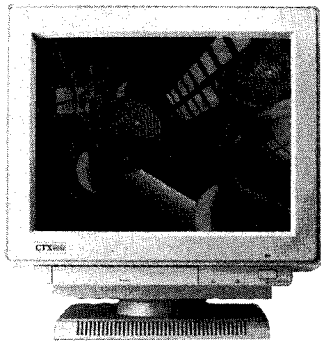


UNIX의 사용은 이렇게(2)

김차순/ITU국

협회 업무 전산화로 UNIX 시스템을 도입하게 되었고 이의 활발한 사용을 위해 UNIX 이야기를 지난호부터 소개해 드리고 있다. 자세한 내용을 상술했으면 하는 아쉬움도 있지만 초보자를 위한 가이드 정도로서 지난 호에 이은 나머지 부분에 대해 앞서 소개되었던 'UNIX 시작하기'와 '화일(FILE)과 디렉토리(DIRECTORY)'에 이어 '화일(FILE) 편집', '쉘(SHELL) 이란', '자료 관리 툴(TOOL)'에 대해서 알아보기로 하자.

화일 (FILE) 편집

UNIX에서는 화일(FILE)을 편집하는 명령어가 여러 가지가 있다. 이들 중 몇가지를 소개하고자 한다.

① cat : 짧은 화일을 작성할 때 워드프로세서나 텍스트 에디터(vi)를 사용하지 않고 간단하게 사용하고자 할 때 사용된다. DOS와 비교해 볼 경우 DOS에서는 COPY 명령으로 kkk라는 화일을 작성하는 예는 아래와 같이 한다.

```
C:\> copy con kkk ↵
     푸르른 하늘을 보자
     ^Z
     1 Files copied
C:\>
```

똑같은 일을 UNIX에서는 cat 명령을 사용한다.

```
% cat > kkk ↵
  푸르른 하늘을 보자
    ^D
%
```

화일을 저장하려면 ^D를 입력하며, 화일이 작성되면 다음과 똑같은 cat 명령을 사용하여 표시할 수 있다.

```
% cat kkk ↵
  푸르른 하늘을 보자
```

여기에서 잠깐 리디렉션을 배워보기로 하자.

DOS 또는 UNIX는 입력과 출력을 표준 입력인 키보드와 표준 출력인 화면에 나타낸다. 여기서 입, 출력 행선방향을 변경하고자 할 때 리

디렉션 기능을 사용한다.

리디렉션은 UNIX에서 도입된 기능이므로, DOS, UNIX 모두 같은 기호를 사용하고 있다.

< 입력의 리디렉션
> 출력의 리디렉션
>> 출력의 리디렉션의 추가

이 기호들은 하나의 흐름 방향을 나타내는 화살표라고 할 수 있다. 예를 들어 cat 명령으로 화일을 하나 작성하여 화일의 내용을 메일(mail)로 보내고자 할 때

% mail candy < kkk ↵

kkk라는 화일의 내용을 입력으로 받아 상대방에게 메일(mail)을 보낼 수 있다.

② vi : DOS에서는 많은 워드프로세서를 사용할 수 있으므로 EDLIN이라는 에디터는 거의 사용하지 않을 것이다. EDLIN은 DOS에 내장된 가장 기본적인 라인 에디터(Line editor)이다. 라인 에디터라는 것은 각 행에 행 번호를 붙이고, 이 번호를 이용하여 편집을 한다. UNIX에도 ed라고 불리우는 라인 에디터가 있다.

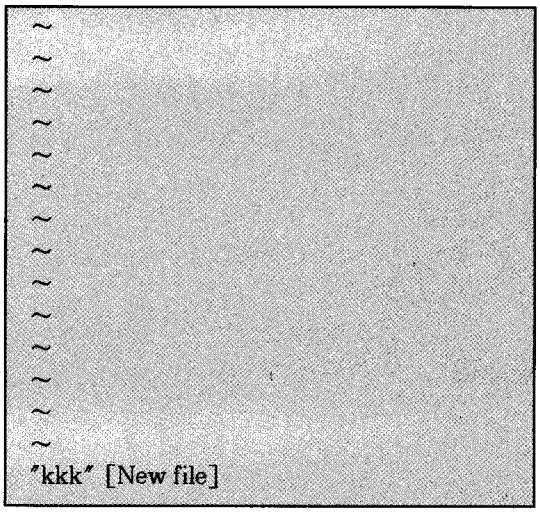
이전엔 UNIX에서 텍스트를 편집할 수 있는 유일한 프로그램이 바로 ed이었다. 그러나 1970년대에 캘리포니아 대학에서 vi(visual interpreter)라고 불리우는 프로그램이 개발되어 80년대에는 가장 많이 쓰이는 편집기로서 위치를 굳혔다.

