉터리를 기준으로 상위 디렉터리로 가려면 ..을 추가
- 상대 경로명은 현재 디렉터리가 어디냐에 따라 달라짐

01 리눅스 파일의 종류와 특징



[그림 2-2] 디렉터리 계층구조 예

현재 디렉터리가 user1일 때

- user1의 절대 경로명: /home/user1
- user1 아래 '다운로드'의 절대 경로명: /home/user1/다운로드
- '다운로드'의 상대 경로명: 다운로드 또는 ./다운로드
- hosts 파일의 상대 경로명: ../../etc/hosts

디렉터리/파일명	절대 경로	상대 경로
/		
home		
tmp		
lib		
ls		

01 리눅스 파일의 종류와 특징

파일과 디렉터리 이름 규칙

- 파일과 디렉터리 이름에는 /를 사용할 수 없다. /는 경로명에서 구분자로 사용하기 때문이다.
- 파일과 디렉터리 이름에는 알파벳, 숫자, 붙임표(-), 밑줄(_), 점(.)만 사용한다.
- 파일과 디렉터리 이름에는 공백 문자, *, |, ", ', @, #, $, %, ^, & 등을 사용하면 안 된다.
- 파일과 디렉터리 이름의 영문자는 대문자와 소문자를 구별하여 다른 글자로 취급한다.
- 파일과 디렉터리 이름이 '.'으로 시작하면 숨김 파일로 간주한다.

파일 이름 예

- 좋은 이름 : game.txt, hello.c, test, sample11
- 나쁜 이름 : &game, *dir, my home, game₩
- 사용할 수 없는 이름 : myhome/, /test, bad/name

02 디렉터리 사용 명령

현재 디렉터리 확인하기

pwd

기능	현재 위치를 확인한다. 즉, 현재 디렉터리의 절대 경로를 출력한다.
형식	pwd

```
[user1@localhost ~]$ pwd
/home/user1
[user1@localhost ~]$
```

02 디렉터리 사용 명령

디렉터리 이동하기

cd

기능	현재 디렉터리를 변경한다.
형식	cd [디렉터리명]
사용 예	cd, cd /tmp cd 다운로드

- 절대 경로명으로 이동할 디렉터리 지정

```
[user1@localhost ~]$ cd /tmp
[user1@localhost tmp]$ pwd
/tmp
[user1@localhost tmp]$
```

- 상대 경로명으로 이동할 디렉터리 지정

```
[user1@localhost tmp]$ cd ../usr/lib
[user1@localhost lib]$ pwd
/usr/lib
[user1@localhost lib]$
```

02 디렉터리 사용 명령

홈 디렉터리로 이동하는 방법

- cd /home/user1 : 절대 경로명을 사용하여 홈 디렉터리로 이동
- cd ../../home/user1 : 현재 /usr/lib 디렉터리에 있었으므로 이를 기준으로 상대 경로명을 사용하여 홈 디렉터리로 이동
- cd ~ : 홈 디렉터리를 나타내는 기호인 ~를 사용하여 홈 디렉터리로 이동
- cd : 목적지를 지정하지 않고 그냥 cd 명령만 사용하면 해당 계정의 홈 디렉터리로 이동

- 이 중 가장 간단한 방법은 당연히 그냥 cd 명령 사용

```
[user1@localhost lib]$ cd
[user1@localhost ~]$ pwd
/home/user1
[user1@localhost ~]$
```

02 디렉터리 사용 명령

■ 디렉터리 내용보기

ls

기능 디렉터리의 내용을 출력한다.

형식 ls [옵션] [파일 또는 디렉터리명]

옵션
- -a : 숨김 파일을 포함하여 모든 파일 목록을 출력한다.
- -d : 지정한 디렉터리 자체의 정보를 출력한다.
- -i : 첫 번째 행에 inode 번호를 출력한다.
- -l : 파일의 상세 정보를 출력한다.
- -A : .(마침표)와 ..(마침표 두 개)를 제외한 모든 파일 목록을 출력한다.
- -F : 파일의 종류를 표시한다(* : 실행 파일, / : 디렉터리, @ : 심벌릭 링크).
- -L : 심벌릭 링크 파일의 경우 원본 파일의 정보를 출력한다.
- -R : 하위 디렉터리 목록까지 출력한다.

사용 예 ls　　ls -F　　ls -al /tmp

02 디렉터리 사용 명령

현재 디렉터리 내용 확인: ls

- 옵션이나 디렉터리를 지정하지 않고 ls 명령만 사용

```
[user1@localhost ~]$ ls
공개  다운로드  문서  바탕화면  비디오  사진  서식  음악
[user1@localhost ~]$
```

숨김 파일 확인: ls –a

- 리눅스에서 .으로 시작하면 숨김 파일이며 그냥 ls 명령으로는 볼 수 없음
- -a 옵션을 사용하면 숨김 파일 확인 가능

```
[user1@localhost ~]$ ls -a
.              .bashrc    .gstreamer-0.10  .pulse-cookie  비디오
..             .cache     .gtk-bookmarks   .speech-dispatcher    사진
.ICEauthority  .config    .local             공개  서식
.bash_history  .esd_auth  .mozilla           다운로드  음악
.bash_logout   .gconf     .pki  문서
.bash_profile  .gphoto    .pulse  바탕화면
[user1@localhost ~]$
```

