



PC 실습 @ IT 학부

- UNIX 파일 시스템과 셸 -



학습 목표

- UNIX 파일 시스템을 이해한다.
- 파일의 종류와 특징을 이해하고 올바른 사용 방법을 익힌다.
- 디렉토리 관련 명령의 사용 방법을 익힌다.
- 파일 관련 기본 명령의 사용 방법을 익힌다.
- 파일의 내용 보기, 복사, 삭제, 이동, 링크 방법을 익힌다.
- 쉘의 기능과 종류를 이해한다.

UNIX 파일 시스템

- 파일(file)
 - 정의: "관련 있는 정보들의 집합"
 - 예제
 - 'HelloWorld.c': "C 프로그래밍 수업에서 처음 배웠던 'Hello, World!'를 출력하는 C 프로그램 명령문의 집합"
 - 'PC-report1.hwp': "PC 실습 수업에서 처음 나온 PC 부품 조립 보고서를 아래아한글 워드프로세서로 작성한 내용의 집합"
- 디렉토리(directory)
 - 정의: "파일은 파일이되, 그 내용이 다른 파일이나 다른 디렉토리에 관한 정보인 (특수) 파일"
- 파일 시스템
 - 정의: "효과적으로 관리하기 위해 파일이나 디렉토리들을 계층적으로 구성해 놓은 집합"

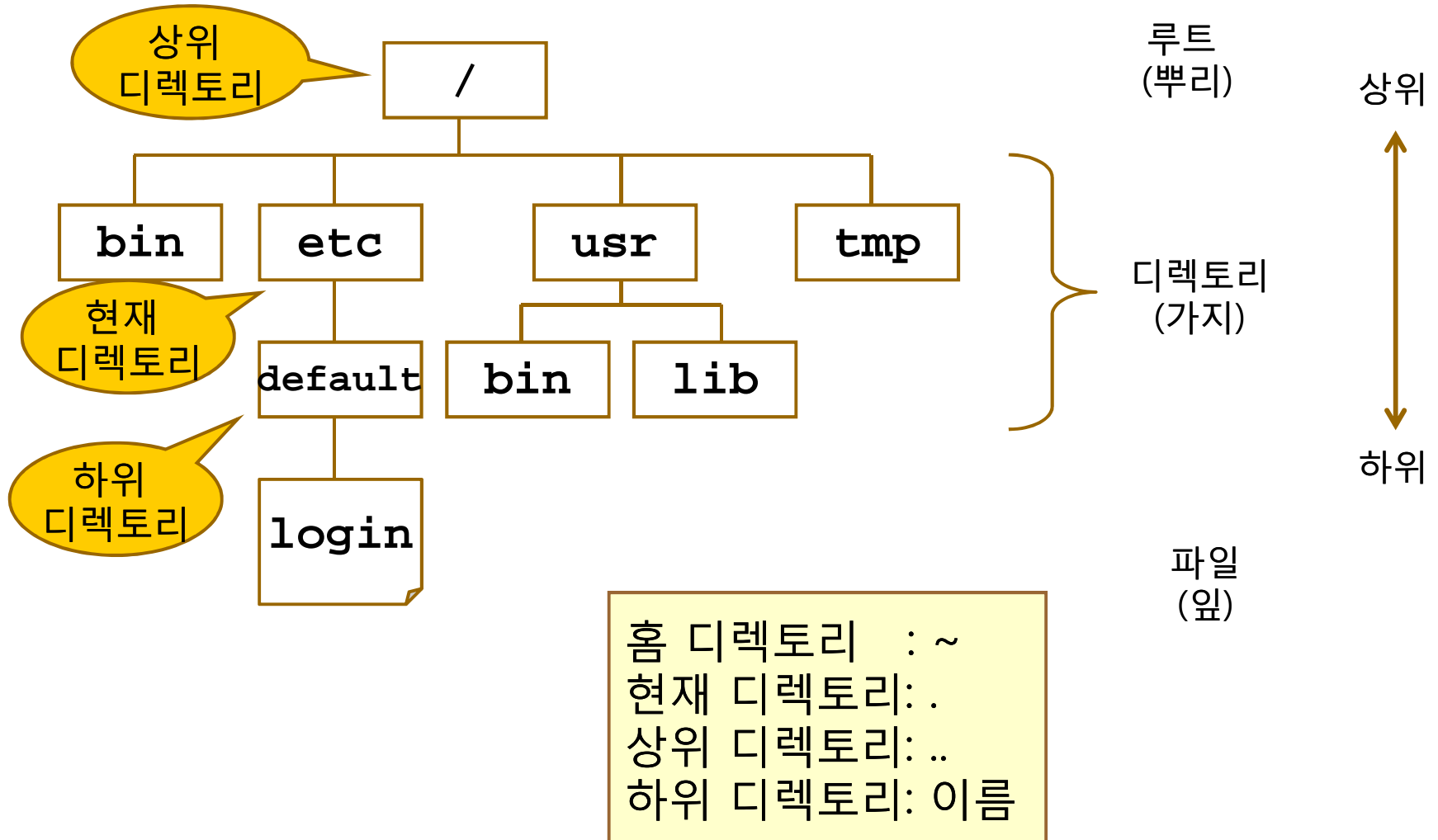
파일의 종류 (1/2)