UNIX에서는 ed, pg등 다수의 편집기가 존재하지만 vi에 관해서만 설명하기로 한다. vi는 풀 스크린 에디터이며 조작성이나 기능을 현재의 Unix에서 사용되는 모든 어플리케이션(application)의 조작의 기본이 된다.

vi에서는 편집 전이든 후이든 원하는 때에 화일명을 지정할 수 있다.

% vi kkk ↵

그러면 다음과 같은 화면이 표시된다.



이 화면에서 화일을 편집할 수 있다. 기타 vi에서 필요한 명령은 아래를 참고한다.

vi 시작	
vi filename	화일 열기 또는 작성
vi+18 filename	18번째 행으로 화일 열기
vi-r filename	손상된 화일 회복
view filename	읽기 전용 화일 열기
커서 명령	
h	왼쪽으로 이동
j	아래로 이동
k	위로 이동
l	오른쪽으로 이동
w	한 단어 오른쪽으로 이동
W	한 단어 오른쪽으로 이동 (past punctuation)
b	한 단어 왼쪽으로 이동
B	한 단어 왼쪽으로 이동 (past punctuation)
Return	한 행 아래로 이동
Back Space	한 문자 왼쪽으로 이동
Space Bar	한 문자 오른쪽으로 이동
H	화면의 맨 위로 이동
M	화면의 중간으로 이동
L	화면의 맨 아래로 이동
Ctrl+F	한 화면 앞으로 이동
Ctrl+D	반 화면 앞으로 이동
Ctrl+B	한 화면 뒤로 이동
Ctrl+U	반 화면 뒤로 이동

문자와 행 삽입	
a	커서 오른쪽에 문자 삽입
A	커서 오른쪽, 행의 끝에 문자 삽입
i	커서 왼쪽에 문자 삽입
I	커서 왼쪽, 행의 처음에 문자 삽입
o	커서 아래에 행 삽입
O	커서 위에 행 삽입

텍스트 변경	
cw	단어 변경
cc	행 변경
C	커서 오른쪽에 행 일부 변경
s	커서가 위치한 문자열 대체
r	커서가 위치한 문자를다른 문자로대체
r+Return	행 분리
J	현재 행과 아래 행 결합
xp	커서가 위치한 문자와 오른쪽에 있는 문자 교환
~	문자형(대문자, 소문자)변경
u	이전 명령 취소
U	행 변경 사항 취소
:u	이전에 최종 행 취소

텍스트 삭제	
x	문자 삭제
dw	단어 삭제
dd	행 삭제
D	커서 오른쪽의 행 일부 삭제
:5, 10 d	5-10번째 행 삭제

텍스트 복사 및 이동	
yy	행 yank 또는 복사
Y	행 yank 또는 복사
P	yank 되거나 삭제된 행을 현재 행 위에 삽입
p	yank 되거나 삭제된 행을 현재 행 아래에 삽입
:1,2 co 3	1-2 행을 3행 다음으로 복사
:4,5 m 6	4-5 행을 6행 다음으로 이동

행 번호 설정	
:set nu	행 번호 표시
:set nonu	행 번호 숨기기

행 찾기	
G	화일의 마지막 행으로 가기
21G	21번째 행으로 가기

탐색 및 대체	
/string/	스트링 탐색
?string?	스트링을 역방향으로 탐색
n	스트링의 다음(또는 이전)발생 찾기

화일을 화일로 삽입	
:r filename	커서 다음에 화일 삽입(읽어들임)
:34 r filename	화일을 34번째 행 다음에 삽입

보관 및 종료	
:w	변경 사항 보관(버퍼 기록)
:w filename	버퍼를 화일로 기록
:wq	변경 사항 보관 후 vi 종료
ZZ	변경 사항 보관 후 vi 종료
:q!	변경 사항 보관하지 않고 종료

