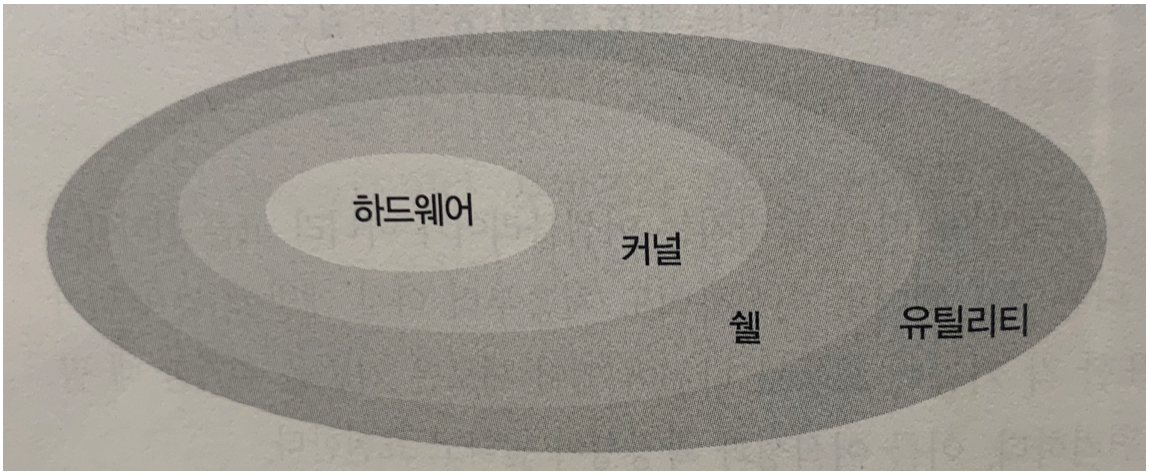
Unix 2학기 중간고사 정리

1장 개요 및 기본 사용법 P. 22

유닉스의 구조 P. 26



- 커널 : 유닉스 운영체제의 핵심으로 프로세스 관리, 메모리 관리, 파일 시스템 관리, 장치 관리 등 컴퓨터의 모든 자원을 초기화 하고 제어하는 기능을 수행한다.
- 쉘 : 사용자와 커널 사이의 중간자 역할을 담당하는 특별한 프로그램이다. 쉘은 사용자가 입력한 명령을 해석하여 커널에 넘겨준다. 그러면 커널이 명령의 수행 결과를 돌려주고 쉘은 다시 사용자가 이해할 수 있는 형태로 바꾸어 출력한다. 유닉스에서 쉘은 한 가지만 있는 것이 아니다.
- 유틸리티 : 유닉스는 각종 개발 도구, 문서 편집 도구, 네트워크 관련 도구 등 매우 다양한 유틸리티를 제공한다.

기초 명령 사용법 P. 33

- `banner [문자열]` : 인자로 입력한문자를 큰 글씨로 출력해준다. 문자열을 10자로 제한되며, 문자열 사이에 공백이 있을 경우 `""`로 묶어준다.
- `date` : 현재 날짜와 시각을 출력한다.
- `clear` : 화면을 지운다.
- `man [명령]` : 인자로 입력한 명령에 대한 사용법을 자세히 보여준다.
- `passwd [인자]` : 인자로 로그인ID를 넘겨주면 해당 계정의 비밀번호를 바꿀 수 있다.
- `logout` : 로그인 쉘에서 사용 가능하며, 유닉스 시스템의 접속을 해제한다.
- `exit` : 유닉스 시스템의 접속을 해제한다.

