**Report**

**서버관리실무**

**제목 : 13주차 과제**

**학과 : 컴퓨터소프트웨어과  
학번 : 2017100301  
이름 : 이상남  
 담당교수 : 김형철 교수님  
제출일 : 2020.11.24**

1. **리눅스 편집기**

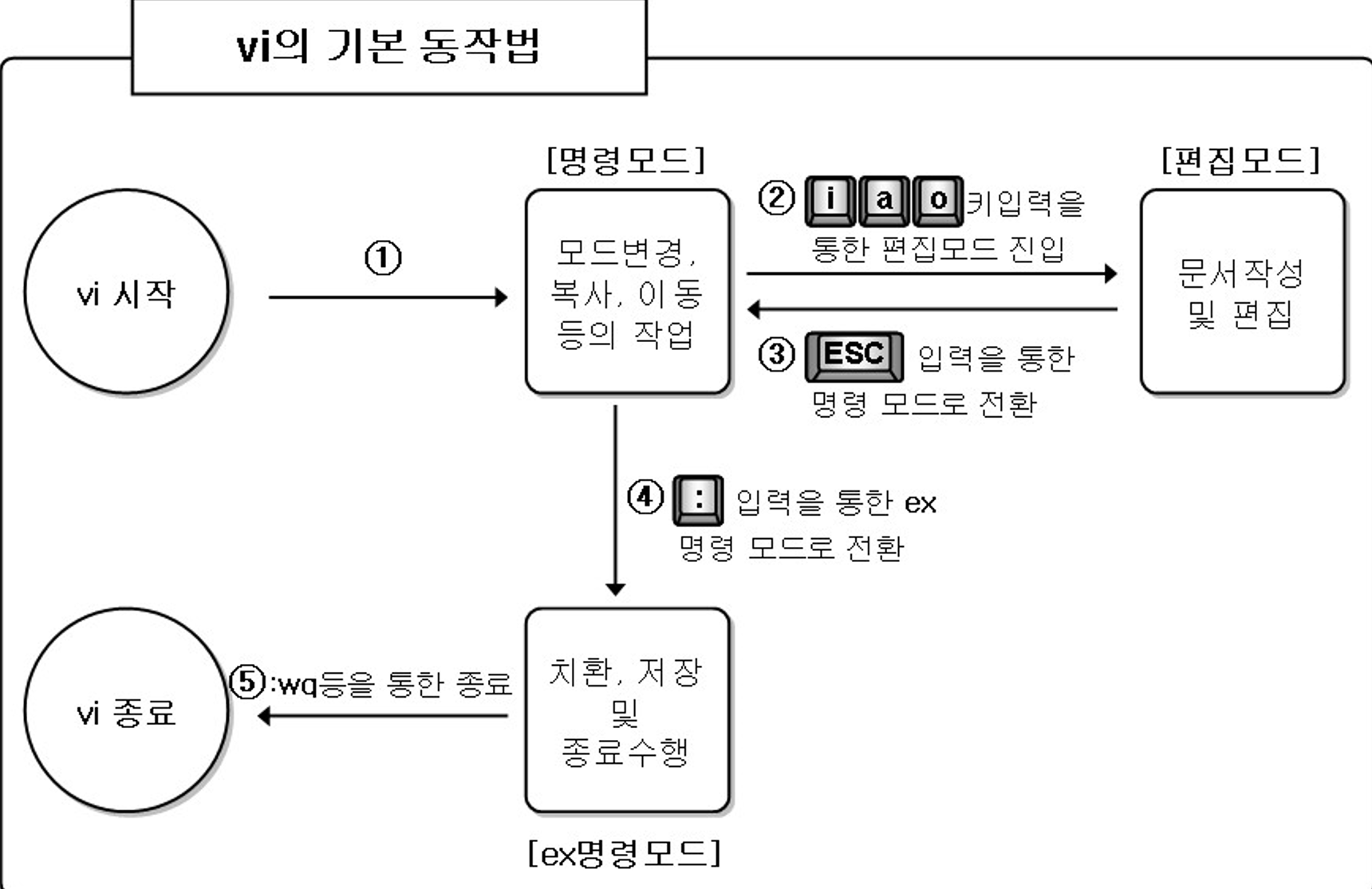
* **편집기(Editor)는 텍스트파일을 생성 또는 수정할 때 사용하는 프로그램 이다.**
* **리눅스는 환경설정파일을 비롯하여 중요 정보를 담고 있는 파일들의 대 부분이 텍스트파일로 이루어져 있다**
* **리눅스에서 전통적으로 사용하는 편집기에는 vi, emacs, pico를 들 수 있다.**
* **현재 리눅스 배포판에 사용되는 vi는 오리지널버전은 아니고, 기능을 대폭 강화하여 만든 vim(ViImproved)이다**
* **설치가 되어 있지 않다면 ‘yum install emacs’를 실행한다.**