쉘 (Shell) 이란

DOS 버전 4에서는 커맨드 라인(Command line) 대신, 윈도우 표시나 메뉴를 선택하여 명령을 실행하는 가시(visual) 프로그램이 포함되어 있다. 이 프로그램은 MS-DOS 쉘이라고 불리우며 표준적인 UNIX 쉘과는 다르다. 쉘이란 간단히 말해서 사용자가 입력한 명령어를 해석하는 명령어 해석기이다.

우리가 login하고 password를 맞게 입력하면 프롬프트(%)가 디스플레이(display)되며, 이때부터 쉘이 수행하기 시작하고 사용자가 명령을 입력하기를 기다린다. 명령을 입력하고 리턴키를 누를 때마다 쉘은 사용자가 친 행을 분석하여 요청한 바를 수행하기 시작한다.

만일 사용자가 특수 프로그램을 실행하기를 요청한다면 쉘은 이름지어진 프로그램을 찾을 때까지 디스크를 찾는다. 일단 쉘이 프로그램을 찾게 되면 쉘은 프로그램을 실행키 위해 커널에게 요청을 하고 프로그램을 끝마칠 때까지 정지한다. 커널은 기억 장소에서 특수 프로그램을 복사하고 실행한다. 이 복사 프로그램을 프로세스라 한다.

명령이 다 실행되면 다음 명령을 기다리는 쉘로 다시 돌아가서 제어하게 된다. 사용자가 컴퓨터 시스템 사용을 종료하면 쉘의 실행은 끝나고

UNIX 시스템은 터미널에서 다른 누군가가 로그인하기를 기다리는 새로운 프로세스를 시작한다.

쉘은 누구나가 만들 수 있는 프로그램이며, 그것은 시스템에서 특별한 특권을 가지고 있지않다.

UNIX 시스템에서는 세가지 기본적인 쉘(shell)이 있는데 그중 오리지널 UNIX 쉘은 본(Bourne) 쉘이라고 불리우고 이 쉘은 1970년대 초에 AT&T벨 연구소에서 Stephen R. Bourne에 의해 개발되었다. 이것은 공식 UNIX 쉘로서 최초로 개발되었기 때문에 가장 널리 사용하고 있다.

또 하나의 쉘은 C 쉘이라고 불리우고, 1970년 중엽 캘리포니아 대학에서 William Joy를 중심으로 하는 사람들에 의해 개발되었다. 이것은 C언어로 부터 많은 기능을 도입하고, Bourne쉘보다도 다채로운 기능을 가지며 공식 쉘은 아니지만 UNIX 시스템 상에서 많이 이용하고 있다.

세번째 쉘로는 콘(Korn)쉘이라고 불리우며, 1980년대 초에 David Korn에 의해 개발되었다. Korn쉘은 Bourne쉘과 C쉘의 장점을 합친 것인데, 해가 갈수록 널리 사용되어 오고 있다. AT&T 쉘과 100% 호환된다.

Bourne 쉘과 C쉘, Korn쉘은 화면상에 표시되는 프롬프트에 의해 아래와 같이 구별할 수 있다.

Bourne, Korn 쉘 : %
C쉘 : %

UNIX 명령어

① find : 디렉토리에서 특정의 화일을 찾아 낼 수 있다. 검색 조건으로서 화일명이나 화일형, 소유자, 그룹, 액세스권 및 최종 갱신일을 지정할 수 있다.