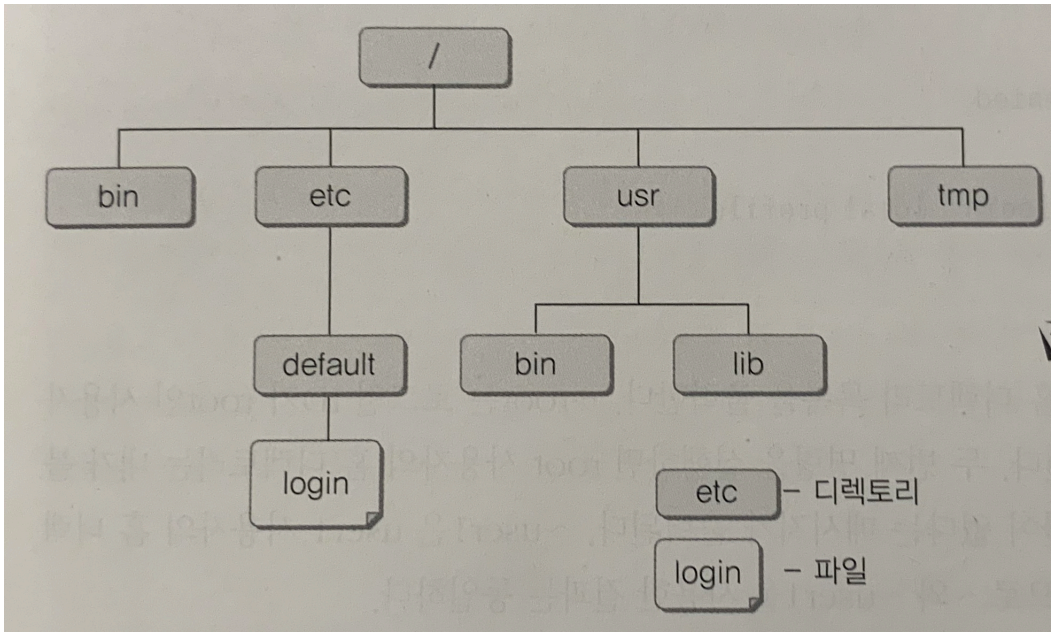
2장 디렉토리 다루기 P. 42

파일의 종류 P. 42

- 일반 파일 : 데이터 저장이 목적이며 편집기를 사용해 생성한 파일, 각종 응용 프로그램을 사용해 생성한 파일 등이 이에 해당한다.

- 디렉토리 파일 : 내용이 다른 파일이나 하위 디렉토리의 이름인 특수 파일이다.
- 심볼릭 링크 파일 : 윈도우 시스템의 바로가기에 해당하는 파일로, 원본 파일을 가리키는 역할만 하는 특수 파일이다.
- 장치 파일 : 유닉스 시스템에 부착된 장치들을 관리하기 위한 특수 파일이다. 유닉스는 하드 디스크, CD 드라이브, 프린터 등 시스템에 부착된 대부분의 장치를 파일로 관리하기 때문에 시스템 관리자는 각종 장치를 관리하기 위해 해당 장치 파일에 접근하게 된다. 이와 같은 장치 파일들은 /dev, /devices 디렉토리 아래에 있는데, 크게 블록 장치 특수 파일과 문자 장치 특수 파일이 있다.

디렉토리 계층 구조 P. 45



파일 시스템을 계층적인 디렉토리 트리 구조로 구성한다. 디렉토리 계층 구조에서 모든 디렉토리, 파일의 원조 및 시작은 루트 디렉토리이다. 루트 디렉토리는 `/` 기호로 표시한다.

default 디렉토리는 etc 디렉토리의 아래에 있다. 이 때 default 디렉토리는 etc 디렉토리의 하위 디렉토리 또는 서브 디렉토리라고 한다. 반대로 etc 디렉토리는 default 디렉토리의 상위 디렉토리 또는 부모 디렉토리라고 부른다. 상위 디렉토리는 기호 `..`으로 표시한다. 모든 디렉토리의 최상위 디렉토리는 루트 디렉토리 `/`이다. 따라서 **루트 디렉토리는 유닉스 시스템에서 상위 디렉토리가 없는 유일한 디렉토리이다.**

- 루트 디렉토리 : `/` 기호로 표시하며 유닉스 시스템에서 상위 디렉토리가 없는 유일한 디렉토리이다.
- 현재 디렉토리 : `.` 기호로 표시하며 현재 작업중인 디렉토리를 나타낸다.
- 홈 디렉토리 : `~` 기호로 표시하며 홈 디렉토리를 나타낸다.