- 종류: (일반) 파일, 디렉토리 (파일), 심볼릭 링크 (파일), 장치 파일
- (일반) 파일
 - 데이터의 저장을 목적으로 하는 파일
 - 텍스트 파일
 - 파일의 내용이 아스키(ASCII) 코드로 구성된 파일
 - 텍스트 편집기나 `cat`, `more` 명령을 이용하여 그 내용을 볼 수 있다.
 - 바이너리 파일 또는 이진 파일
 - 내용이 아스키 코드가 아닌 모든 파일
 - 지정된 응용 프로그램을 이용하여야 내용을 볼 수 있다.
- 디렉토리 (파일)
 - 내용이 다른 파일이나 디렉토리에 관한 정보인 특수 파일

파일의 종류 (2/2)

- 심볼릭 링크 (파일)
 - 원본 파일을 가리키는 역할만 하는 특수 파일
 - Windows 운영체제에서의 '바로가기' 아이콘과 유사한 파일 생성 가능
- 장치 파일
 - UNIX 시스템에 부착된 장치를 관리하기 위한 특수 파일
 - UNIX는 하드디스크, OD 드라이브, 프린터 등 시스템에 부착된 대부분의 장치를 파일로 관리
 - 각종 장치를 관리/사용하기 위해 각각에 해당하는 장치 파일을 통해 접근
 - 장치 파일들은 대개 /dev 또는 /devices 디렉토리 아래에 있음
 - 하드웨어적 성격이 강한 주변장치들과 소프트웨어적 성격이 강한 파일들을 구별하지 않고 통합적인 시각에서 공히 파일로 보는 것은 UNIX 운영체제의 독창적인 특징 중 하나였으나, 오늘날의 운영체제에서는 보편적 기능이 됨