형식 : find path명 화일의 선택 방법 동작

화면에 출력 ↓

예) %find /usr −name kkk −print

↑ ↑

화일명이 kkk라는 화일검색

/usr과 그 서브디렉토리 부터 검색

find 명령의 이름과 형(type)에 의한 검색키

옵 션	의 미
−typef	화일의 검색
−type d	디렉토리의 검색
−name(화일)	(화일)을 검색
−newer(화일)	(화일)보다도 나중에 변경된화일의 검색
−user(사용자)	(사용자)가 소유하는 화일의 검색
−group(그룹)	(그룹)의 멤버가 소유하는 화일의 검색
−size(블록수)	크기가 (블록수)인 화일을 검색
−links(링크수)	(링크수)의 화일을 검색
−atime(d일수)	(d일수)전에 액세스된 화일을 검색
−ctime(d일수)	(d일수)전에 작성된 화일을 검색
−mtime(d일수)	(d일수)전에 변경된 화일을 검색

예) % find /bin −type d −group informix
−ctime 30 print ↵
/bin/time
/bin/place
%

* 해설
/bin: 검색을 /bin과 그 서브 디렉토리에 한정한다.
−type d : 디렉토리인가를 검색한다.
group informix: informix 그룹의 멤버가 소유하는 화일이나 디렉토리를 검색한다.
−ctime 30 : 30일 전에 변경된 화일 검색
−print : 디렉토리명을 화면에 표시한다.

② grep : 하나의 화일 또는 여러개의 화일 내의 텍스트를 검색하는 명령어

DOS의 find 명령은 UNIX의 find 명령의 이름을 빌려 쓰고 있으나, 그 기능은 UNIX의 grep 명령에 대응된다.

형식 : grep 옵션 "문자열" 화일명

예) DOS의 경우

C:\> find "apple" a:kkk ↵
−−−−− a:kkk
····· apple ····

C:\>

UNIX의 경우

% grep apple kkk ↵

　　… apple …

%

이들 두 예에서 보인 바와 같이 DOS의 find 명령과 UNIX의 grep 명령은 매우 비슷하지만, UNIX의 grep 명령은 다시금 몇개의 기능을 가지고 있다. grep과 DOS에서 find의 그밖의 유사점을 나타내기 위해 디렉토리내에서 apple이라는 문자를 포함한 화일명을 표시시켜 보자.

예) DOS의 경우

C:\> dir ¦ find "apple" ↵

apple －220 721 May 10 12:00

C:\>

UNIX의 경우

% ls －1 ¦ grep apple ↵

－rw x rw －r－x 2 kkk 333 May 10 12:00 apple－220

%

두 명령의 차이점은 DOS의 find 명령에서는 검색 문자열의 이중 인용부호(")가 반드시 필요하지만, grep에서는 정규 문자열 속에 정규 표현이나 복수의 어구가 존재하는 경우만 필요하다. 또 find로 복수의 화일을 검색하는 경우, 각각의 화일명을 명기해야 하지만, grep에서는 화일의 그룹을 나타내는 와일드 카드 문자를 사용할 수 있다.

예) % grep apple kkk* ↵

kkkabc:apple

kkk:* apple,

%

옵션	의　미
－b	각 행의 앞에 블록 번호를 표시한다.
－c	일치된 행의 수를 표시하지만, 행 자신은 표시하지 않는다.
－i	검색에서 대문자와 소문자를 구별하지 않는다.
－l	텍스트 행은 표시시키지 않고, 화일명만을 표시한다.
－n	텍스트와 함께 행 번호를 표시한다.
－s	검색에서의 화일의 에러 메시지를 표시하지 않는다.
－v	검색의 의미를 역으로 만든다(검색 문자열을 포함하지 않는 행만을 표시한다).

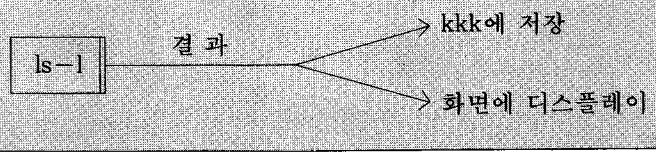
이 외에, UNIX의 grep 명령에는 fgrep(고속 grep)과 egrep(확장 grep)이라는 두개의 비슷한 명령어가 있다. fgrep는 문자열밖에 지정할 수 없으나, grep은 정규 표현을 지정할 수 있으며, egrep은 복잡한 식을 사용할 수 있다.

③ tee : 출력을 화면에 표시하고, 화일에 써 넣기라는 두 작업을 동시에 하는 명령어로써 파이프와 동시에 사용한다. 파이프는 참고란을 참조하자.