1. **vi편집기 시작하기.**

* **셀 명령행에서 vi[피일명] 이라고 입력하면 된다.**
* **vi를 처음 실행하면 명령모드로 진입하고, 명령모드상태에서 입력명령**

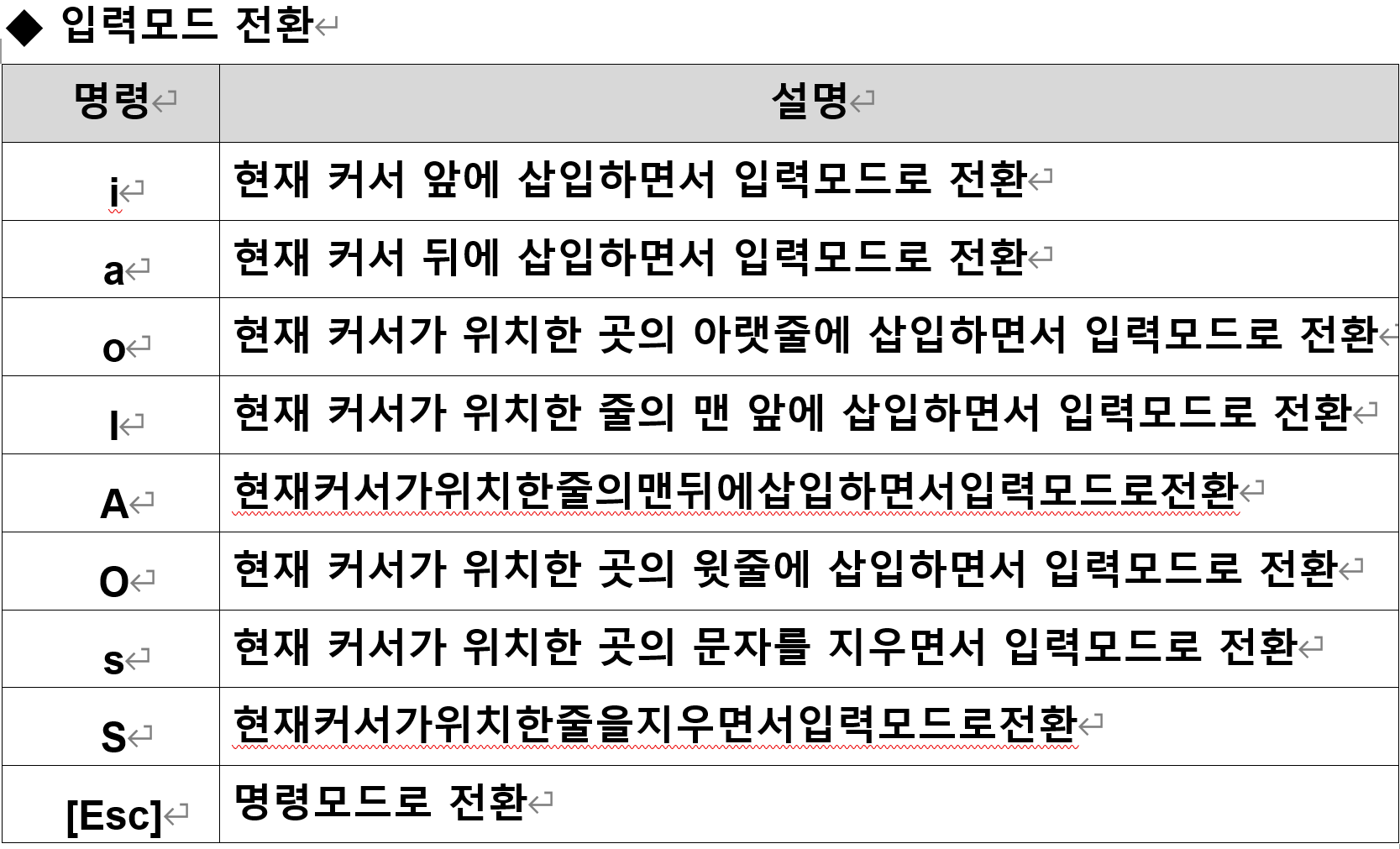
**(i,I,a,A,o,O 등)을 실행하면 입력모드로 전환된다.**

* **입력모드에서 [Esc]키를 누르면 명령모드로 돌아온다.**

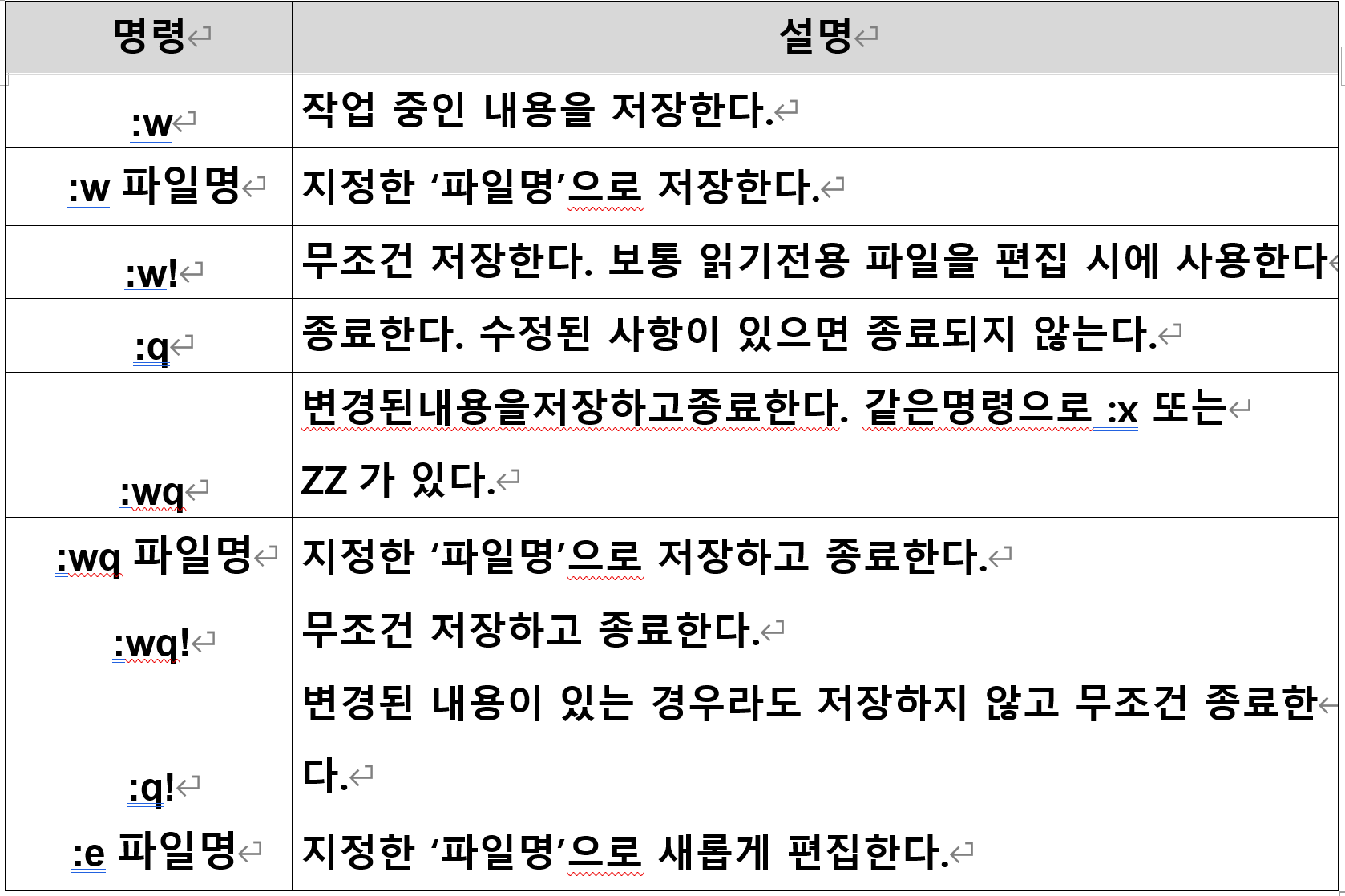
****

****

**.**

****

* **ex 명령모드에서 사용하는 주요 명령어**

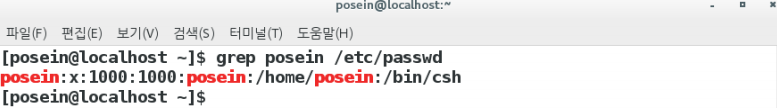
****

1. **셸의 개념 및 특징**

* **셸(Shell)은 커널(Kernel)과 사용자간의 다리역할을 하는 것으로 사용자 로부터 명령을 받아 그것을 해석하고 프로그램을 실행하는 역할을 한다**
* **셸은 사용자가 시스템에 로그인(login)을 하게 되면 각 사용자에게 설정 된 셸이 부여되면서 다양한 명령을 수행할 수 있게 된다.**