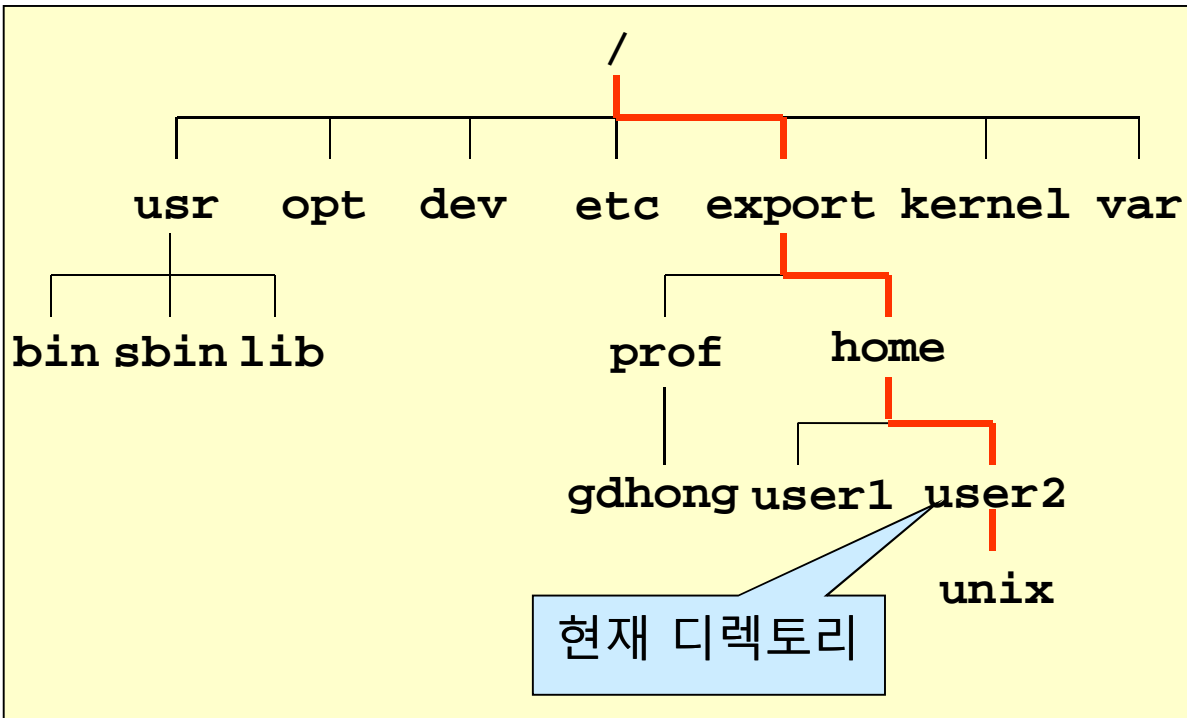
디렉토리 계층 구조



절대 경로와 상대 경로 (1/2)

- 경로
 - 파일 시스템에서 특정 파일의 위치
- 절대 경로
 - 루트 디렉토리를 기준으로 함
 - 루트 디렉토리부터 특정 파일까지 가는데 거치는 모든 디렉토리의 이름 표시
 - 항상 / 로 시작
- 상대 경로
 - 현재 위치를 기준으로 함
 - 하위로 내려갈 때는 디렉토리의 이름을, 상위로 올라갈 때는 .. 추가
 - 슬래시 이외의 문자로 시작
 - 같은 파일의 상대 경로라도 현재 위치에 따라 달라짐

절대 경로와 상대 경로 (2/2)



unix의 절대경로:
/export/home/user2
/unix

unix의 상대경로:
unix

파일과 디렉토리 명명 규칙

- 유의 사항
 - 사용 가능
 - 알파벳(대소문자 구분), 숫자, 하이픈(-), 밑줄(_), 점(.)
 - 사용 자제
 - 공백, *, &, |, ", ', ` , ~, #, \$, (,), |, ;, <, >
 - 쓰는 경우 이름을 따옴표로 감싸거나 모든 특수문자 앞에 ₩를 추가
 - 사용 불가 : /
- 예제
 - 좋은 이름
 - C, helloWorld.c, unix, .secret, sample12
 - 나쁜 이름
 - *hl, I'am, #77dir, my dir, book₩
 - 쓸 수 없는 이름
 - Mydir/, /test, wrong/name

디렉토리 관련 명령들 (1/6)

- 현재 디렉토리 확인

```
pwd
```

- 현재 작업 디렉토리의 절대 경로 출력
- 사용법

```
telnet ce.sungshin.ac.kr
```

```
$ pwd  
/under/2009/s0000001  
$
```

디렉토리 관련 명령들 (2/6)

- 디렉토리 이동

cd [디렉토리명]

- 현재 작업 디렉토리를 지정한 디렉토리로 이동
- 디렉토리명을 지정하지 않으면 자신의 홈 디렉토리로 이동
- 사용법

```
telnet ce.sungshin.ac.kr
$ cd /tmp
$ pwd
/tmp
$ cd
$ pwd
/under/2009/s0000001
$
```

/tmp 로
이동

홈 디렉토리로
이동

디렉토리 관련 명령들 (3/6)

- 디렉토리 파일 목록 확인 (1/2)

```
ls [옵션] [파일 또는 디렉토리명]
```

- 옵션

옵션	기능
a	All: 시스템에 숨겨진 파일(디렉토리)들을 모두 보여줌
l	Long: 파일과 디렉토리에 대한 상세 정보를 보여줌
d	Directory: 디렉토리에 대한 정보만 출력
R	Recursive: 하위 디렉토리 내의 모든 디렉토리들을 반복 출력
F	File: 파일 종류를 구분. (실행 파일은 *, 디렉토리는 /를 붙임)
u	Used: 파일, 디렉토리 이름을 최종 접근 시간 순으로 화면에 출력

디렉토리 관련 명령들 (4/6)

- 디렉토리 파일 목록 확인 (2/2)
 - 사용법

현재 디렉토리
의 모든 파일목
록

```
telnet ce angshin.ac.kr
$ ls -a
.  ..  .profile
$ ls .profile
.profile
$
```

지정한 파일
정보

- 실습

- 1) `ls`
- 2) `ls -a`
- 3) `ls -l`
- 4) `ls -al`
- 5) `ls -d`
- 6) `ls -F`
- 7) `ls -u`
- 8) `ls /tmp`
- 9) `ls /etc`
- 10) `ls /etc/passwd`
- 11) `ls -l /etc/passwd`