예) % ls－1 ¦ tee kkk ↵

```
drwx--x---  1 root   367  Mar 15 08:37 tta/
-rwxrw----  2 atom  1290  Jan 18 12:40 end.odd*
-rwx------  1 root  3980  May 24 10:03 duck
lrwxrwxrwx  9 atom  8825  Jun 02 11:00 kim -> /usr/kim*
```

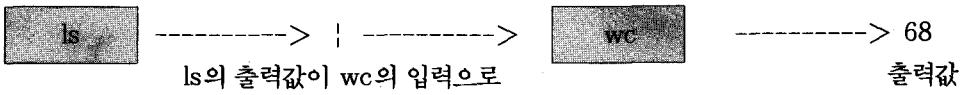
* 해설



＊참　고＊

파이프 : 한 명령의 출력이 직접 다른 명령의 입력으로 사용될 수 있다. 이와같이 연결된 일련의 명령을 파이프 라인이라 한다. 입출력 연결을 위한 기호는 파이프라 하는 수직선(¦)이다.

예) % ls ¦ wc ↵
68



와일드 카드문자 : UNIX 시스템에서는 DOS와 마찬가지로 와일드 카드 문자(wild card character)라는 화일명내의 임의의 그룹을 나타내는 문자를 제공함으로써 한번에 두 화일 이상에 대해 작업할 수 있다. 별표(＊)는 화일명에서 임의의 갯수의 문자를 가리킨다. 물음표(?)는 화일명내의 임의의 한문자를 나타낸다. 따라서 이것을 사용하여 지정된 패턴과 일치하고 지정된 문자수와 같은 문자수로 된 화일을 선택할 수 있다. 대괄호 문자([])은 정확한 문자열이 아니라 괄호내의 어떤 종류의 문자열을 지정할 경우에 사용된다.

예) % ls m＊ ↵
mwm mesg miujbhggdhf machine

% ls a?????a ↵
arizona alabama

% ls k[a－z]m ↵
kim kkm kam kzm

④ write : 단말간, 호스트 단말 간 송수신하는 명령어.

예)

```
% write candy ↵
안녕 하세요 ↵
^D
%
```

〈로그인 명이 atom인 경우〉

상대방이 mesg n 상태이면
메세지를 송신할 수 없다.
write : Permission denied

```
Message from atom@diehard2 on ttyp2 at 16:48
안녕하세요
% write atom ↵
```

〈로그인 명이 candy인 경우〉

mesg y 상태이면 상대방이 송신하는 내용이 화면에 나타나며, 중요한 작업시 방해받고 싶지 않을때는 mesg n을 입력한다.
상호통신을 원하면 write 명령을 입력한다.

⑤ mail : 전자 우편을 주고 받는 명령어. 이것은 편지를 읽고, 쓰고, 주고, 받고, 보관하고, 삭제하는 기능을 제공한다.

예) 전자우편의 송신

```
% mail candy ↵
안녕하세요. 92년 6월1일에 제 3 회의실에서 UNIX 교육이 있습니다.
^ D
%
```

전자우편의 수신

```
% mail ↵
·
·   -----> 메세지가 있으면 화면에 디스플레이 된다.
·          메세지가 없으면 프롬프트(%)가 바로 나타난다.
?t ↵
usage
q          quit
x          exit without changing mail
p          print
s[file]    save(default mbox)
w[file]    same without header
—          print previous
d          delete
t          next(no delete)
m user     mail to user
! cmd      execute cmd
?
```

⑥ cut : 자료 화일이나 출력 명령으로부터 여러 가지 다양한 자료 필드를 추출할 때 사용하는 명령어

cut—c chars file

예) % cut —cl, 13, 50 kkk ↵

⑦ set prompt : 쉘 프롬프트를 설정하는 명령어

예) % set prompt="＼ ! %" ↵

→ 프롬프트가 일련의 번호로 바뀐다.

```
% set prompt="UFO" ↵
↳                  프롬프트가 UFO로 바뀐다.
```

⑧ sort : 화일 내의 각 행을 정렬하는 명령어 UNIX의 sort 명령은 DOS의 sort 명령에 대응된다. DOS의 sort 명령을 사용할 때는 입력을 입력 화일에서 리디렉트해야 하지만, UNIX의 sort 명령에서는 리디렉트할 필요없이 입력 화일명을 지정할 수 있다. UNIX와 DOS를 막론하고 소트는 첫번째 컬럼부터 행해진다. UNIX의 sort 명령은 임의의 필드로부터 소트를 시작할 수 있다.

