

[리눅스] 05. 셸(Shell)과 커널(Kernel)

05. 셸(shell)과 커널

학습 목표

- 가. 셸과 커널의 개념에 대해 이해해 본다.
- 나. 간단한 셸 스크립트 동작에 대해 이해할 수 있다.

목차

05. 셸(shell)과 커널(Kernel)

5-1 셸과 커널(Shell, Kernel)

5-2 셸의 종류 간단 비교 : bash vs zsh

5-3 간단 셸 스크립트(Shell script) 만들어보기

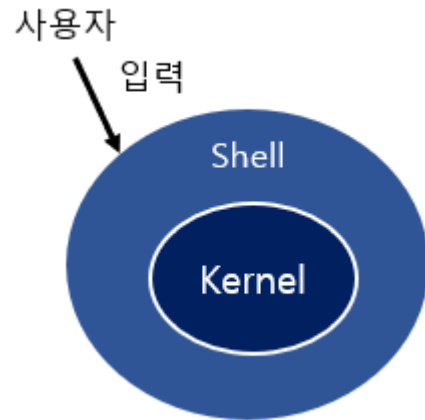
5-4 SUMMARY

5-5 Quiz 및 실습

5-6 도전과제

5-1 셸과 커널

셸은 우리가 입력한 명령을 커널이 이해할 수 있는 언어로 전달해 주는 역할을 한다.
커널은 이를 이해하여 하드웨어가 이 명령을 수행할 수 있도록 해 준다.



5-2 bash vs zsh

(가) 간단 실습

```
echo "hello"
```

```
echo $0
```

zsh # 설치가 안되어 있다면 다음줄을 실행하여 실행한다.

```
sudo apt-get install zsh
```

```
zsh
```

```
echo $0
```

[실행결과]

```
data@data:~$ echo "hello"
hello
data@data:~$ echo $0
bash
```

```

data@data:~$ sudo apt-get install zsh
[sudo] password for data:
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
The following additional packages will be installed:
  zsh-common
제안하는 패키지:
  zsh-doc
다음 새 패키지를 설치할 것입니다:
  zsh zsh-common
0개 업그레이드, 2개 새로 설치, 0개 제거 및 294개 업그레이드 안
함.
3,822 k바이트 아카이브를 받아야 합니다.
이 작업 후 14.2 m바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까? [Y/n] y
받기:1 http://kr.archive.ubuntu.com/ubuntu xenial/main amd64 z
sh-common all 5.1.1-1ubuntu2 [3,170 kB]
받기:2 http://kr.archive.ubuntu.com/ubuntu xenial/main amd64 z
sh amd64 5.1.1-1ubuntu2 [651 kB]
내려받기 3,822 k바이트, 소요시간 3초 (1,022 k바이트/초)
Selecting previously unselected package zsh-common.
data@data:~$ zsh

data%
data%
data%

```

(나) bash와 zsh의 차이

	bash	zsh
cd + [tab]	숨겨진 디렉터리 표시	디렉터리 표시
pwd	현재 작업 디렉터리	현재 작업 디렉터리
폴더 이동	cd /home/data	cd /h/d +[tab]

[실행결과] zsh

```
data% cd /home/data
data% pwd
/home/data
data% cd
data% cd
cd
cd-fix-profile      cd-it8
cd-create-profile   cd-iccdump   cdpath
```

[실습해보기]

bash 에 대해 실습하고 zsh과 비교해 보자.

5-3 간단 Shell script 입문해보기

여러가지 명령어를 한번에 실행시켜보기.

왜? 여러가지 반복적인 작업에 대해 매번 실행하기 보다 shell script를 만들어 자동화를 시키자.

로그에 대한 백업 파일을 만들어보자.

```
mkdir script      # 디렉터리 만들기
cd script/        # 스크립트 디렉터리로 이동하기
touch a.log b.log c.log # 빈 파일 만들기
mkdir bak         # bak 디렉터리 만들기
cp *.log bak      # 백업 파일 만들기
ls -l bak         # 복사되어 있는지 확인해 보기
```

[입력내용]

```
mkdir script
cd script/
touch a.log b.log c.log
mkdir bak
cp *.log bak
ls -l bak
```

==== 간단히 셸 스크립터 만들어보고 이런 것이 있다라고 알아보자.

```
$ ls /bin # 여러가지 프로그램들이 위치하고 있는 디렉터리 (mv, mkdir, rm...)
$ nano backup
```

(가) 아래 소스 코드를 입력한다.

```
#!/bin/bash # bin/bash 의 shell로 해석된다. (약속이다.)
DIRNAME="./bak"
if [ ! -d "$DIRNAME" ];then # 현재 디렉터리에 bak 디렉터리가 존재하는가?
    mkdir "$DIRNAME" # bak가 존재하지 않는다면 bak를 만든다.
fi # if문 종료
```

[소스 코드]

```
#!/bin/bash
DIRNAME="./bak"
if [ ! -d "$DIRNAME" ];then
    mkdir "$DIRNAME"
fi
```

CTRL + X => Enter를 키를 누른다.

(나) 셸 스크립트 실행

./backup # 권한 (permission)이 없다라고 나오면 chmod를 실행해야 함.

chmod +x backup # backup에 x에 권한을 추가해 준다.

```
ls -l
./backup # 실행시킨다.
```

[소스 코드]

```
./backup
chmod +x backup
ls -l
./backup
```

=> 실행결과 : 디렉터리가 생기고 a.log, b.log, c.log가 복사된다.

5-4 SUMMARY

(가) 셸이란? 커널이란?

셸은 우리가 입력한 명령을 커널이 이해할 수 있는 언어로 전달해 주는 역할을 한다.

커널은 이를 이해하여 하드웨어가 이 명령을 수행할 수 있도록 해 준다.

(나) bash, zsh

(다) 간단 셸 스크립트 만들어보기

nano shell1

```
-----  
#!/bin/bash  
touch a.log b.log c.log  
mkdir bak  
cp *.log bak  
ls -l bak  
-----  
chmod +x shell1  
./shell1
```

5-5 Quiz 및 실습

(가) 셸 스크립트는 bash로만 만들 수 있다. (O, X)

(나) 만들어진 셸 스크립트의 권한을 변경해 주는 명령어는 무엇인가?

5-6 도전과제

(가) 나만의 셸 스크립트를 만들어보자.

A. 디렉터리(bak) 있는지 체크하고, 디렉터를 만든다.

B. 해당 디렉터리로 이동한 후, 빈파일 세개 log1, log2, log3를 만든다.