디렉토리 관련 명령들 (5/6)

- 디렉토리 생성

```
mkdir [옵션] 디렉토리명
```

- 새로운 디렉토리 생성
- 옵션
 - -p: 디렉토리 생성에 필요한 하위 디렉토리도 함께 생성

- 사용법

실습

```
telnet ce.sungshin.ac.kr
```

```
$ mkdir unix
```

```
$ ls -F
```

```
unix/
```

```
$
```

```
1)cd
```

```
2)mkdir temp dir test
```

```
3)mkdir tmp
```

```
4)cd tmp
```

```
5)mkdir temp
```

```
6)pwd
```

디렉토리 관련 명령들 (6/6)

- 디렉토리 삭제

`rmdir [옵션] 디렉토리명`

- 지정한 디렉토리 삭제
- 디렉토리가 비어 있어야 삭제 가능
- 옵션
 - `-p`: 지정한 디렉토리 삭제 후 부모 디렉토리가 비어있으면 삭제
- 사용법

실습

```
telnet ce.sungshin.ac.kr
```

```
$ rmdir dir
```

```
$
```

1) `cd`

2) `rmdir temp`

3) `cd tmp`

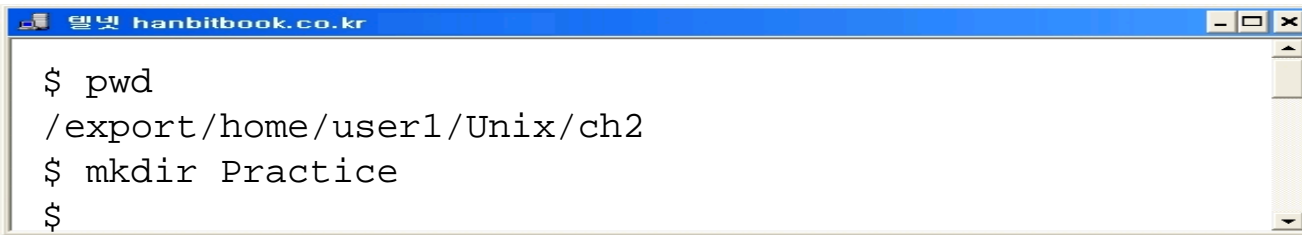
4) `ls`

5) `rmdir temp`

6) `ls`

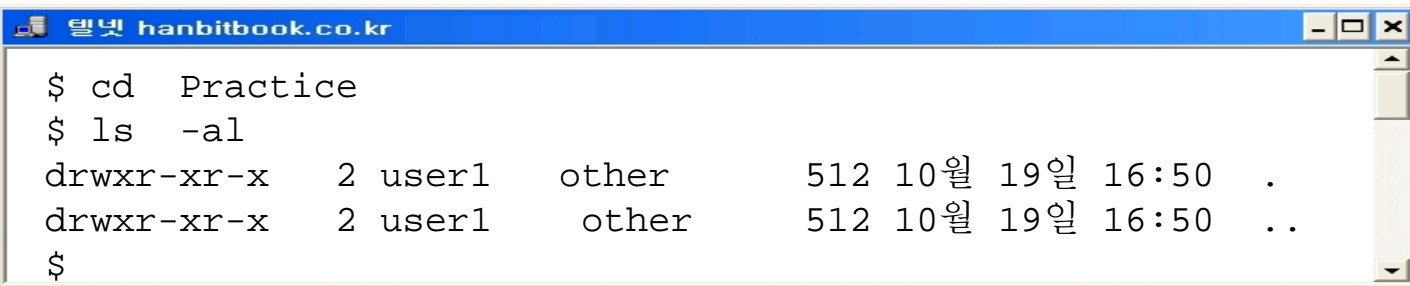
실습하기 (1/2)

① 실습용 디렉토리 만들기



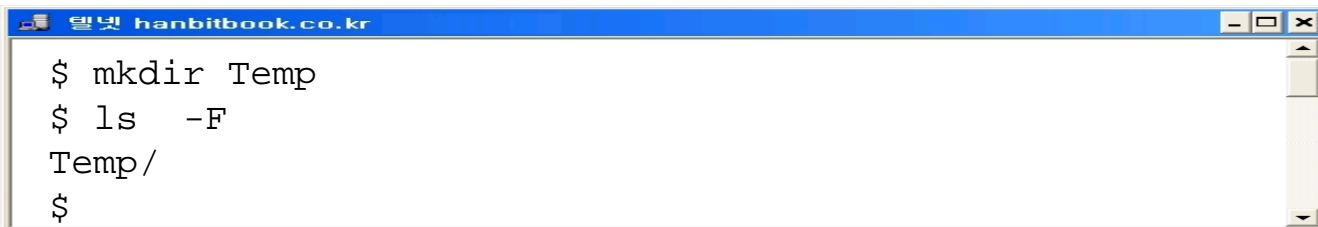
```
hanbitbook.co.kr
$ pwd
/export/home/user1/Unix/ch2
$ mkdir Practice
$
```