예) % cat kkkk ↵

```
blueberry   10    92. 10. 11
apple        5    93. 12. 10
pear         7    86.  3.  7
```

sort 명령에서는 공백을 필드를 구분하는 구분자로 간주한다.

화일의 필드

화일명	갯 수	날 짜
blueberry	10	92.10.11
apple	5	93. 12. 10
pear	7	86. 3. 7

```
% sort +2 kkk ↵
pear         7    86.  3.  7
blueberry   10    92. 10. 11
apple        5    93. 12. 10
```

이와 같은 화일을 날짜순으로 sort하고 싶으면 옵션 +2(앞에서 2 필드를 뛰어넘고)를 지정 3번째 필드순으로 sort 할 수 있다.

DOS의 sort 명령에는 옵션이 하나밖에 없으나 UNIX의 sort 명령에는 많은 옵션이 있다.

옵션	의 미
—b	소트시에 뒤의 공백을 무시한다.
—d	사전순으로 소트한다.

-f	대문자와 소문자를 무시한다.
-i	비표시문자를 무시한다.
-n	필드를 공백, -, 0, 소수점을 포함한 수치로서 다룬다.(-b옵션이 자동적으로 설정된다)
-M	월(JAN, FEB, MAR,…, DEC)로 소트한다.
-o〈화일〉	결과를 〈화일〉에 출력한다(> 〈화일〉과 같다)
-r	역순으로 소트한다(Z부터 A, 9부터 0으로).
-t〈문자〉	〈문자〉를 필드 세파레이터(separator)로 만든다.
-u	동일 내용의 행을 삭제한다.
-y〈수〉	소트 영역에 〈수〉 킬로바이트를 확보한다.
-z〈수〉	입력행의 최대 문자수를 〈수〉로 만든다.

예) % sortk -nr +1 kkk ↵

```
blueberry   10    92. 10. 11
pear         7    86.  3.  7
apple        5    93. 12. 10
```

두번째 필드로 소트하고 두 개의 옵션을 사용한다.
-n(수치)와 -r(역순)옵션을 결합하여 사용한 예이다.

⑨ lp : 화일을 프린트하는 명령어 UNIX에서 텍스트를 프린트하려면 lp(line printer)명령을 사용한다. UNIX와 DOS의 큰 차이점은, 하나의 UNIX 시스템에 많은 프린터를 접속할 수 있는 점이다. 이 때, 하나의 프린터가 디폴트 프린터에 설정된다. lp 명령은 UNIX 시스템상의 임의의 프린터에 텍스트를 보낼 수 있다. 디폴트 프린터에 프린트하는 경우는 아래와 같다.

예) % lp kkk ↵

```
Request id is 220
```

시스템이 여러 대의 프린터를 가지고 있는 경우, 사용자는 동작 중의 임의의 프린터에 프린트 잡(job)을 보낼 수 있다.

예) % lpr -Pprinter2 kkk ↵

-P는 옵션이고 printer2는 프린터명이며 kkk는 인쇄할 화일명이다.

표지 페이지를 인쇄하는 것이 필요하지 않을 경우, 그것이 인쇄되지 않도록 지정하기 위해 lpr -h를 사용한다.
화일을 인쇄할 때, 그 화일의 사본은 인쇄 대기 행렬(Queue)로 보내진다. 인쇄 대기 행렬에는 해당 프린터에서 인쇄되기를 기다리는 모든 화일이 있다. 인쇄되기를 기다리는 각 화일은 프린트 작업(job)이라 하며 각 프린터 작업은 그것과 연관된 번호를 가집니다.
lpq 명령은 인쇄 대기 행렬 내에 있는 내용 및 프린터가 올바르게 인쇄 중인지의 여부를 알려준다.
다른 프린터의 상태를 점검하려면, -P 프린터 옵션을 사용한다.