현재 디렉토리 확인 P. 49

`pwd` 명령어를 사용하며, 현재 위치(디렉토리의 절대경로)를 출력한다.

디렉토리 이동 P. 50

`cd [디렉토리명]` 명령어를 사용하며, 디렉토리명으로 입력된 디렉토리로 이동하게 된다.

파일 목록 확인 P. 52

`ls [옵션] [디렉토리명]` 명령어를 사용하며, 디렉토리 내의 파일 목록을 확인 할 때 사용된다.

옵션	옵션 내용
-F	파일 종류를 함께 표시한다.
-l	파일들의 상세 정보를 표시한다.
-d	디렉토리 자체의 정보를 확인할 수 있다.

디렉토리 생성 P. 56

`mkdir [옵션] [디렉토리명]` 명령어를 사용하며 디렉토리명을 입력하면 디렉토리가 생성된다.

옵션	옵션 내용
-p	하위 디렉토리를 계층적으로 생성할 때 중간의 디렉토리가 없으면, 생성하면서 전체 디렉토리를 생성

디렉토리 한개 생성

`mkdir [디렉토리명]` 명령어를 입력하면 디렉토리명으로 디렉토리가 한개 생성된다.

디렉토리 여러개를 동시에 생성

`mkdir [디렉토리명] [디렉토리명] ...` 명령어를 입력하면 디렉토리명으로 디렉토리가 여러개 생성된다.

중간 디렉토리 자동 생성

`mkdir -p [디렉토리]` 명령어를 입력하면 디렉토리 내에 포함된 하위 디렉토리까지 자동으로 생성된다.

디렉토리 삭제 P. 61

`rmdir [옵션] [디렉토리명]` 명령어를 사용하며 비어있는 디렉토리를 삭제할 수 있다.

옵션	옵션 내용
-p	지정한 디렉토리를 삭제한 뒤, 그 디렉토리의 부모 디렉토리가 빈 디렉토리일 경우 부모 디렉토리도 자동으로 삭제

3장 파일 다루기 P. 66

파일 내용 보기

연속출력

`cat [옵션] [파일명]` 명령어를 사용하며, 파일의 내용을 화면에 연속적으로 출력한다.

옵션	옵션 내용
-n	행 번호를 붙여서 출력

화면 단위로 출력

`more [옵션] [파일명]` 명령어를 사용하며, 파일의 내용을 화면 단위로 출력한다.

옵션	옵션 내용
+행번호	출력을 시작할 행 번호를 지정

파일의 뒷부분 출력

`tail [옵션] 파일명` 명령어를 사용하며, 파일의 뒷부분 몇 행을 출력한다. 기본값은 10으로 파일의 뒷부분 10행이 출력된다.

옵션	옵션 내용
+행번호	지정한 행부터 끝까지 출력
-숫자	화면에 출력할 행의 수를 지정(기본 값은 10)
-f	파일 출력이 종료되지 않고, 주기적으로 계속 출력

파일 복사 P. 70

`cp [옵션] [파일/디렉토리] [파일/디렉토리]` 명령어를 사용하며, [파일/디렉토리]에서 제일 뒤의 [파일/디렉토리]로 복사한다.

옵션	옵션 내용
-i	제일 뒤의 [파일/디렉토리]로 복사할 때 덮어 쓸 것인지 물어본다.
-r	디렉토리를 복사할 때 사용된다.

파일 삭제 P. 78

`rm [옵션] [파일/디렉토리]` 명령어를 사용하며, [파일]을 삭제할 때 사용한다.

옵션	옵션 내용
-i	지정한 파일을 정말로 삭제할 것인지 물어본다.
-r	디렉토리를 삭제할 때 사용된다.

파일 이동 P. 81

`mv [옵션] [파일/디렉토리명] [파일/디렉토리명]` 명령어를 사용하며, [파일/디렉토리]를 제일 뒤의 [파일/디렉토리]로 옮긴다.

옵션	옵션 내용
-i	제일 뒤의 [파일/디렉토리]가 존재하면 덮어쓸 것인지 물어본다.