② 현재 위치 이동하기



```
hanbitbook.co.kr
$ cd Practice
$ ls -al
drwxr-xr-x  2 user1  other    512 10월 19일 16:50 .
drwxr-xr-x  2 user1  other    512 10월 19일 16:50 ..
$
```

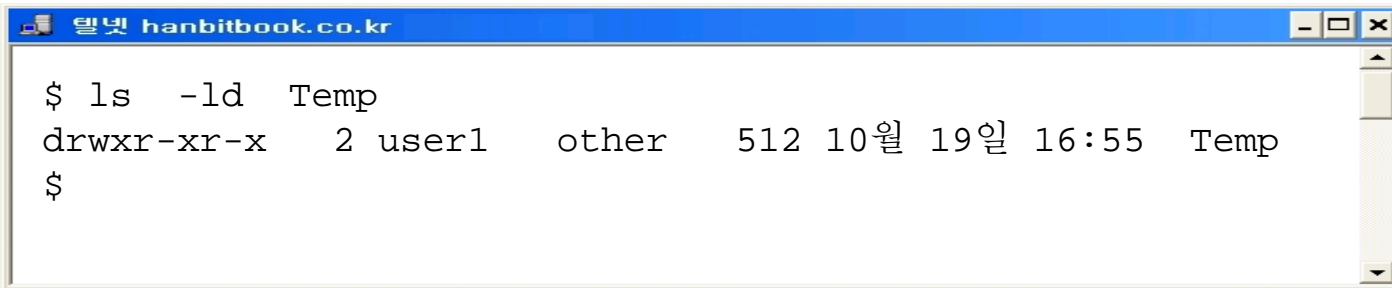
③ 실습용 디렉토리 만들기



```
hanbitbook.co.kr
$ mkdir Temp
$ ls -F
Temp/
$
```

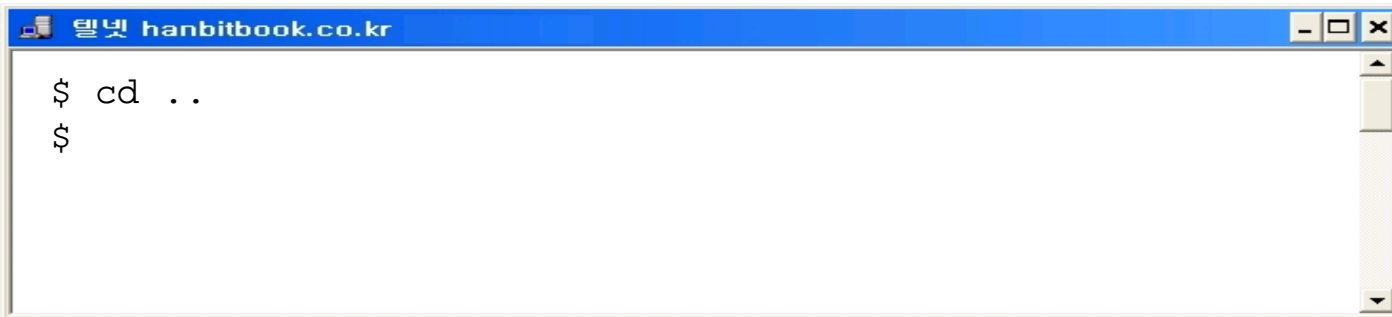
실습하기 (2/2)

④ 디렉토리 상세 정보 보기



```
말뚝 hanbitbook.co.kr  
$ ls -ld Temp  
drwxr-xr-x  2 user1  other  512 10월 19일 16:55 Temp  
$
```

⑤ 디렉토리 위치 변경하기



```
말뚝 hanbitbook.co.kr  
$ cd ..  
$
```

파일 관련 명령들 (1/8)

- 파일 내용 보기

```
cat [옵션] 파일명
```

- 간단한 파일 보기 명령으로 짧은 파일에 적합
- 옵션
 - -n: 행 번호 출력
- 사용법

```
telnet ce.sungshin.ac.kr
```

```
$ cat /etc/hosts
#
# Internet host table
#
127.0.0.1      localhost
192.168.10.118 loghost
$
```