02 디렉터리 사용 명령

파일의 종류 표시: ls –F

- -F 옵션 : 파일의 종류를 구분하여 표시
- /: 디렉터리, @: 심벌릭 링크, *: 실행파일, 표시없음: 일반파일

```
[user1@localhost ~]$ ls -F
공개/  다운로드/  문서/  바탕화면/  비디오/  사진/  서식/  음악/
[user1@localhost ~]$
```

- -a 옵션과 연결하여 사용

```
[user1@localhost ~]$ ls -aF
./              .bashrc     .gstreamer-0.10/  .pulse-cookie          비디오/
../             .cache/     .gtk-bookmarks    .speech-dispatcher/    사진/
.ICEauthority   .config/    .local/             공개/                  서식/
.bash_history   .esd_auth   .mozilla/           다운로드/              음악/
.bash_logout    .gconf/     .pki/               문서/
.bash_profile   .gphoto/    .pulse/             바탕화면/
[user1@localhost ~]$
```

02 디렉터리 사용 명령

지정한 디렉터리 내용 출력하기

- 인자로 디렉터리 지정하면 해당 디렉터리 내용을 출력

```
[user1@localhost ~]$ ls /tmp
pulse-SfQSCP89Eogc      systemd-private-qVL6BO
pulse-ll1JJ5E7GzfF      yum_save_tx.2013-03-16.22-36.5jeP_F.yumtx
systemd-private-7fCx4q  yum_save_tx.2013-03-18.02-42.rS5itA.yumtx
systemd-private-jqJUdd  yum_save_tx.2013-03-18.02-43.XtCWDC.yumtx
[user1@localhost ~]$
```

- 옵션과 인자를 함께 사용: -F 옵션

```
[user1@localhost ~]$ ls -F /tmp
pulse-SfQSCP89Eogc/      systemd-private-qVL6BO/
pulse-ll1JJ5E7GzfF/      yum_save_tx.2013-03-16.22-36.5jeP_F.yumtx
systemd-private-7fCx4q/  yum_save_tx.2013-03-18.02-42.rS5itA.yumtx
systemd-private-jqJUdd/  yum_save_tx.2013-03-18.02-43.XtCWDC.yumtx
[user1@localhost ~]$
```

02 디렉터리 사용 명령

상세한 정보 출력하기: -l 옵션

- 디렉터리에 있는 파일들의 상세 정보 출력

```
[user1@localhost ~]$ ls -l
합계 32
drwxr-xr-x. 2 user1 user1 4096  2월  5 22:14 공개
drwxr-xr-x. 2 user1 user1 4096  2월  5 22:14 다운로드
drwxr-xr-x. 2 user1 user1 4096  2월  5 22:14 문서
drwxr-xr-x. 2 user1 user1 4096  2월  5 22:14 바탕화면
drwxr-xr-x. 2 user1 user1 4096  2월  5 22:14 비디오
drwxr-xr-x. 2 user1 user1 4096  2월  5 22:14 사진
drwxr-xr-x. 2 user1 user1 4096  2월  5 22:14 서식
drwxr-xr-x. 2 user1 user1 4096  2월  5 22:14 음악
[user1@localhost ~]$
```

[표 2-2] 파일 상세 정보

필드번호	필드 값	의미
1	d	파일 종류
2	rwxr-xr-x	접근권한
3	2	하드링크 개수
4	user1	파일 소유자
5	user1	파일이 속한 그룹
6	4096	파일크기(바이트)
7	2월 5 22:14	마지막 수정시간
8	공개	파일 이름

[표 2-3] 파일 종류

문자	파일 종류
-	일반 파일
d	디렉터리
l	심벌릭 링크
b	블록 장치 파일
c	문자 장치 파일
p	파이프 파일
s	소켓 파일

02 디렉터리 사용 명령

디렉터리 자체 정보 확인: -d 옵션

- 디렉터리의 자체 정보 출력

```
[user1@localhost ~]$ ls -l /
합계 62
lrwxrwxrwx.   1 root root     7    2월  5 21:38 bin -> usr/bin
dr-xr-xr-x.   4 root root  1024    2월  5 22:03 boot
drwxr-xr-x.  19 root root  3380    3월 16 22:30 dev
drwxr-xr-x. 127 root root 12288    3월 18 04:48 etc
(생략)
[user1@localhost ~]$ ls -ld /
dr-xr-xr-x. 18 root root 4096    3월 17 15:05 /
[user1@localhost ~]$
```

파일 존재 확인

- 인자로 지정한 파일이 없으면 없다는 메시지 출력

```
[user1@localhost ~]$ ls .bash_profile
.bash_profile
[user1@localhost ~]$ ls game
ls: cannot access game: 그런 파일이나 디렉터리가 없습니다
[user1@localhost ~]$
```

