

20240826

리눅스
셸 스크립트

Shell (셸)

- 셸은 사용자가 입력한 명령을 해석해서 커널로 전달하거나 처리결과를 사용자에게 전달하는 역할

ex) ServerB의 '텍스트 모드' 나 X 윈도의 '터미널' 처럼 명령을 입력하는 환경

환경 변수

- 셸은 여러가지 환경 변수값을 가지는데 설정된 환경 변수는 `echo $환경변수이름` 형식으로 명령을 실행하면 확인 가능
예) 호스트이름을 출력하려면 `echo $HOSTNAME` (대문자로 입력)
P.426 참조

셸 스크립트 작성

```
guru@master: ~/workshe
guru@master:~$ ls -l
합계 36
drwxrwxr-x 2 guru guru 4096 8월 26 17:06 workshell
drwxr-xr-x 2 guru guru 4096 8월 21 12:42 공개
drwxr-xr-x 2 guru guru 4096 8월 21 12:42 다운로드
drwxr-xr-x 2 guru guru 4096 8월 21 12:42 문서
drwxr-xr-x 2 guru guru 4096 8월 21 12:42 바탕화면
drwxr-xr-x 2 guru guru 4096 8월 21 12:42 비디오
drwxr-xr-x 2 guru guru 4096 8월 21 12:42 사진
drwxr-xr-x 2 guru guru 4096 8월 21 12:42 음악
drwxr-xr-x 2 guru guru 4096 8월 21 12:42 템플릿
guru@master:~$ cd workshell
guru@master:~/workshell$ ls -l
합계 60
-rwxr-xr-x 1 guru guru 196 8월 26 16:54 andor.sh
-rwxr-xr-x 1 guru guru 154 8월 26 16:26 case1.sh
-rwxr-xr-x 1 guru guru 288 8월 26 16:38 case2.sh
-rwxr-xr-x 1 guru guru 130 8월 26 17:01 forin1.sh
-rwxr-xr-x 1 guru guru 92 8월 26 17:06 forin2.sh
-rwxrwxr-x 1 guru guru 58 8월 26 16:06 if1.sh
-rwxrwxr-x 1 guru guru 78 8월 26 16:10 if2.sh
-rwxrwxr-x 1 guru guru 114 8월 26 16:13 if3.sh
-rw-r--r-- 1 guru guru 156 8월 26 16:18 if4.sh
-rwxr-xr-x 1 guru guru 130 8월 26 16:22 if5.sh
-rwxr-xr-x 1 guru guru 79 8월 26 15:18 name.sh
-rw-rw-r-- 1 guru guru 139 8월 26 15:54 numcalc.sh
-rw-rw-r-- 1 guru guru 178 8월 26 16:04 paravar.sh
-rwxr-xr-x 1 guru guru 15 8월 26 15:07 rm.sh
-rw-rw-r-- 1 guru guru 140 8월 26 15:36 var1.sh
```

workshell 생성후

vi name.sh 로 셸스크립트 생성

```
#!/bin/sh
echo "사용자 이름:" $USER
echo "홈 디렉터리:" $HOME
exit 0
~
~
~
~
```

name.sh에
위 화면의 텍스트를 입력

chmod 755 name.sh 실행후
./name.sh 실행

```
-rw-rw-r-- 1 guru guru 140 8월 26 15:36 var1.sh
guru@master:~/workshell$ vi name.sh
guru@master:~/workshell$ vi name.sh
guru@master:~/workshell$ ./name.sh
사용자 이름: guru
홈 디렉터리: /home/guru
guru@master:~/workshell$
```

셸 스크립트

변수의 기본

-> 변수에 넣는 모든값은 문자열(String)로 취급한다.

숫자를 넣어도 문자로 취급

-> 변수 이름은 대소문자를 구분한다

-> 변수를 대입할때 '='좌우에는 공백이 없어야 한다.

변수의 입력과 출력

-> \$라는 문자가 들어간 글자를 출력하려면 "로 묶거나 앞에 \를 붙여야 한다.

또한 ""로 변수를 묶어도되고, 묶지않아도된다.