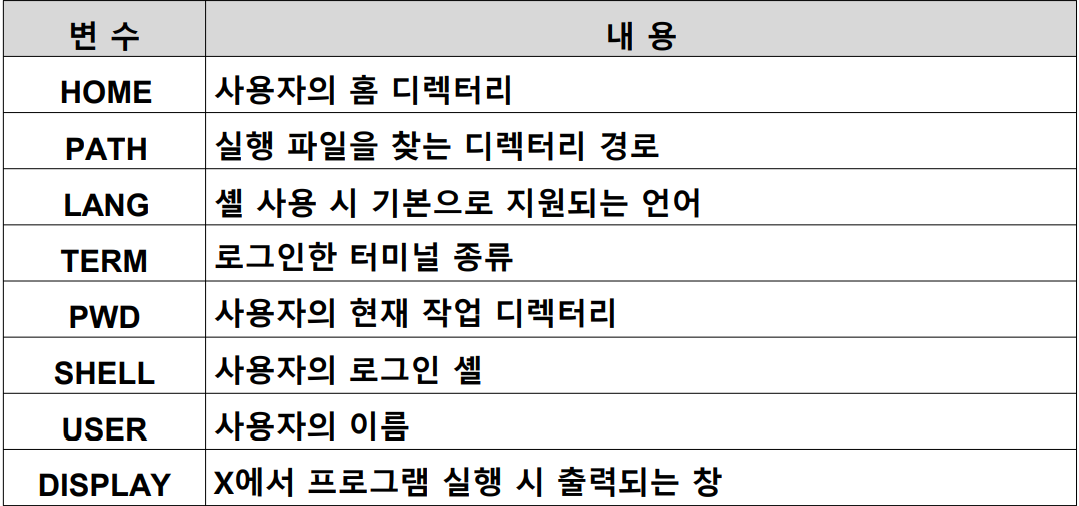
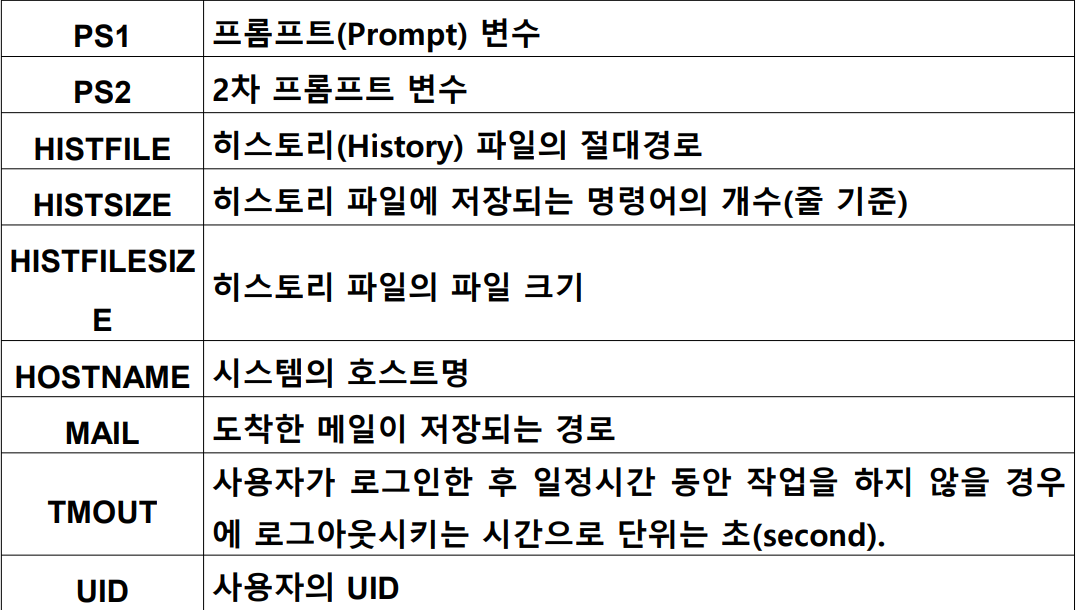
1. **셸의 확인과 변경**