02 디렉터리 사용 명령

ls 명령의 심벌릭 링크 : dir, vdir

- 윈도의 cmd 창에서 사용하는 명령과 동일

```
[user1@localhost ~]$ dir
공개  다운로드  문서  바탕화면  비디오  사진  서식  음악
[user1@localhost ~]$ vdir
합계 32
drwxr-xr-x. 2 user1 user1 4096    2월  5 22:14 공개
drwxr-xr-x. 2 user1 user1 4096    2월  5 22:14 다운로드
drwxr-xr-x. 2 user1 user1 4096    2월  5 22:14 문서
drwxr-xr-x. 2 user1 user1 4096    2월  5 22:14 바탕화면
drwxr-xr-x. 2 user1 user1 4096    2월  5 22:14 비디오
drwxr-xr-x. 2 user1 user1 4096    2월  5 22:14 사진
drwxr-xr-x. 2 user1 user1 4096    2월  5 22:14 서식
drwxr-xr-x. 2 user1 user1 4096    2월  5 22:14 음악
[user1@localhost ~]$
```

02 디렉터리 사용 명령

디렉터리 만들기

mkdir

기능	디렉터리를 생성한다.
형식	mkdir [옵션] 디렉터리명
옵션	-p : 하위 디렉터리를 계층적으로 생성할 때 중간의 디렉터리가 없으면 생성하면서 전체 디렉터리를 생성한다.
사용 예	mkdir temp

02 디렉터리 사용 명령

디렉터리 한 개 만들기

- 디렉터리를 한 개만 만들려면 mkdir 명령에 인자로 생성하려는 디렉터리명을 지정
- 디렉터리명은 상대 경로명이나 절대 경로명으로 지정

```
[user1@localhost ~]$ mkdir temp
[user1@localhost ~]$ ls temp
[user1@localhost ~]$ ls
temp  공개  다운로드  문서  바탕화면  비디오  사진  서식  음악
[user1@localhost ~]$
```

동시에 디렉터리 여러 개 만들기

- 디렉터리 이름을 여러 개 지정하면 동시에 만들수 있음
- 디렉터리 이름은 공백 문자로 구분

```
[user1@localhost ~]$ mkdir tmp1 tmp2 tmp3
[user1@localhost ~]$ ls
temp  tmp2  공개      문서      비디오  서식
tmp1  tmp3  다운로드  바탕화면  사진    음악
[user1@localhost ~]$
```

02 디렉터리 사용 명령

중간 디렉터리 자동으로 만들기 : -p 옵션

- -p 옵션: 디렉터리명으로 지정한 경로 중 중간 단계의 디렉터리가 없을 경우 자동으로 중간 단계 디렉터리를 생성한 후 최종 디렉터리를 생성
- 예: 경로에서 중간 단계 디렉터리가 없으므로 디렉터리를 생성 못함

```
[user1@localhost ~]$ mkdir temp/mid/han
mkdir: `temp/mid/han' 디렉터리를 만들 수 없습니다: 그런 파일이나 디렉터리가 없습니다
[user1@localhost ~]$
```

- 예: -p 옵션 사용

```
[user1@localhost ~]$ mkdir -p temp/mid/han
[user1@localhost ~]$ ls -R temp
temp:
mid
temp/mid:
han
temp/mid/han:
[user1@localhost ~]$
```

02 디렉터리 사용 명령

■ 디렉터리 삭제하기

rmdir

기능	디렉터리를 삭제한다.
형식	rmdir [옵션] 디렉터리명
옵션	-p : 지정한 디렉터리를 삭제한 뒤, 그 디렉터리의 부모 디렉터리가 빈 디렉터리일 경우 부모 디렉터리도 자동으로 삭제한다.
사용 예	rmdir temp

- 예: tmp3 디렉터리 삭제

```
[user1@localhost ~]$ rmdir tmp3
[user1@localhost ~]$ ls
temp  tmp1  tmp2  공개  다운로드  문서  바탕화면  비디오  사진  서식  음악
[user1@localhost ~]$
```

- 디렉터리가 비어있지 않으면 삭제 불가

```
[user1@localhost ~]$ rmdir temp
rmdir: failed to remove 'temp': 디렉터리가 비어 있지 않음
[user1@localhost ~]$
```

02 디렉터리 사용 명령

실습

① 현재 위치를 확인한다. 홈 디렉터리가 아니면 홈 디렉터리로 이동한다.

② 실습을 위한 기본 디렉터리를 만든다.

③ ch2 디렉터리를 만들고 그 디렉터리로 이동하여 현재 위치를 확인한다.

④ one, two, three 디렉터리를 동시에 만든다.

⑤ one 디렉터리 아래에 tmp/test 디렉터리를 만든다. 중간 경로인 tmp 디렉터리가 자동 생성되도록 한다.

⑥ two, three 디렉터리를 동시에 삭제한다.

⑦ 실습을 마치고 홈 디렉터리로 이동한다.