셸 스크립트 예제

vi var1.sh 생성 후 텍스트 입력

```
#!/bin/sh
myvar="Hi world!"
echo $myvar
echo "$myvar"
echo '$myvar'
echo \ $myvar
echo 값 입력:
read myvar
echo '$myvar' = $myvar
exit 0
~
~
~
~
```

./var1.sh 실행 후 값 입력

```
guru@master:~/workshell$ vi var1.sh
guru@master:~/workshell$ ./var1.sh
Hi world!
Hi world!
$myvar
$myvar
값 입력:
```

값 입력 후

```
guru@master:~/workshell$ vi var1.sh
guru@master:~/workshell$ ./var1.sh
Hi world!
Hi world!
$myvar
$myvar
값 입력:
Hello world
$myvar = Hello world
guru@master:~/workshell$
```

숫자 계산

변수에 넣은 값은 모두 문자열로 취급하기 때문에
만약 변수에 들어있는 값에 연산을 하려면 expr 키워드와 역따옴표 `` 로 묶어줘야 한다.

수식에 괄호를 사용하려면 그 앞에 꼭 역슬래시\를 붙여야 한다.
곱하기 기호(*) 은 (*) 역슬래시를 붙여서 사용한다.

셸 스크립트 예제

vi numcalc.sh 생성 후 텍스트 입력

```
#!/bin/sh
myvar="Hi world!"
echo $myvar
echo "$myvar"
echo '$myvar'
echo \ $myvar
echo 값 입력:
read myvar
echo '$myvar' = $myvar
exit 0
~
~
~
~
```

실행 후

```
-rwxr-xr-x 1 guru guru 140  8월 26 18:37 var1.sh
guru@master:~/workshell$ chmod 755 numcalc.sh
guru@master:~/workshell$ ./numcalc.sh
100+200
300
60
guru@master:~/workshell$
```


if문 예제

- vi if1.sh 생성

```
#!/bin/sh
if [ "100" = "100" ]
then
    echo "true"
fi
exit 0
~
~
~
~
~
~
~
```

기본 if문
if [조건]
then
 참일 경우실행
fi
exit 0

주의
[조건] 사이의 각 단어에는 공백이 필요.

```
60
guru@master:~/workshell$ vi if1.sh
guru@master:~/workshell$ ./if1.sh
true
./if1.sh: 6: exit0: not found
guru@master:~/workshell$
```

결과

if문 예제2

- vi if2.sh 생성

```
#!/bin/sh
if [ "100" = "110" ]
then
    echo "true"
else
    echo "false"
fi
exit 0
```

- 기본 if문
- if [조건]
- then
- 참일 경우실행
- else
- 거짓인 경우 실행
- fi
- exit 0

```
guru@master:~/workshell$ vi if2.sh
guru@master:~/workshell$ ./if2.sh
false
guru@master:~/workshell$
```

결과

조건문에 들어가는 비교 연산자

- 수식 1 -eq 수식2 (두 수식이 같으면 참)
- 수식 1 -ne 수식2 (두 수식이 같지않으면 참)
- 수식 1 -gt 수식2 (수식1이 크면 참)
- 수식 1 -lt 수식2 (수식1이 작으면 참)
- 수식 1 -le 수식2 (수식1이 작거나 같으면 참)
- !수식 (수식이 거짓이면 참)
- -n 문자열 문자열이 NULL 이 아니면 참
- -z 문자열 문자열이 NULL 이면 참

if문 예제3

- vi if3.sh 생성

```
#!/bin/sh
if [ 100 -eq 200 ]
then
    echo "100과 200은 같 다"
else
    echo "100과 200은 다르다"
fi

exit 0
```

~
~
~

```
guru@master:~/workshell$ vi if3.sh
guru@master:~/workshell$ ./if3.sh
100과 200은 다르다
guru@master:~/workshell$
```

결과

파일과 관련된 조건

- -d 파일이름 : 파일이 디렉터리면 참
- -e 파일이름 : 파일이 존재하면 참
- -f 파일이름 : 파일이 일반 파일이면 참
- -g 파일이름 : 파일에 set-group-id가 설정되면 참
- -r 파일이름 : 파일이 읽기 가능이면 참
- -s 파일이름: 파일크기가 0이 아니면 참
- -u 파일이름: 파일에 set-user-id가 설정되면 참
- -w 파일이름: 파일이 쓰기 가능 상태면 참
- -x 파일이름 : 파일이 실행가능 상태면 참

if문 예제4

- vi if4.sh 생성

```
#!/bin/sh
fname=/lib/systemd/system/cron.service
if [ -f $fname ]
then
    head -5 $fname
else
    echo "cron 서버가 설치되지 않았습니다."
fi
exit 0
```

```
guru@master:~/workshell$ ./if4.sh
[Unit]
Description=Regular background program processing daemon
Documentation=man:cron(8)
After=remote-fs.target nss-user-lookup.target
```