파일 관련 명령들 (2/8)

- 파일 내용 보기

```
more [옵션] 파일명
```

- 파일의 내용을 한 화면씩 출력
- 옵션
 - +행 번호: 지정한 행부터 출력
- 명령
 - 다음 페이지: space
 - 이전 페이지: b
 - 종료: q
 - 문자열 찾기: /문자열
- 사용법

```
telnet ce.sungshin.ac.kr
```

```
$ more /etc/services
```

파일 관련 명령들 (3/8)

- 파일 내용 보기

```
tail [옵션] 파일명
```

- 파일의 마지막 부분 출력
- 옵션
 - +행 번호: 지정한 행부터 끝까지 출력
 - -숫자: 화면에 출력할 행 수
- 사용법

```
telnet ce.sungshin.ac.kr
```

```
$ tail /etc/services
```

파일 관련 명령들 (4/8)

- 파일 복사

```
cp [옵션] 원본파일명 복사파일명
```

- 특정 파일이나 디렉토리를 다른 이름으로 복사
- 옵션
 - -i: 파일을 덮어쓸 때 경고 메시지 출력
 - -r: 디렉토리 복사
- 사용 형태

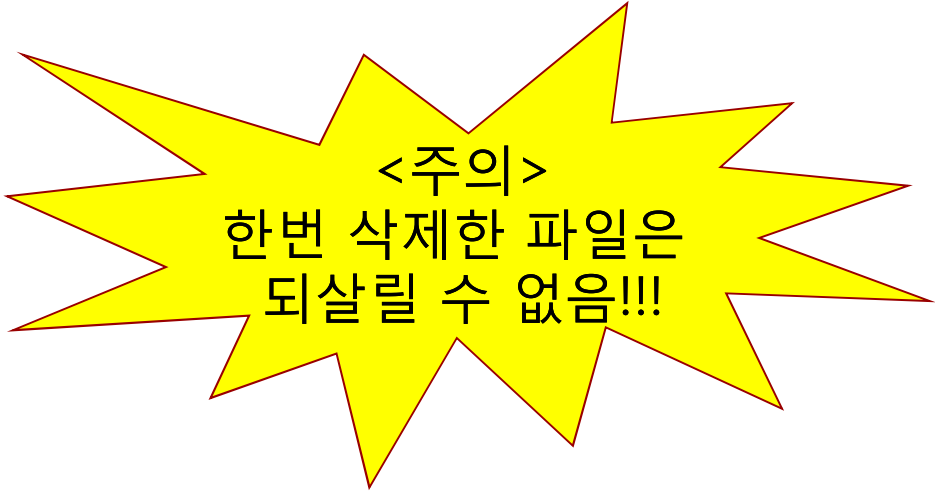
사용 형식	동 작
cp 파일1 파일2	파일1을 파일2로 복사
cp 파일들 디렉토리	파일들을 디렉토리 밑에 같은 이름으로 복사
cp -r 디렉토리1 디렉토리2	디렉토리1을 디렉토리2로 복사; 디렉토리1의 파일도 모두 복사됨.

파일 관련 명령들 (5/8)

- 파일 삭제

```
rm [옵션] 파일명/디렉토리명
```

- 지정한 파일이나 디렉토리를 삭제
- 옵션
 - -i: 확인 후 삭제
 - -f: 무조건 삭제
 - -r: 디렉토리 삭제



<주의>
한번 삭제한 파일은
되살릴 수 없음!!!

파일 관련 명령들 (6/8)

- 파일 이동

```
mv [옵션] 원래이름 새이름
```

- 파일 이름을 바꾸거나 이동
- 옵션
 - -i: 확인 후 이동
- 사용 형식

사용 형식	동 작
<code>mv</code> 파일1 파일2	파일1의 이름을 파일2로 바꿈
<code>mv</code> 파일명... 디렉토리	지정한 파일들을 디렉토리로 옮김
<code>mv</code> 디렉토리1 디렉토리2	디렉토리1의 이름을 디렉토리2로 바꿈

파일 관련 명령들 (7/8)

- 파일 링크 (1/2)

ln [옵션] 원래이름 새이름

- 하나의 파일에 새로운 이름을 붙임
- 경로를 다르게 지정할 수 있음
- 옵션
 - -s: 심볼릭 링크 파일 생성(윈도우의 바로가기 아이콘)
- 사용법

```
telnet ce.sungshin.ac.kr
$ ls -l ln.txt
-rw-r-r-- 1 user1 2007 50 4월10일 12:00 ln.txt
$ ln ln.txt ln.hd
$ ls -l ln.txt ln.hd
-rw-r-r-- 2 user1 2007 50 4월10일 12:00 ln.txt
-rw-r-r-- 2 user1 2007 50 4월10일 12:00 ln.hd
```

ln 명령을 실행하면 링크의 수가 증가함

파일 관련 명령들 (8/8)