03 파일 다루기

- 파일의 내용을 보는 명령
- 파일을 복사하는 명령
- 파일을 삭제하고 이동하는 명령
- 하드 링크와 심벌릭 링크를 생성하는 명령
- 빈 파일을 만드는 명령

03 파일 다루기

파일 내용 연속 출력하기

- 텍스트 파일 내용 확인

cat	
기능	파일 내용을 출력한다.
형식	cat [옵션] 파일명...
옵션	-n : 행 번호를 붙여서 출력한다.
사용 예	cat file1 cat -n file1

- 예: /etc/hosts 파일 내용 확인

```
[user1@localhost ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
[user1@localhost ~]$
```

- 예: 행 번호 붙이기(-n 옵션)

```
[user1@localhost ~]$ cat -n /etc/hosts
  1  127.0.0.1  localhost localhost.localdomain localhost4 localhost4.localdomain4
  2  ::1  localhost localhost.localdomain localhost6 localhost6.localdomain6
[user1@localhost ~]$
```

03 파일 다루기

화면 단위로 파일 내용 출력하기

more

기능	파일 내용을 화면 단위로 출력한다.
형식	more [옵션] 파일명...
옵션	+ 행 번호 : 출력을 시작할 행 번호를 지정한다.
사용 예	more file1

- 아직 출력되지 않은 내용이 더 있으면 화면 하단에 '--More--(0%)'와 같이 표시
- 예: /etc/services 파일 내용 보기

```
[user1@localhost ~]$ more /etc/services
# /etc/services:
# $Id: services,v 1.53 2011/06/13 15:00:06 ovasik Exp $
#
# Network services, Internet style
(생략)
#
# service-name  port/protocol  [aliases ...]   [# comment]
tcpmux          1/tcp                           # TCP port service multiplexer
--More--(0%)
```

- 스페이스바: 다음 화면 출력, 엔터키: 한 줄씩 스크롤, /문자열: 해당 문자열 검색, q: 종료

03 파일 다루기

■ 개선된 화면 단위 파일 내용 출력하기

- 스크롤 되어 지나간 내용도 확인 가능

less

기능	파일 내용을 화면 단위로 출력한다.
형식	less 파일명...
사용 예	less file1

[표 2-4] less 명령에서 사용하는 키와 동작

키	동작
j	한 줄씩 다음 행으로 스크롤한다.
k	한 줄씩 이전 행으로 스크롤한다.
Space Bar, ^f	다음 화면으로 이동한다.
^b	이전 화면으로 이동한다.

- 예: /etc/services

```
[user1@localhost ~]$ less /etc/services
# /etc/services:
# $Id: services,v 1.53 2011/06/13 15:00:06 ovasik Exp $
#
# Network services, Internet style
(생략)
#
# service-name  port/protocol  [aliases ...]   [# comment]
tcpmux          1/tcp                           # TCP port service multiplexer
/etc/services
```

03 파일 다루기

■ 파일 뒷부분 출력하기

tail

기능	파일의 뒷부분 몇 행을 출력한다.
형식	tail [옵션] 파일명...
옵션	+행 번호 : 지정한 행부터 끝까지 출력한다. -숫자 : 화면에 출력할 행의 수를 지정한다(기본 값은 10). -f : 파일 출력이 종료되지 않고 주기적으로 계속 출력한다.

- 예: /etc/services 파일의 마지막 10행 출력

```
[user1@localhost ~]$ tail /etc/services
3gpp-cbsp       48049/tcp  # 3GPP Cell Broadcast Service Protocol
isnetserv       48128/tcp  # Image Systems Network Services
isnetserv       48128/udp  # Image Systems Network Services
blp5            48129/tcp  # Bloomberg locator
blp5            48129/udp  # Bloomberg locator
com-bardac-dw   48556/tcp  # com-bardac-dw
com-bardac-dw   48556/udp  # com-bardac-dw
iqobject        48619/tcp  # iqobject
iqobject        48619/udp  # iqobject
matahari        49000/tcp  # Matahari Broker
[user1@localhost ~]$
```

03 파일 다루기

파일 뒷부분 출력하기

- 지정한 숫자만큼 출력하기 : - 숫자 옵션
- 예: /etc/services 파일의 마지막 7

```
[user1@localhost ~]$ tail -7 /etc/services
blp5            48129/tcp  # Bloomberg locator
blp5            48129/udp                  # Bloomberg locator
com-bardac-dw   48556/tcp  # com-bardac-dw
com-bardac-dw   48556/udp  # com-bardac-dw
iqobject        48619/tcp  # iqobject
iqobject        48619/udp  # iqobject
matahari        49000/tcp  # Matahari Broker
[user1@localhost ~]$
```

- 파일 내용을 주기적으로 반복 출력하기 : -f 옵션
  - -f 옵션을 사용하면 파일 출력이 종료되지 않고 대기 상태가 되며 파일 내용이 주기적으로 반복 출력

```
[user1@localhost ~]$ tail -f /etc/services
3gpp-cbsp       48049/tcp                  # 3GPP Cell Broadcast Service Protocol
isnetserv       48128/tcp                  # Image Systems Network Services
(생략)
iqobject        48619/udp                  # iqobject
matahari        49000/tcp                  # Matahari Broker
^C
[user1@localhost ~]$tcp  # Matahari Broker
[user1@localhost ~]$
```