결과

```
guru@master:~/workshell$
```

case문 예제

```
#!/bin/sh
echo "리눅스가 재미있나요? (yes / no)"
read answer
case $answer in
    yes | y | Y | yes | YES)
        echo "다행 입니다"
        echo "더욱 열심히 하세요 ";;
    [nN]*)
        echo "안타깝네요 ";;
    *)
        echo "yes 아니면 no 만 입력하세요!"
        exit 1;;
esac
exit 0
```

yes의 y Y yes YES 어떤걸 입력해도 출력

n, N * 예) Ne 로 입력해도 출력

```
guru@master:~/workshell$ vi case2.sh
guru@master:~/workshell$ ./case2.sh
리눅스가 재미있나요? (yes / no)
yes
다행 입니다
더욱 열심히 하세요
guru@master:~/workshell$ ./case2.sh
리눅스가 재미있나요? (yes / no)
no
안타깝네요
guru@master:~/workshell$ ./case2.sh
리눅스가 재미있나요? (yes / no)
fdas
yes 아니면 no 만 입력하세요!
guru@master:~/workshell$
```

결과

and,or문 예제

```
#!/bin/sh
echo "보고싶은 파일명을 입력하세요"
read fname
if [ -f $fname ] && [ -s $fname ] ;
then
    head -5 $fname
else
    echo "파일이 없거나, 크기가 0입니다."
fi
exit 0
```

yes 아니면 no 만 입력하세요!

```
guru@master:~/workshell$ vi andor.sh
```

```
guru@master:~/workshell$ ./andor.sh
```

보고싶은 파일명을 입력하세요

/etc/fstab

/etc/fstab: static file system information. # # Use 'blkid' to print the universally unique identifier for a # device; this may be used with UUID= as a more robust way to name devices

that works even if disks are added and removed. See fstab(5).

#

<file system> <mount point> <type> <options> <dump> <pass>

/ was on /dev/sda5 during installation

```
guru@master:~/workshell$
```

결과

반복문 예제

```
#1/bin/sh
hap=0
for i in 1 2 3 4 5 6 7 8 9 10
do
    sum=`expr $sum + $i`
done
echo "1부터 10까지의 합: $sum 입니다"
exit 0
~
~
~
~
~
~
~
```

```
guru@master:~/workshell$ vi forin1.sh
guru@master:~/workshell$ ./forin1.sh
1부터 10까지의 합: 55 입니다
guru@master:~/workshell$
```

결과

```
for 변수 in 값1 값2 값3 ....
do
    반복할 문장
done
```

반복문 예제2

현재 디렉터리에 있는 셸 스크립트 파일(*.sh)의
파일 이름과 앞에 3줄을 출력하는 프로그램

```
#!/bin/sh
for fname in $(ls *.sh)
do
    echo "-----$fname-----"
    head -3 $fname
done
exit 0
```

~
~
~
~
~
~

```
guru@master:~/workshell$ ./forin2.sh
-----andor.sh-----
#!/bin/sh
echo "보고싶은 파일명을 입력하세요"
read fname
-----case1.sh-----
#!/bin/sh
case "$1" in
    start)
-----case2.sh-----
#!/bin/sh
echo "리눅스가 재미있나요? (yes / no)"
read answer
-----forin1.sh-----
#!/bin/sh
hap=0
for i in 1 2 3 4 5 6 7 8 9 10
-----forin2.sh-----
#!/bin/sh
for fname in $(ls *.sh)
do
    -----if1.sh-----
    #!/bin/sh
    if [ "100" = "100" ]
    then
        -----if2.sh-----
        #!/bin/sh
        if [ "100" = "110" ]
        then
            -----if3.sh-----
            #!/bin/sh
            if [ 100 -eq 200 ]
            then
```

결과