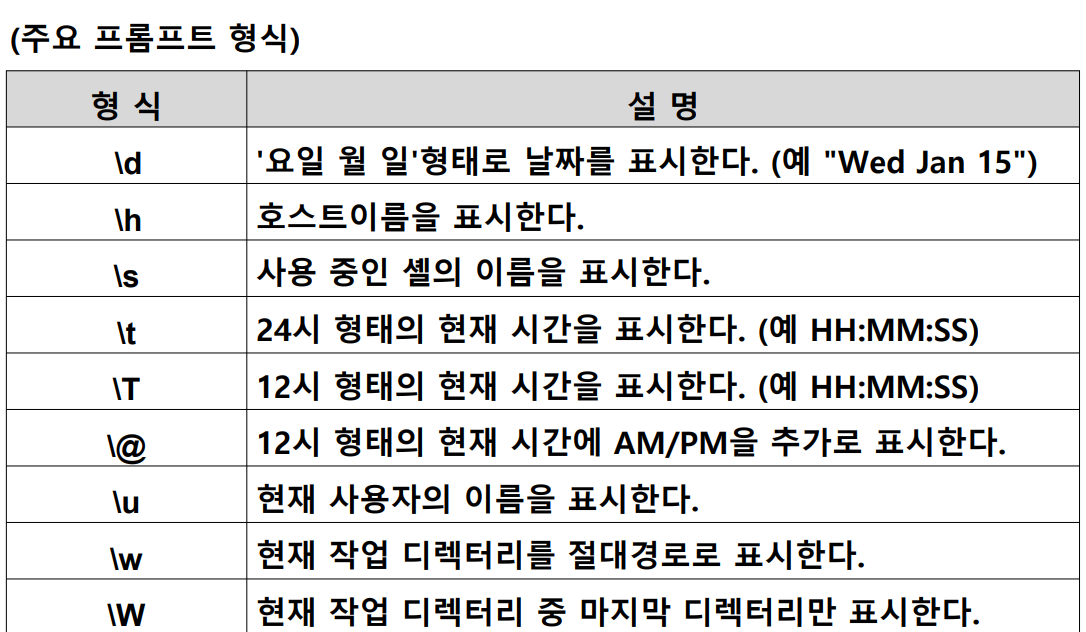
**◆ 셸의 확인  
• 시스템에 로그인한 후에 사용 중인 셸을 확인하려면 로그인 셸관련 환경 변수 SHELL을 통해 가능하다. • 명령 프롬프트 상에서 ‘echo $SHELL’이라고 실행하면 확인가능하다.   
•다른 셸을 사용하기 위해 변경 가능한 셸의 확인은 ‘chsh –l’ 명령 이나 ‘cat /etc/shells’로 확인할 수 있다.**

**◆ 사용자의 로그인 셸 정보 확인  
 • 사용자의 로그인 셸 정보는 /etc/passwd의 7번째 필드에 기록되므로 다음과 같이 확인할 수 있다.  
**

**5. 셸 변수와 환경변수  
 ◆ 셸 변수   