03 파일 다루기

파일(디렉터리) 복사하기

cp

기능	파일이나 디렉터리를 복사한다.
형식	cp [옵션] 파일명1/디렉터리명1 파일명2/디렉터리명2
옵션	-i : 대화식 복사 방법으로 파일명2가 이미 존재할 경우 덮어쓸 것인지 물어본다. -r : 디렉터리를 복사할 때 지정한다.
사용 예	cp file1 file2 cp f1 f2 f3 dir1 cp -r dir1 dir2

두 인자가 모두 파일인 경우 : 파일을 다른 파일로 복사

- 예: /etc/hosts 파일을 현재 디렉터리에 text1 파일로 복사

```
[user1@localhost ch2]$ ls
[user1@localhost ch2]$ cp /etc/hosts text1
[user1@localhost ch2]$ ls
text1
[user1@localhost ch2]$
```

03 파일 다루기

두 번째 인자가 디렉터리인 경우

- 파일을 해당 디렉터리 아래에 복사
- 예: temp 디렉터리에 text1 파일 복사

```
[user1@localhost ch2]$ mkdir temp
[user1@localhost ch2]$ cp text1 temp
[user1@localhost ch2]$ ls temp
text1
[user1@localhost ch2]$
```

- 예: 원본 파일과 다른 이름으로 복사

```
[user1@localhost ch2]$ cp text1 temp/text2
[user1@localhost ch2]$ ls temp
text1  text2
[user1@localhost ch2]$
```

- 예: 쓰기 권한이 없는 디렉터리에 파일을 복사하려고 하면 다음과 같은 오류가 발생

```
[user1@localhost ch2]$ cp text1 /etc
cp: cannot create regular file '/etc/text1': 허가 거부
[user1@localhost ch2]$
```

03 파일 다루기

인자를 여러 개 지정할 경우

- cp 명령에서 첫 번째 인자의 자리에 파일명을 여러 개 지정할 수 있는데, 두 번째 인자는 반드시 디렉터리여야 한다.
- 이럴 경우 마지막에 지정한 디렉터리로 앞서 지정한 파일들이 모두 복사된다.
- 예: /etc/hosts와 /etc/services를 temp 디렉터리에 복사

```
[user1@localhost ch2]$ cp /etc/hosts /etc/services temp
[user1@localhost ch2]$ ls temp
hosts  services  text1  text2
[user1@localhost ch2]$
```

-i 옵션 사용하기

- 두 번째 인자로 지정한 파일명이 이미 있는 파일인 경우 덮어서 복사할 것인지 물어봄

```
[user1@localhost ch2]$ cp -i /etc/hosts text1
cp: overwrite `text1'? n
[user1@localhost ch2]$
```

03 파일 다루기

디렉터리 복사하기

- 디렉터리를 복사하려면 -r 옵션 사용
- 예: -r 옵션을 지정하지 않을 경우

```
[user1@localhost ch2]$ cp temp temp2
cp: omitting directory `temp'
[user1@localhost ch2]$
```

- 두 번째 인자로 지정한 목적지 디렉터리가 존재하지 않는 경우 새로 생성
- 디렉터리가 복사되면 원본 디렉터리 아래에 있던 모든 내용도 함께 복사
- 예: temp 디렉터리를 temp2 디렉터리로 복사

```
[user1@localhost ch2]$ cp -r temp temp2
[user1@localhost ch2]$ ls temp2
hosts  services  text1  text2
[user1@localhost ch2]$
```

- 두 번째 인자로 지정한 디렉터리가 이미 있는 디렉터리일 경우, 원본 디렉터리가 목적지 디렉터리 아래에 원본 디렉터리와 같은 이름으로 복사
- 예: temp 디렉터리를 다시 temp2 디렉터리로 복사(이미 앞에서 temp2 디렉터리가 생성되었으므로 이번에는 temp 디렉터리가 temp2 디렉터리 아래에 복사)

```
[user1@localhost ch2]$ cp -r temp temp2
[user1@localhost ch2]$ ls temp2
hosts  services  temp  text1  text2
[user1@localhost ch2]$
```

03 파일 다루기

파일 이동하기

mv

기능	파일을 이동한다.
형식	mv [옵션] 파일명1/디렉터리명1 파일명2/디렉터리명2
옵션	-i : 파일명2/디렉터리명2가 존재하면 덮어쓸 것인지 물어본다.
사용 예	mv file1 file2

- 파일을 다른 디렉터리로 이동하거나 파일명을 바꿀 때는 mv(move) 명령을 사용
- 디렉터리를 이동하거나 디렉터리명을 바꿀 때도 mv 명령을 사용
- mv 명령의 첫 번째 인자는 원본 파일명이나 디렉터리명을 지정하며, 두 번째 인자는 목적지 파일명이나 디렉터리명을 지정