파일 링크 P. 86

`ln [옵션] [원본파일] [링크파일]` 명령어를 사용하며, [원본파일]에 대한 링크 파일이 [링크파일]로 생성된다.

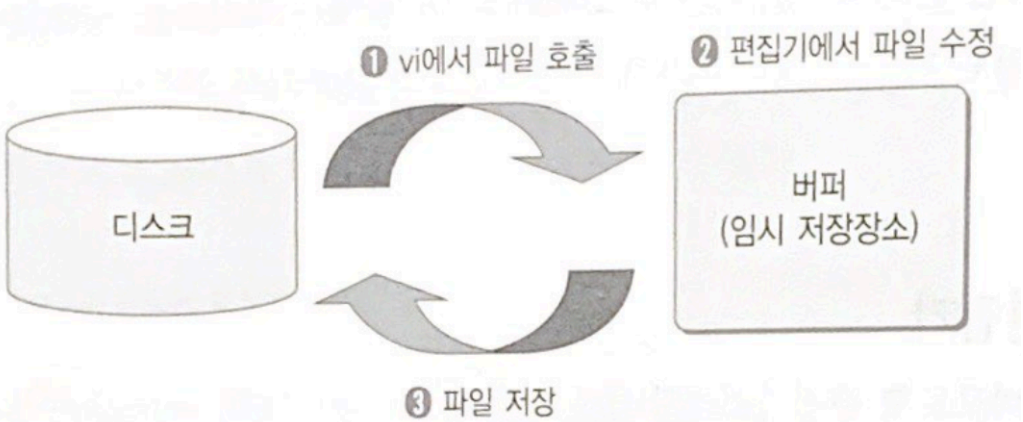
옵션	옵션 내용
-s	심볼릭 링크 파일 생성

파일 생성 P. 91

`touch [-acm] [-r ref_file | -t time] [파일]` 명령어를 사용하며, 새로운 파일을 생성하거나 파일 관련 시각을 변경하는 명령이다.

옵션	옵션 내용
-a	접근 시각만 변경
-c	지정된 파일이 없는 경우 새로 생성하지 않음
-m	수정 시각만 변경
-r [ref_file]	시각을 파일에서 읽음
-t [time]	시각을 직접 입력

유닉스 편집기 P. 100



[그림 4-1] vi의 동작 구조

vi 에서 편집할때에는 디스크에서 파일을 읽어들여 버퍼에서 작업한다. 파일을 저장할 때 버퍼의 내용이 디스크로 작성된다.

vi의 동작 모드

- 명령모드 : 입력한 내용을 명령어로 해석
- 입력모드 : 입력한 내용이 버퍼로 옮겨져 추가, 수정
- 마지막 행 모드 : 검색, 저장, 바꾸기, 행 이동

vi 에디터 열기

`vi [파일명]` 명령어를 사용하며, [파일명]을 입력하면 파일을 vi에디터로 열고 vi만 입력하게 되면 빈 문서가 vi에디터로 열린다.

vi 종료하고 파일 저장하기

명령키	기능
:q	vi에서 작업한 것이 없을 때 그냥 종료한다.
:q!	작업한 내용을 저장하지 않고 종료한다.
:w [파일명]	작업한 내용을 저장만 한다. 파일명을 지정하면 새 파일로 저장한다.
:wq	작업한 내용을 저장하고 vi를 종료한다.
:wq!	작업한 내용을 저장하고 vi를 종료한다.
ZZ	작업한 내용을 저장하고 vi를 종료한다.

입력 모드로의 전환

명령키	기능
i	커서 앞에 입력한다. (현재 커서 자리에 입력)
a	커서 뒤에 입력한다. (현재 커서 다음 자리에 입력)
o	커서가 위치한 행의 마지막 컬럼으로 이동해 입력한다.
I	커서가 위치한 행의 첫 컬럼으로 이동해 입력한다.
A	커서가 위치한 행의 마지막 컬럼으로 이동해 입력한다.
O	커서가 위치한 행의 이전 행에 입력한다.