예) % lpq ↵

```
printer 1 is ready and printing
Rank  Owner  Job  Files  Total  Size
active candy  16   kkk    7402   bytes
active atom   17   JJJ    1024   bytes
```

사용자가 화일을 인쇄 대기 행렬로 보낸 후 그것을 인쇄하지 않으려면 lprm 명령을 사용한다.

```
% lprm 16 ↵
printerl : dfA016 candy dequeued
```

사용자의 인쇄 작업을 한번에 모두 제거하려면 lprm뒤에 하이픈(-)을 입력하면 된다.

명 령	동 작
lp	화일인쇄
lpr	화일인쇄
lpr-h	표제 없이 인쇄
lpr-p	프린트를 지정하여 인쇄
lpq	프린트의 상태보기
lprm	인쇄 작업 보기

⑩ tty : 현재 사용하는 단말 장치를 알아 보는 명령어.

window를 지원하는 단말기에서는 여러 window를 띄울 수 있는데 각각의 window는 각각의 모니터(단말기)라고 할 수 있다. 5개의 window를 띄울 경우 5개의 단말기를 사용하고 있다고 할 수 있다.

예) % tty ↵
/dev/ttyp 2

⑪ wc : 화일에 있는 단어, 라인, 문자수를 표시하는 명령어

예) % who ¦ wc ↵
7 14 970
who 명령의 결과는 7줄로 되어있고 14단어로 구성되었으며, 970 문자로 되어 있음을 알수 있다.

⑫ set : 시스템의 환경정보를 출력하는 명령어

예) % set ↵

```
argv        ( )
cwd         /user1/candy
filec
history     32
home        /user1/candy
lpath       ( )
noclobber
path        (. /user1/candy/bin/usr/local/usr
            /ucb/usr/bin)
prompt      /user1/candy%
shell       /bin/csh
status      0
term        sun-cmd
user        candy
```

cwd : 현재 명령을 실행하고 있는 디렉토리
history 32 : 명령을 32개까지 기억 가능
home : 시스템 관리자에 의해 할당된 홈 디렉토리
path : 명령어가 실행되어지는 경로. 즉 명령들이 들어 있는 디렉토리
prompt : 실제 명령어를 쓸 때 떨어지는 기호
shell : sh는 본(Bourne)쉘이고,ksh는 콘(Korn)쉘이며 csh는 C쉘이다.
term : 터미널 타입
user : Login 명

⑬ & : 명령을 백그라운드로 실행하는 명령어.
멀티타스킹(multi-tasking)은 UNIX의 주요기능인데, UNIX와 DOS를 명확히 구별한다. 멀티타스킹을 사용하면 각 사용자는 시스템 상에서 동시에 복수의 프로세스를 실행한다. 멀티타스크에는 포어그라운드(foreground) 프로세스와 백그라운드 프로세스가 있는데, 포그라운드 프로세스는 처음부터 끝까지 사용자와의 대화를 필요로 하고 항상 화면을 사용한다. 백그라운드 프로세스는 사용자가 명령을 내리고 다른 프로세스를 기동시킬 수 있다. 예를 들어 화일의 복사를 백그라운드 프로세스로 기동하고, 계속해서 포어그라운드로 화일을 편집할 수 있다. 백그라운드로 명령을 수행하기 위해서는 명령어 뒤에 & (앰퍼샌드)를 입력하면 된다.

예) % lpr kkk & ↵
311

쉘은 백그라운드 프로세스를 개시하고, 프로세스 번호(311)를 커널에 의해 할당된다.

이제까지 UNIX 시스템을 사용하는 데 필요한 명령어를 살펴 보았다. 지면과 시간의 제약으로 좀 더 상세한 내용을 소개해 드리지 못한 것을 아쉬워 하며 작은 글이 나마 도움이 되었으면 하는 마음으로 이글을 마치고자 한다.

UNIX 명령

UNIX 명령	기 능
cat	화일 편집 및 보기
vi	화일 편집
find	특정 화일 검색
grep	특정 텍스트 검색
tee	동시에 화일저장과 화면에 출력
write	단말간 송수신 명령
mail	전자우편
cut	필드 추출
set prompt	쉘 프롬프트 설정
sort	화일 정렬
lpr	화일 인쇄
tty	단말명 보기
wc	화일내 정보보기
set	시스템 환경 보기
&	백그라운드 명령