



# 엔터프라이즈 서버관리

## 9. 쉘 프로그램

**SangJun, Im**

Department of Computer Science  
e\_mail : [imsangjun@gmail.com](mailto:imsangjun@gmail.com)



# 엔터프라이즈 서버관리

## 9. 쉘 프로그램

**SangJun, Im**

Department of Computer Science  
e\_mail : [imsangjun@gmail.com](mailto:imsangjun@gmail.com)

- 서버 구성
  - AWS 회원 가입 및 준비
- 실습 서버 설치
  - 레드햇 리눅스 서버 설치
  - 운영환경 기본 구성
  - 사용자 생성 (inhatc)
- 서버 모듈 설치 및 기본 명령 실습
  - 필수 모듈 설치
  - 공용서버 접근 확인
  - 기본명령어 실습
  - 고급 명령어 실습
- 서비스 구성
  - 웹서비스 구성
  - 이메일 서비스 구성
  - 보안 서비스 구성
  - DNS 서비스
  - Database
  - sftp

- 프로그래밍
  - 셸 프로그래밍
  - 백업 리커버리
  - 시스템 모니터링 프로그램
  - 개발 환경 구축 (C/C++ , JAVA, php)
- 서비스 분석
  - 서비스 관리
  - 시스템 서비스 모니터링
- 보안 및 튜닝
  - 네트워크 보안
  - 시스템 보안
  - 어플리케이션 보안



### ■ 필요사항

- 접근가능한 리눅스서버
- 웹 (기본 제공)
- 편집기 (VI 등)

### ■ 셸이란?

- 해석기
- 커널에 명령어를 전달하는 기능
- csh, tcsh, ksh, bash

### ■ 웹프로그램

- 웹 스크립트(명령)을 이용한 작성



- Shell Programming 환경
  - 리눅스 / 유닉스 셸 환경
  - 윈도우 : 커맨드 프롬프트 환경 (\*.bat)
- 파일 생성
  - vi 프로그램명.sh
- 권한 설정
  - chmod 이용하여 실행 권한 부여
- 실행방법
  - ./프로그램명.sh

컴파일 필요 없음



### ■ 쉘스크립트의 시작

□ `#!/bin/bash`



#! 다음에 오는 아규먼트를 실행프로그램으로 사용한다는 것을 알려주기 위해서 사용

첫번째 라인에 포함되어야 함

□ 예

```
#!/bin/bash
ftype=`file "$1"`
case "$ftype" in
    "$1: gzip compressed"*)
        echo "gzip 압축";;
    "$1: Zip archive"*)
        echo "Zip 압축";;
    *)
        echo "FLE $1 can not be uncompressed with smartzip";;
esac
```



## ■ 주석

```
# 프로그램 제목
```

```
Echo " hello world" # 문자열 출력
```

# 에서부터 라인 마지막까지 주석으로 인식

## ■ 변수

- ❑ 데이터 타입은 문자열(string)만을 가짐
- ❑ 변수선언의 필요가 없음
- ❑ 사용을 위해 변수앞에 \$ 붙임

```
#!/bin/bsh  
# 변수 할당  
a = " hello!!"
```

```
#변수 사용  
echo " A value : $a "
```



## ■ 기본 환경변수

\$0 → 실행된 셸스크립트 이름  
\$# → 스크립트에 넘겨진 인자의 개수  
\$\$ → 셸스크립트의 프로세스 ID

## ■ 인자변수

\$1~ \$nnn → 넘겨진 인자들  
\$\* → 스크립트에 전달된 인자들을 모아놓은 문자열. 하나의 변수에 저장되며  
IFS 환경변수의 첫번째 문자로 구분된다.  
\$@ → \$\*과 같다. 다만 구분자가 IFS변수의 영향을 받지 않는다.

## ■ 일반변수

문자열변수 = "문자열"

\$문자열변수





### ■ 주요명령어

- ❑ `echo "some text"` some text 를 화면에 출력한다
- ❑ `wc -l file` 파일의 라인수
- ❑ `cp sourcefile destfile` sourcefile 을 destfile 로 복사
- ❑ `mv oldname newname` 파일이름을 바꾸거나 파일의 이동
- ❑ `rm file` 파일 지우기
- ❑ `grep 'pattern' file` 파일에서 pattern의 문자열을 찾기
- ❑ `cut -b colnum file` 파일에서 문자열을 컬럼단위로 잘라서 보여줌
- ❑ `cat file.txt` file.txt 를 표준출력(stdout) 시킴
- ❑ `file somefile` somefile 의 파일타입 알아내기
- ❑ `read var` 입력값을 변수명var 에 대입
- ❑ `sort file.txt` file.txt 를 라인단위로 정렬
- ❑ `uniq` 파일에서 중복되는 문자열을 제거
- ❑ `tee` 표준출력되는 정보를 파일로 쓰기
- ❑ `basename file` 디렉토리명을 제외한 파일의 실제이름을 돌려줌
- ❑ `dirname file` 파일이름을 제외한 디렉토리의 이름을 돌려줌
- ❑ `head file` 파일의 처음 몇라인을 출력함
- ❑ `tail file` 파일의 마지막 몇라인을 출력함
- ❑ `sed` 정규표현에 의한 문자열의 검색및 치환에 사용됨



### ■ IF

```
#!/bin/bash
buffer = "hello"
if [ "$buffer" = "hello" ]; then
    echo expression evaluated as true
else
    echo expression evaluated as false
fi
```

### ■ while

```
#!/bin/bash
COUNTER=0
while [ $COUNTER -lt 10 ]; do
    echo The counter is $COUNTER
    let COUNTER=COUNTER+1
done
```



### ■ For

```
#!/bin/bash  
for i in $( ls ); do  
echo item: $i  
done
```

### ■ 산술관계자

```
-lt (<)  
-gt (>)  
-le (<=)  
-ge (>=)  
-eq (==)  
-ne (!=)
```



## 셸프로그램 예

vi argtest.sh

```
#!/bin/bash
echo "Total arguments : $#"  
echo "1st Argument = $1"  
echo "2nd argument = $2"
```

./argtest.sh

vi datetest.sh

```
Year=`date +%Y`  
Month=`date +%m`  
Day=`date +%d`  
Hour=`date +%H`  
Minute=`date +%M`  
Second=`date +%S`  
echo `date`  
echo "Current Date is: $Day-$Month-$Year"  
echo "Current Time is: $Hour:$Minute:$Second"  
./datetest.sh
```

vi functest.sh

```
#!/bin/bash  
function greeting() {  
  
    str="Hello, $name"  
    echo $str  
  
}  
  
echo "Enter your name"  
read name  
  
val=$(greeting)  
echo "Return value of the function is $val"  
  
./functest.sh
```



시 연



## 실습 과제 #1

- Shell script programming
- 강의에 제시된 프로그램 작성
  - argtest.sh
  - functest.sh
  - timetest.sh
  - + 알파 (옵션)
- 샘플 프로그램 작성 (개인서버 / 공영서버 )
  - 작성위치 : [개인 홈디렉토리]/work0522
  - 프로그램 코드 작성
  - 프로그램 명은 강의 내용 확인
- 제출
  - 서버에 작성
- 공용서버 주소 변경 없음



END