03 파일 다루기

파일을 파일로 이동하기

- 파일을 다른 파일로 이동하는 것은 결국 원본 파일의 파일명을 다른 파일명으로 바꾸는 것
- 만약 두 번째 인자로 지정한 파일명이 이미 존재하는 파일이면 원본 파일의 내용으로 덮어쓰고 기존의 내용이 삭제
- 두 번째 인자로 지정한 파일명이 존재하지 않는 파일이라면 새 파일이 생성
- 예: text1 파일을 data1 파일로 이동(파일명 변경)

```
[user1@localhost ch2]$ mv text1 data1
[user1@localhost ch2]$ ls
data1  temp                    temp2
[user1@localhost ch2]$
```

03 파일 다루기

파일을 다른 디렉터리로 이동하기

- 두 번째 인자로 디렉터리를 지정할 경우 원본 파일을 지정한 디렉터리로 이동
- 예: data1 파일을 temp 디렉터리로 이동

```
[user1@localhost ch2]$ mv data1 temp
[user1@localhost ch2]$ ls
temp  temp2
[user1@localhost ch2]$ ls temp
data1  hosts  services  text1  text2
[user1@localhost ch2]$
```

- 두 번째 인자에 디렉터리와 파일명을 함께 지정할 경우, 파일이 지정한 디렉터리로 이동하면 파일명도 바뀌게 됨

```
[user1@localhost ch2]$ cp temp/data1 text1
[user1@localhost ch2]$ ls
temp                temp2  text1
[user1@localhost ch2]$ mv text1 temp/data2
[user1@localhost ch2]$ ls temp
data1  data2  hosts  services  text1  text2
[user1@localhost ch2]$
```

- 쓰기 권한이 없는 디렉터리로 파일을 이동하려고 할 경우 오류 발생

```
[user1@localhost ch2]$ mv temp/data2 /etc
mv: cannot move `temp/data2' to `/etc/data2': 허가 거부
[user1@localhost ch2]$
```

03 파일 다루기

파일 여러 개를 디렉터리로 이동하기

- mv 명령으로 파일 여러 개를 지정한 디렉터리로 한 번에 이동 가능
- 두 번째 인자는 반드시 디렉터리여야 함

```
[user1@localhost ch2]$ ls temp
data1  data2  hosts  services  text1  text2
[user1@localhost ch2]$ mv temp/data1 temp/data2 .
[user1@localhost ch2]$ ls
data1  data2  temp            temp2
[user1@localhost ch2]$ ls temp
hosts  services  text1  text2
[user1@localhost ch2]$
```

-i 옵션 사용하기

- 두 번째 인자에 지정한 파일명이 기존에 있는 파일일 경우 덮어서 이동할 것인지를 물어봄

```
[user1@localhost ch2]$ mv -i data1 data2
mv: overwrite `data2'? n
[user1@localhost ch2]$ ls
data1  data2  temp            temp2
[user1@localhost ch2]$
```

03 파일 다루기

디렉터리를 디렉터리로 이동하기

- 인자를 모두 디렉터리로 지정하면 디렉터리가 이동
- 두 번째 인자가 기존에 있던 디렉터리가 아닐 경우에는 디렉터리명이 변경
- 예: temp2 디렉터리가 temp3 디렉터리로 이름 변경

```
[user1@localhost ch2]$ mv temp2 temp3
[user1@localhost ch2]$ ls
data1  data2  temp            temp3
[user1@localhost ch2]$
```

- 두 번째 인자가 기존에 있던 디렉터리일 경우, 원본 디렉터리가 두 번째 인자로 지정된 디렉터리 아래로 이동
- 예: temp3 디렉터리가 temp 디렉터리 아래로 이동

```
[user1@localhost ch2]$ ls
data1  data2  temp            temp3
[user1@localhost ch2]$ mv temp3 temp
[user1@localhost ch2]$ ls
data1  data2  temp
[user1@localhost ch2]$ ls temp
hosts  services  temp3  text1  text2
[user1@localhost ch2]$
```

03 파일 다루기

파일 삭제하기

rm

기능	파일을 삭제한다.
형식	rm [옵션] 파일명/디렉터리명 ...
옵션	-i : 대화식으로 지정한 파일을 정말 삭제할 것인지 확인한다.
	-r : 디렉터리를 삭제할 때 지정한다.
사용 예	rm file rm -r dir

- 삭제할 파일을 인자로 지정하면 해당 파일이 삭제
- 바로 삭제되어 복구할 수도 없으므로 파일을 삭제할 때는 신중해야 함
- 예: data2 파일 삭제

```
[user1@localhost ch2]$ ls
data1  data2  temp
[user1@localhost ch2]$ rm data2
[user1@localhost ch2]$ ls
data1  temp
[user1@localhost ch2]$
```

03 파일 다루기

-i 옵션 사용하기

- -i 옵션을 지정하고 rm 명령을 사용하면 정말 삭제할 것인지 물어봄

```
[user1@localhost ch2]$ rm -i data1
rm: remove 일반 파일 `data1'? n
[user1@localhost ch2]$ ls
data1  temp
[user1@localhost ch2]$
```