- 파일 링크 (2/2)
 - 복사(cp)와 링크(ln)의 차이는?

cp	ln
<ul style="list-style-type: none">- 완전 별도 파일 생성- 둘 중 하나를 수정해도 다른 파일에 영향 없음- 같은 파일을 별도로 수정하여 작업할 때 사용	<ul style="list-style-type: none">- 이름만 다르고 내용은 동일- 둘 중 하나를 수정하면 두 파일이 같이 수정됨- 파일을 공동으로 관리해야 할 때 사용

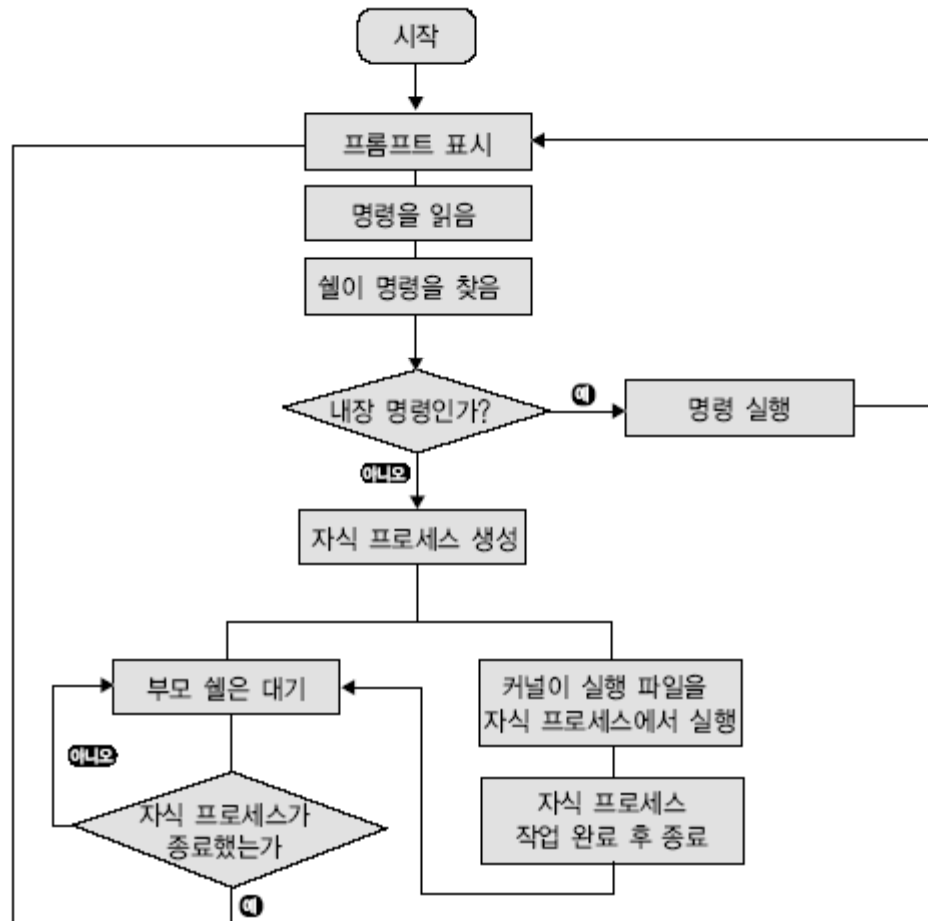
```
1) vi a.txt
2) ln a.txt b.txt
3) cp a.txt c.txt
4) ls -l
5) vi b.txt
6) vi c.txt
```

- 1)
- 2) a.txt와 b.txt의 링크수는?
(ls -l로 확인)
- 3)
- 4) a.txt, b.txt, c.txt의 링크 수는?
- 5) a.txt의 내용 확인
- 6) a.txt, b.txt, c.txt의 내용 확인

