

# 리눅스시스템 및 응용

Week 6

## Shell Programming

# 학습목표

---

- Concept of the Shell
- Environment of Shell
- Shell as programming language

# What is Shell?

---

- Shell 이란 ?
  - 명령어 해석기
  - 시스템과 사용자간의 대화창구 == Shell 인터페이스
- Shell 의 기능 3가지
  - Login시 명령어 해석 및 실행 기능
  - Kernel 및 응용프로그램과의 인터페이스
  - 프로그래밍 언어로서의 Shell
- Shell의종류
  - Bourne Shell → Bourne Again Shell (bash) (\$)
  - csh → 확장 csh (tcsh) (%)
  - Korn Shell, V Shell....
  - 기본기능이나 작동법은 동일

---

- Shell 기능

- 내부명령, 외부명령 실행 □ 명령프롬프트 (whereis !)
- 여러명령 연결 - 파이프(|)
- 입/출력 리다이렉션 ( >, <, >>, <<, 2> ) : 명령방향 전환
- 특수문자 해석 및 치환, 명령어 치환
- 지역/환경변수 관리
- 스크립트 프로그래밍 언어 → 자동화 작업(batch processing)

---

- Shell 프롬프트

- echo SHELL
- echo는 뒤에 나오는 문자열 출력
- echo \$SHELL
- '\$'는 뒤 스트링을 변수로 인식하고 변수내용 출력

- Meta Characters

- Shell이 해석하는 특수한 기능을 가진 문자 □ \$
- 메타문자 해석 금지 → \ , " , "
- 예) echo ₩SHELL

---

- Linux의 기본 Shell

- : GNU Bourne Again Shell (/bin/bash) → C 문법을 많이 이용

- bash의 환경설정 파일

- 사용자별 환경설정 파일 (**cd ; ls -aF**)

- .bash\_logout, .bash\_profile, .bashrc

- 원본파일 /etc/skel/ 디렉토리에 존재
- 새로운 사용자 추가시마다 각 사용자의 홈디렉토리에 복사됨.
- "."으로 시작하는 파일은 hidden file

- 모든사용자 공통의 환경설정 파일

- /etc/profile , /etc/bashrc

- 환경설정 파일 인식 순서

- /etc/profile → ~/.bash\_profile → ~/.bashrc → /etc/bashrc

- 
- 현재 path 관련 변수 확인
    - % echo \$PATH
  - path 디렉토리 추가
    - 전체 사용자에게 추가 : /etc/profile 편집
    - 특정 사용자에게 추가 : ~/.bash\_profile 편집
    - 현재 로그에서만 추가 (로그아웃시 소멸)
      - echo \$PATH
      - PATH=\$PATH:<추가할 디렉토리>
  - 개인별 별칭(alias) 설정
    - 단축명령어 등록
    - ~/.bashrc 편집 및 활성화( . .bashrc)
      - 예) alias ls='ls -a'

---

- 사용자 프롬프트 변경

- 절대경로 표시 : /etc/bashrc 편집
- PS1="[Wu@Wh WW] WW\$  
→ PS1="[Wu@Wh W\$PWD] WW\$



- 
- 로그아웃시의 실행
    - ~/.bash\_logout 편집
  - 도스키 및 자동 완성
    - 도스키: 기존사용 명령어 다시불러오기
      - 상/하 키 사용
  - 자동완성
    - 일부 앞글자만 입력후 [TAB] 키 사용
      - 디렉토리 이동시 편리함

- 
- `ps | sort | more`
  - `ls -l > out.txt`
  - `ps >> out.txt`
    - 표준에러 출력 redirection, 2>와 동일
    - `Kill -9 1234 > out.txt 2>&1`
    - `Kill -9 1234 > out.txt 2> error.txt`

---

sh           <enter>

\$ for i in \*

: 현재 디렉토리의 모든 파일

> do

: for의 do

> if grep -l root \$i

: if문

> then

: 조건 만족시 실행

> more \$i

> fi

: if의 다음

> done

\$

: do 의 다음

- 
- 이전 예제를 파일로 작성하면 셸프프로그램이 된다

```
#!/bin/sh
```

```
# exam1.sh
```

```
# 예제 1 #으로 라인을 시작하면 comment 주석문
```

```
for i in *
```

```
do
```

```
if grep -l root $i then
```

```
more $i fi done
```

```
exit 0
```

---

- 셸 스크립트 파일 실행 방법

- 1. 스크립트 실행 명령어로 실행

```
/bin/sh exam1.sh
```

- 2. 스크립트파일을 실행가능으로 변경 후 실행

```
chmod +x exam1.sh
```

```
/exam1.sh <return>
```

**Question?**