• 특정한 셸(Shell)에서만 적용되는 변수를 말한다.   
• 리눅스에서는 명령행에서 ‘변수명=값’ 형태로 지정하여 사용할 수 있고, 변수값을 출력할 때는 변수명 앞에 $을 붙이고 echo명령으로 확인할 수 있다.**

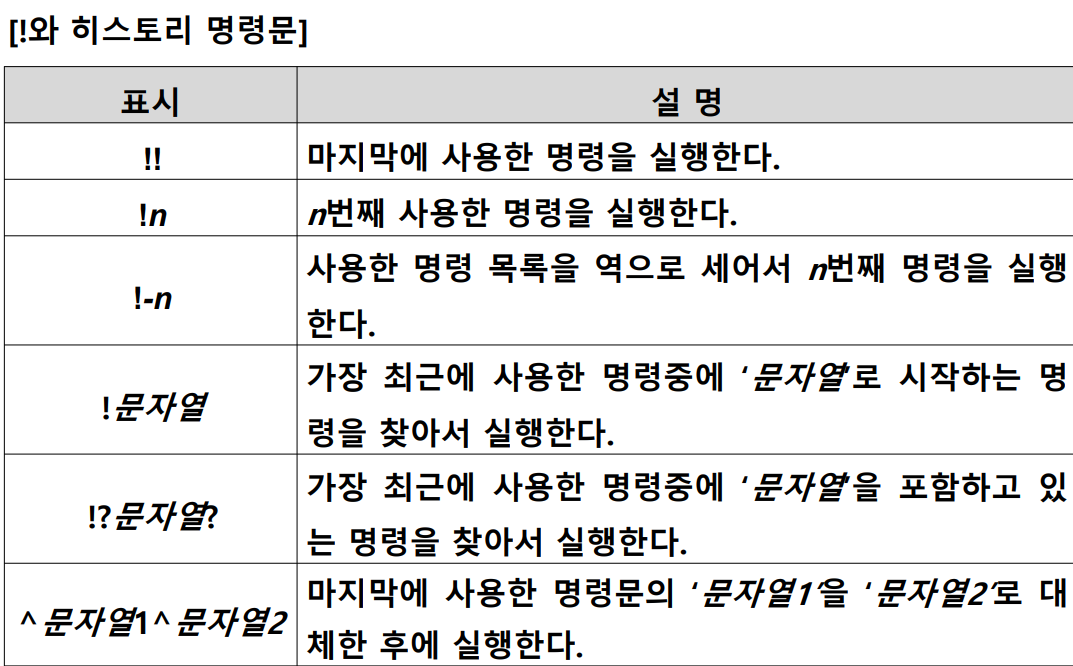
**◆ 환경변수   
• 환경변수란 프롬프트 변경, PATH 변경 등과 같이 셸의 환경을 정의하는 중요한 역할을 수행하는 변수를 말한다.   
• 미리 예약된 변수명을 사용하고, bash에서는 PATH, SHELL 등과 같이 대문자로 된 변수로 구성되어 있다.  
 • 현재 설정된 전체 환경변수의 값은 env 명령으로 확인 가능하고, 특정 환경변수의 값 확인과 설정은 일반 셸 변수 설정과 같다.**