UNIX 셸의 기능과 종류

- 셸(shell)의 기능
 - 명령어 해석기(command interpreter) 기능
 - 사용자와 커널 사이에서 명령을 해석하여 전달
 - 사용자가 입력한 명령이나 파일에서 읽어 들인 명령을 해석하고 적절한 프로그램을 실행
 - 프로그래밍 기능
 - 셸은 자체 내에 프로그래밍 기능이 있어 프로그램 작성 가능
 - 셸의 프로그래밍 기능을 이용하면 여러 명령을 사용해 반복적으로 수행하는 작업을 하나의 프로그램으로 제작 가능
 - 셸 프로그램을 셸 스크립트(shell script)라고 부름
 - 사용자 환경설정 기능
 - 초기화 파일 기능을 이용해 사용자 환경을 설정
 - 명령을 찾아오는 경로 설정, 새로운 파일의 기본 권한 설정, 다양한 환경 변수 설정 등 사용자별로 사용 환경의 특성을 초기화 파일에 설정 가능
 - 로그인할 때 이 초기화 파일이 실행되어 사용자의 초기 환경이 설정됨

셸의 동작 순서



셸의 종류 (1/3)

- 본 셸(Bourne Shell, sh)
 - 최초의 본격적인 셸로 UNIX 버전 7(1978)에 처음 등장
 - AT&T Bell Laboratories의 스티븐 본(Stephen Bourne)이 개발
 - 현재도 모든 UNIX 시스템에 기본적으로 설치
 - 시스템 관리 작업을 수행하는 많은 셸 스크립트들은 본 셸을 기반으로 하고 있음
 - 히스토리 기능, 앨리어스 기능, 작업 제어 등 사용자의 편의를 위한 기능을 다양하게 제공하지 않는 것이 단점
- 콘셸(Korn Shell, ksh)
 - 1982 AT&T Bell Laboratories의 데이비드 콘(David Korn)이 개발
 - 유닉스 SVR4에 포함되어 발표되었는데, 본 셸과의 호환성을 유지
 - 히스토리 기능, 앨리어스 기능 등 C 셸의 특징들도 모두 제공하면서 처리 속도도 빠르다는 장점을 가지고 있음

셸의 종류 (2/3)

- C 셸(C Shell, csh)
 - University of California at Berkeley에서 빌 조이(Bill Joy)가 개발
 - 2BSD UNIX(1978)에 포함되어 발표
 - 본 셸의 기능을 확장한 것으로 앨리어스나 히스토리 같은 사용자 편의 기능을 포함
 - 셸 스크립트 작성을 위한 구문 형식이 C 언어와 같아 C 셸이라 불림
 - 크기가 커지고 처리 속도도 느려졌다는 단점이 있으나 편리한 기능 때문에 일반 사용자들이 즐겨 사용
- 배시 셸 (Bourne Again Shell, bash)
 - 본 셸을 기반으로 1987년 Free Software Foundation의 브라이언 폭스 (Brian Fox)에 의해 개발
 - 본 셸과 호환성을 유지하면서 C 셸, 콘 셸의 편리한 기능도 포함
 - GNU에 따르는 공개 소프트웨어로, 리눅스의 기본 셸로 동작하고 있어 리눅스 셸로도 많이 알려져 있음

셸의 종류 (3/3)

- 기타 셸
 - TC 셸(tcsh)
 - C 셸의 기능 확장
 - 명령행 편집 기능, 명령어 완성 기능, 히스토리 목록에서의 시간 표시 등
 - 데스크탑 콘 셸(dtksh)
 - X 윈도우의 기능을 지원할 수 있도록 콘 셸의 기능을 확장
 - Z 셸(zsh)
 - TC 셸과 콘 셸의 기능을 함께 가지고 있는 셸

셸의 변경 (1/2)

- 사용자의 기본 셸 확인
 - 사용자 등록시 시스템 관리자가 지정
 - /etc/passwd에 저장

```
telnet ce.sungshin.ac.kr
```

```
$ cat /etc/passwd
```

```
root:x:0:1:Super-User:/:/bin/sh
```

```
sys:x:3:3:/:
```

```
adm:x:4:4:Admin:/var/adm:
```

```
...
```

```
user1:x:40577:1::/export/home/user1:/bin/ksh
```

```
user2:x:40578:1::/export/home/user2:/bin/csh
```

셸의 변경 (2/2)

- 사용자의 기본 셸 변경은 시스템관리자만 가능
- 일반 사용자는 현재 실행중인 셸을 임시로 바꿀 수 있음
 - 해당 셸을 실행시킴

```
telnet ce.sungshin.ac.kr
$ ps
  PID TTY          TIME CMD
 28868 pts/1    0:00 ksh
$
```

현재 셸 확인

```
telnet ce.sungshin.ac.kr
$ csh
%
```

C 셸로 변경
(프롬프트 : %)