커서 이동

명령키	기능
h	커서를 한 문자 왼쪽으로 이동시킨다
j	커서를 한 문자 아래로 이동시킨다.
k	커서를 한 문자 위로 이동시킨다.
l	커서를 한 문자 오른쪽으로 이동시킨다.
^ 또는 0	커서를 현재 행의 처음으로 이동시킨다.
$	커서를 현재 행의 마지막으로 이동시킨다.
-	커서를 이전 행의 처음으로 이동시킨다.
+ 또는 Enter키	커서를 다음 행의 처음으로 이동시킨다.
H	커서를 화면의 맨 위 행으로 이동시킨다.
M	커서를 화면의 중간 행으로 이동시킨다.
L	커서를 화면의 맨 아래 행으로 이동시킨다.
w	커서를 다음 단어의 첫 글자 위치로 이동시킨다.
b	커서를 앞 단어의 첫 글자 위치로 이동시킨다.
e	커서를 다음 단어의 마지막 글자 위치로 이동시킨다.

화면 이동

명령키	기능
^u	반 화면 위로 이동시킨다.
^d	반 화면 아래로 이동시킨다.
^b	한 화면 위로 이동시킨다.
^f	한 화면 아래로 이동시킨다.
^y	화면을 한 행만 위로 이동시킨다.
^e	화면을 한 행만 아래로 이동시킨다.
G	마지막 행으로 이동
행번호G	지정한 행으로 이동
:행번호	지정한 행으로 이동
:$	마지막 행으로 이동

수정 및 삭제

명령키	기능
r	커서가 위치한 문자를 다른 문자로 수정한다.
cw, #cw	커서 위치부터 현재 단어 끝까지 수정한다. (#에는 수정할 단어의 수를 지정, 예를 들어 3cw는 커서 위치부터 3단어 수정)
s, #s	커서의 위치부터 ESC 키를 입력할 때까지 수정한다.
cc	커서가 위치한 행의 내용을 모두 수정한다.
C	커서의 위치부터 행의 끝까지 수정한다.
x, #x	커서가 위치한 문자를 삭제한다 (#은 삭제할 문자 수, 예를 들어 3x 하면 세 글자 삭제)
dw, #dw	커서 위치의 단어를 삭제한다. (#은 삭제할 단어 수)
dd, #dd	커서가 위치한 행을 삭제한다. (#은 삭제할 행의 수, 예를 들어 5dd는 커서 위치부터 다섯 행을 삭제)
u	방금 수행한 명령을 취소한다.
U	해당 행에서 한 모든 명령을 취소한다.
:e!	마지막으로 저장한 내용 이후의 것을 버리고 새로 작업한다.

편집 기능

명령키	기능
yy, #yy	커서가 위치한 행을 복사한다. (#에는 복사할 행 수 지정, 예를 들어 3yy하면 세 행 복사)
p	커서가 위치한 행의 아래쪽에 붙인다.
P	커서가 위치한 행의 위쪽에 붙인다.
dd, #dd	커서가 위치한 행을 잘라둔다. (#에는 잘라둘 행의 수 지정, 예를 들어 3dd 명령을 입력하면 세 행 잘라두기)
:#y	#으로 지정한 행을 복사한다. (10y → 10행 복사)
:[범위]y	범위로 지정한 행을 복사한다. (10,20y → 10~20행 복사)
:#d	#으로 지정한 행을 삭제한다.
:[범위]d	범위로 지정한 행을 삭제한다.
:pu	현재 행 다음에 버퍼의 내용을 붙인다.
:#pu	#으로 지정한 행 다음에 버퍼의 내용을 붙인다.

vi 환경설정 P. 131

옵션	기능
:set nu	파일 내용의 각 행에 행 번호를 표시한다.
:set nonu	행 번호를 감춘다
:set list	눈에 보이지 않는 특수 문자를 표시한다.
:set nolist	특수 문자를 감춘다.
:set showmode	현재 모드를 표시한다.
:set noshowmode	모드 표시를 감춘다.
:set	set으로 설정한 모든 vi 변수를 출력한다.
:set all	모든 vi 변수와 현재 값을 출력한다.