디렉터리 삭제하기

- rm 명령으로 디렉터리를 지울 때는 -r 옵션을 지정(삭제된 디렉터리는 복구 불가능)
- 예: -r 옵션을 지정하지 않을 경우 오류 메시지 출력

```
[user1@localhost ch2]$ cd temp
[user1@localhost temp]$ ls
hosts  services  temp3  text1  text2
[user1@localhost temp]$ rm temp3
rm: cannot remove `temp3': 디렉터리입니다
[user1@localhost temp]$
```

03 파일 다루기

디렉터리 삭제하기

- 예: rmdir 명령으로 temp3을 삭제하려고 하면 temp3 디렉터리가 비어 있지 않다고 오류 메시지 출력

```
[user1@localhost temp]$ rmdir temp3
rmdir: failed to remove `temp3': 디렉터리가 비어 있지 않음
[user1@localhost temp]$ ls temp3
hosts  services  temp  text1  text2
[user1@localhost temp]$
```

- 예: -r 옵션 지정

```
[user1@localhost temp]$ ls
hosts  services  temp3  text1  text2
[user1@localhost temp]$ rm -r temp3
[user1@localhost temp]$ ls
hosts  services  text1  text2
[user1@localhost temp]$
```

03 파일 다루기

디렉터리 삭제하기

- -i 옵션을 사용: 삭제하려는 디렉터리 아래에 있는 파일이나 서브 디렉터리를 삭제할 것인지 계속 물어봄

```
[user1@localhost temp]$ cd ..
[user1@localhost ch2]$ ls
data1  temp
[user1@localhost ch2]$ rm -ri temp
rm: descend into directory `temp'? y
rm: remove 일반 파일 `temp/text2'? y
rm: remove 일반 파일 `temp/services'? n
rm: remove 일반 파일 `temp/hosts'? n
rm: remove 일반 파일 `temp/text1'? n
rm: remove 디렉터리 `temp'? n
[user1@localhost ch2]$ ls temp
hosts  services  text1
[user1@localhost ch2]$
```

03 파일 다루기

빈 파일 생성하기, 수정 시간 변경하기 : touch

touch

기능	빈 파일을 생성한다.
형식	touch [-acm] [-r ref_file ¦ -t time] 파일
옵션	-a : 접근 시간만 변경한다. -m : 수정 시간만 변경한다. -t [[CC]YY]MMDDhhmm[.ss] : 시간을 직접 입력한다.
사용 예	touch test

- 인자를 지정하지 않으면 빈 파일 생성

```
[user1@localhost ch2]$ touch test
[user1@localhost ch2]$ ls -l test
-rw-rw-r--. 1 user1 user1 0  3월 19 05:06 test
[user1@localhost ch2]$
```

03 파일 다루기

빈 파일 생성하기, 수정 시간 변경하기 : touch

- 이미 있는 파일을 touch 명령으로 옵션 없이 사용하면 파일의 수정 시간이 현재 시간으로 변경
- 예: data1.cp의 수정 시간을 touch 명령을 사용하여 현재 시간으로 변경

```
[user1@localhost ch2]$ ls -l data1.cp
-rw-r--r--. 1 user1 user1  158  3월 19 03:41 data1.cp
[user1@localhost ch2]$ date
2013. 03. 19. (화) 05:08:52 KST
[user1@localhost ch2]$ touch data1.cp
[user1@localhost ch2]$ ls -l data1.cp
-rw-r--r--. 1 user1 user1  158  3월 19 05:09 data1.cp
[user1@localhost ch2]$
```

- -t 옵션 사용하여 변경할 시간 지정 가능

```
[user1@localhost ch2]$ ls -l test
-rw-rw-r--. 1 user1 user1    0  3월 19 05:06 test
[user1@localhost ch2]$ touch -t 01011200 test
[user1@localhost ch2]$ ls -l test
-rw-rw-r--. 1 user1 user1    0  1월  1 12:00 test
[user1@localhost ch2]$
```

시간 표시

형식	[[CC]YY]MMDDhhmm[.ss]..
설명	-CC : 연도의 첫 두 자리 -YY : 연도의 마지막 두 자리 -MM : 달(01~12 범위 내 지정) -DD : 날짜(01~31 범위 내 지정) -hh : 시간(00~23 범위 내 지정) -mm : 분(00~59 범위 내 지정) -ss : 초(00~59 범위 내 지정)

[표 2-5] 연도 지정 방법

YY	69 – 99	00 – 38	39 – 68
CC	19	20	ERROR

03 파일 다루기

■ 파일 내용 검색하기 : grep

grep

기능 지정한 패턴을 포함하는 행을 찾는다.

형식 grep [옵션] 패턴 [파일명]

옵션 -i : 대 · 소문자를 모두 검색한다.
-l : 해당 패턴을 포함하는 파일 이름을 출력한다.
-n : 행 번호를 출력한다.