**◆ 주요 환경변수   
• 배시셸 에서는 다양한 환경변수를 제공하고 있는데, 많이 사용되는 환경 변수를 나열하면 다음과 같다.  
  
**

**◆ 환경변수의 사용 및 변경   
• 환경변수를 사용하여 각 사용자 고유의 셸 환경을 구축할 수 있고, 명령 어와 결합하여 이용할 수 있다.  
[사용 예] $ mkdir $HOME/data → 해당 사용자 홈 디렉터리안에 data라는 디렉터리를 만든다  
**

**6. bash의 주요 기능   
◆ 명령행 완성 기능  
• 명령 입력시 글자 일부분만 입력하고 [TAB]키를 눌러 나머지 부분을 자 동 완성 시키는 기능이다.   
• 보통 파일이나 디렉터리 입력 시에 첫 글자만 입력하고 [TAB]키를 누르면 뒷부분을 자동으로 완성시켜준다.  
• 첫 글자 입력 후 [TAB]키를 눌렀을 때 여러 개가 존재하면 [TAB]키를 한 번 더 눌러 목록을 확인할 수 있다.   
• 추가로 원하는 글자 입력 후에 다시 [TAB]키를 눌러 완성시키는 기능이다.**

**◆ 명령어 History 기능   
• bash에서는 입력 후 실행했던 모든 명령들은 히스토리 리스트 버퍼에 스택으로 저장된다.   
• 이렇게 저장된 명령들은 위/아래 방향키를 사용하여 검색 및 편집하여 특정 명령을 반복해서 수행할 수 있다.   
• history라는 명령을 입력하면 히스토리 리스트에 있는 명령어들이 출력 된다.   
• 또한 사용자들이 실행한 명령들은 각 사용자의 홈 디렉터리 안에 .bash\_history라는 파일에 추가로 기록되며, 로그아웃할 때 메모리에 기억된 명령의 목록을 파일에 저장한다.  
(1) 관련명령어 : history  
• 사용자가 입력한 명령어를 확인하는 명령으로 ‘!’로 대체하여 사용할 수 있다. [사용법] $ history [숫자값]**

****

**[사용 예]**

**$ history → 히스토리에 저장된 명령어 목록을 출력한다.   
$ history 5 → 최근에 입력한 마지막 5개의 명령어 목록을 출력한다.   
$ !! → 가장 마지막에 실행한 명령을 재실행한다.   
$ !100 → 히스토리 목록의 번호 중에서 100번에 해당하는 명령을 실행한다.   
$ !-4 → 히스토리 명령 목록에서 4만큼 거슬러 올라가서 해당 명령을 실행한다.  
$ !a → 최근 사용한 명령 중에 a로 시작하는 명령을 찾아서 실행한다.   
$!?al → 최근에 사용한 명령 중에 al이라는 문자열을 포함한 명령을 찾아서 실행한다.**