5장 배시 쉘 활용하기 P. 131

문자열 출력

`echo [문자열]` : [문자열]을 출력한다. 쉘 내장 명령을 실행한다.

`/usr/bin/echo [문자열]` : [문자열]을 출력한다. 유닉스 유틸리티를 실행한다.

`print "문자열"` : "문자열"을 출력한다. 쉘 내장 명령을 실행한다.

특수문자

`*` : 여러 파일의 이름을 나열할 때 파일 이름을 간단히 표시하기 위해 사용되며, 파일 이름을 적어야 하는 자리에 *을 사용하면 모든 파일을 나타낸다.

`?` 또는 `[]` : ?는 임의의 한 문자를, []는 괄호 안에 포함된 문자 중 하나를 나타낸다. []는 사용할 여러 문자를 나열하거나 범위를 지정할 수 있으며 다른 특수 문자와도 혼합해 사용할 수도 있다.

`~` 또는 `-` : ~가 단독으로 사용되면 현재 작업중인 사용자의 홈 디렉토리를 나타내고, 다른 사용자의 로그인 ID와 함께 사용되면 해당 사용자의 홈 디렉토리를 나타낸다. -는 cd 명령으로 디렉토리를 이전하기 직전의 작업 디렉토리를 나타낸다.

`;` 또는 `|` : 명령과 명령을 연결하는데 사용된다. ;은 연결된 명령을 왼쪽부터 차례로 실행하고, |는 왼쪽 명령의 실행 결과를 오른쪽 명령의 입력으로 전달한다. 파이프를 이용하면 여러 명령을 사용할 수 있다.

`''` 또는 `""` : 문자열을 감싸며 문자열 안에 사용된 특수 문자의 의미를 없애는 기능을 한다.

`` `: `` : 명령 실행 결과를 문자열로 받고 싶을때 사용하는 특수 기호로, 작은 따옴표와 혼동하기 쉬우니 주의해야 한다.

`\` : 특수 문자를 일반 문자처럼 사용하도록 한다. 특수 문자 바로 앞에 사용되며 해당 특수 문자의 효과를 없앤다. 대부분의 키보드에서 \ 대신에 ₩를 사용한다.

`>` 또는 `>>` 또는 `<` : 입출력의 방향을 바꾸는 특수 문자이다.

입출력 방향 변경 P. 144

출력 리다이렉션

`>` : 표준 출력 파일을 바꾸는 특수 문자이다. 명령은 다음과 같이 사용된다.

`[명령] [파일 디스크립터]> [파일]` : [명령]의 실행 결과를 [파일]에 그대로 작성한다. [파일]이 존재해야 사용할 수 있다.

`>>` : 명령의 실행 결과를 파일의 끝에 덧붙인다.

`[명령] >> [파일]` : [명령]의 실행결과를 [파일]의 뒤에 그대로 이어붙인다.

`<` : 입력 리다이렉션으로 표준 입력 장치 파일을 바꾸는 기능을 제공한다.

`[명령] < [파일]` : [파일]의 내용을 읽어서 [명령]에 보낸다.

배시 쉘 환경 설정 P. 144

변수 출력

환경변수	의미
HOME	사용자 홈 디렉토리의 절대 경로
LOGNAME	사용자 계정 이름
PATH	명령을 탐색할 경로
CDPATH	cd 명령 사용시 인자로 주어진 디렉토리를 찾을 경로
PWD	작업 디렉토리 절대 경로
SHELL	로그인 쉘

변수 정의

`[변수]=[문자열]` : [문자열]을 [변수]로 지정한다.

`export [-n] [변수]` : 지정한 쉘 변수를 환경 변수로 바꿔주고, 그 값을 현재 쉘과 서브 쉘에 적용시키는 쉘의 내부 명령이다.

변수 해제

`unset [변수]` : 지정한 변수를 해제한다.

배시 쉘 명령 다루기

`alias [별칭]=[명령]` : [명령]을 [별칭]으로 대체하여 사용할 수 있다.