사용 예
```
grep root /etc/passwd
grep -n unix ~/".txt
grep -l hello ".c
```

- 예: 인자로 지정한 문자열 검색과 행의 줄 번호 출력(-n)

```
[user1@localhost ch2]$ cp /etc/services data
[user1@localhost ch2]$ grep DHCP data
dhcp-failover   647/tcp                 # DHCP Failover
dhcp-failover   647/udp                 # DHCP Failover
[user1@localhost ch2]$
[user1@localhost ch2]$ grep -n DHCP data
1436:dhcp-failover   647/tcp                 # DHCP Failover
1437:dhcp-failover   647/udp                 # DHCP Failover
[user1@localhost ch2]$
```

03 파일 다루기

파일 찾기 : find

find

기능 조건에 맞는 파일을 지정한 위치에서 찾는다.

형식 find 경로 검색 조건 [동작]

옵션
-name *filename* : 파일 이름으로 검색한다.
-type *파일 종류* : 파일의 종류로 검색한다.
-user *loginID* : 지정한 사용자가 소유한 모든 파일을 검색한다.
-perm *접근 권한* : 지정한 사용 권한과 일치하는 파일을 검색한다.

동작
-exec 명령 {} \; : 검색된 파일에 명령을 실행한다.
-ok 명령 {}\; : 사용자의 확인을 받아서 명령을 실행한다.
-print : 검색된 파일의 절대 경로명을 화면에 출력한다(기본 동작).
-ls : 검색 결과를 긴 목록 형식으로 출력한다.

사용 예
```
$ find ~ -name hello.c
$ find /tmp -user user10 -exec rm {}\;
```

03 파일 다루기

파일 찾기 : find

- 예: /usr 디렉터리에서 ls 파일의 위치를 검색
  - 접근 권한이 없는 디렉터리는 검색할 수 없어서 '허가 거부' 메시지가 출력

```
[user1@localhost ch2]$ find /usr -name ls
find: `/usr/lib/firewalld': 허가 거부
find: `/usr/share/polkit-1/rules.d': 허가 거부
find: `/usr/lib64/audit': 허가 거부
/usr/bin/ls
[user1@localhost ch2]$
```

- 특정 사용자 계정이 소유자인 파일을 찾고 싶으면 다음 예와 같이 -user 옵션을 사용

```
[user1@localhost ch2]$ find /home -user user1
/home/user1
/home/user1/.pulse-cookie
/home/user1/음악
/home/user1/temp
(생략)
```

03 파일 다루기

파일 찾기 : find

- find 명령으로 검색한 모든 파일을 대상으로 동일한 작업을 수행하려면 -exec나 -ok 옵션 사용
- 예: /tmp 디렉터리 아래에 있는 user1 계정 소유의 파일을 전부 찾아서 삭제할 경우
  - find 명령으로 찾은 파일의 절대 경로가 exec 다음의 { }가 있는 위치에 삽입되어 명령이 처리
  - rm 명령과 { } 사이, { }와 \ 사이에 공백이 있어야 하며, \과 ;은 공백 없이 붙어야 함

```
[user1@localhost ch2]$ find /tmp -user user1 -exec rm {} ₩;
rm: cannot remove `/tmp/.esd-1000': 디렉터리입니다
rm: cannot remove `/tmp/pulse-SfQSCP89Eogc': 디렉터리입니다
find: `/tmp/pulse-ll1JJ5E7GzfF': 허가 거부
[user1@localhost ch2]$
```

- find 명령으로 검색한 파일을 삭제하기 전에 하나씩 확인하고 싶으면 -exec 대신 -ok를 사용

```
[user1@localhost ch2]$ find ./temp -user user1 -ok rm {} ₩;
< rm ... ./temp > ? y
rm: cannot remove `./temp': 디렉터리입니다
< rm ... ./temp/data1.cp > ? y
< rm ... ./temp/services > ?
```

03 파일 다루기

명령의 위치 찾기 : whereis

- /bin, /usr/bin, /etc, /usr/etc, /sbin, /usr/sbin, /usr/share/man 등 정해진 디렉터리를 검색하여 명령의 위치검색

whereis

기능	지정된 경로에서 명령의 바이너리 파일이나 매뉴얼 파일의 위치를 찾는다.
형식	whereis [옵션] 명령
옵션	-b : 바이너리 파일만 검색한다. -m : 매뉴얼 파일만 검색한다. -s : 소스 파일만 검색한다.
사용 예	whereis ls

- 예: mv 명령의 위치 검색

```
[user1@localhost ch2]$ whereis mv
mv: /bin/mv /usr/bin/mv /usr/share/man/man1p/mv.1p.gz /usr/share/man/man1/mv.1.gz
[user1@localhost ch2]$
```

03 파일 다루기

명령의 위치 찾기 : which

- 에일리어스나 PATH 환경 변수로 지정된 경로에서 파일을 검색

which

기능 명령어 파일의 위치를 찾아서 그 경로나 에일리어스를 출력한다.

형식 which 명령

사용 예 which ls

- 예: mv 명령의 위치 검색

```
[user1@localhost ch2]$ which mv
/usr/bin/mv
[user1@localhost ch2]$
```