`alias` : 현재 설정된 별칭들을 모두 볼 수 있다.

[명령]에 공백이 있다면 `''` 따옴표 기호로 묶어준다.

`unalias [별칭]` : [별칭]으로 등록된 명령을 해제한다.

이전에 입력한 명령 사용하기: 히스토리

배시 쉘은 사용자가 입력한 명령을 사용자 홈 디렉토리 아래의 특수한 파일에 저장한다. 숨김파일인 .bash_history는 명령을 저장하는 목적으로 사용되어 사용자가 실행한 명령이 저장된다.

`more ~/.bash_history` : ~/.bash_history 파일을 볼 수 있다.

명령 재실행하기

`![문자]` : [문자]로 시작하는 명령 중 최근의 명령을 다시 실행한다.

명령 편집하기

`화살표키` 또는 `ESC + K` : `↑` 화살표 기호로 이전에 사용한 명령을 순서대로 가져올 수 있다.

파일 접근 권한 관리하기 P. 190

파일 종류

`file [파일]` : 지정한 파일의 종류를 알려준다.

`groups [사용자명]` : 사용자가 속한 그룹을 알려준다.

파일 접근 권한 변경

`chmod [옵션] [디렉토리/파일]` : 파일이나 디렉토리의 접근 권한을 변경한다.

구분	문자/기호	의미
사용자 카테고리	u	파일 소유자
사용자 카테고리	g	파일에 속한 그룹
사용자 카테고리	o	소유자와 그룹 이외의 기타 사용자
사용자 카테고리	a	전체 모든 사용자
연산자	+	권한부여
연산자	-	권한 제거
연산자	=	접근 권한 설정
권한	r	읽기 권한
권한	w	쓰기 권한
권한	x	실행 권한

8진수로 접근 권한 표시하기

접근 권한	숫자 모드	접근 권한	숫자 모드
rwxrwxrwx	777	rw-r--r--	644
rwxr-x-r-x	755	rwx------	700
rw-rw-rw	666	rw-r-----	640
r-xr-xr-x	555	r--------	400

숫자 모드를 이용한 접근 권한 변경

`chmod [숫자 모드] [파일]` : [파일]의 권한을 [숫자모드]로 변경한다.

unmask 명령을 이용한 기본 접근 권한 설정

`unmask [마스크 값]` : 기본 접근 권한을 출력하거나 변경한다.

파일과 디렉토리 검색하기 P. 216

`grep [옵션] [패턴] [파일]` : [파일]에서 [패턴]을 찾고 싶을 때 사용하는 명령어이다.

egrep의 사용법

`egrep [정규표현식] [파일]` : grep과 사용법은 동일하며 정규 표현식에 사용할 수 있는 특수 문자가 더 추가된다.

파일 검색

`find [경로] [검색조건] [동작]` : 조건에 맞는 파일을 지정한 위치에서 찾는다.

명령 검색

`which [명령]` : 명령어 파일의 위치를 찾아서 그 경로나 앨리어스를 출력한다.

8장 프로세스와 사용자 명령 익히기 P. 240

프로세스 목록 보기 명령

`ps [옵션]` : 현재 실행중인 프로세스의 정보를 출력한다.

`pgrep [옵션] [패턴]` : 지정한 패턴과 일치하는 프로세스의 정보를 출력한다.

`kill [시그널] PID` : 지정한 PID를 프로세스를 종료한다.

`prstat [옵션]` : 동작중인 프로세스의 통계 정보를 출력한다.

사용자 정보 보기

`user` : 현재 로그인 하고 있는 사용자명을 출력한다.

`who` : 로그인 하고 있는 사용자 뿐만 아니라 로그인 한 단말기 번호와 로그인 시간 등을 함께 출력한다.

`w [사용자명]` : 로그인 한 사용자 정보와 현재 작업의 정보를 출력한다.

`who am i` : 로그인 정보 출력하기

`whoami` : 자신의 로그인 사용자명 출력하기

`id [옵션]` : 사용자의 로그인 ID와 그룹 정보를 출력한다.