

SHELL SCRIPT

System Programming Basic Exercise's lecture note 5



셸 스크립트

- 스크립트 란?
 - 인터프리터가 실행하는 코드
 - Perl, Java Script, Shell Script, etc.
- 셸스크립트
 - 리눅스 시스템과 사용자 간의 소통할 수 있는 수단
 - 윈도우의 command와 유사
 - 유닉스로부터 왔으며, 리눅스의 기본은 bash
 - 파일로 저장시 확장자는 sh



실행 방법

- 셸에 직접 입력

```
spbe2013@ubuntu:~/Documents$ echo "Hello, world"  
Hello, world  
spbe2013@ubuntu:~/Documents$
```

- 파일로 저장하여 실행

```
spbe2013@ubuntu:~/Documents$ cat helloworld.sh  
#!/bin/bash  
  
echo "Hello, world"  
spbe2013@ubuntu:~/Documents$ ./helloworld.sh  
Hello, world  
spbe2013@ubuntu:~/Documents$
```



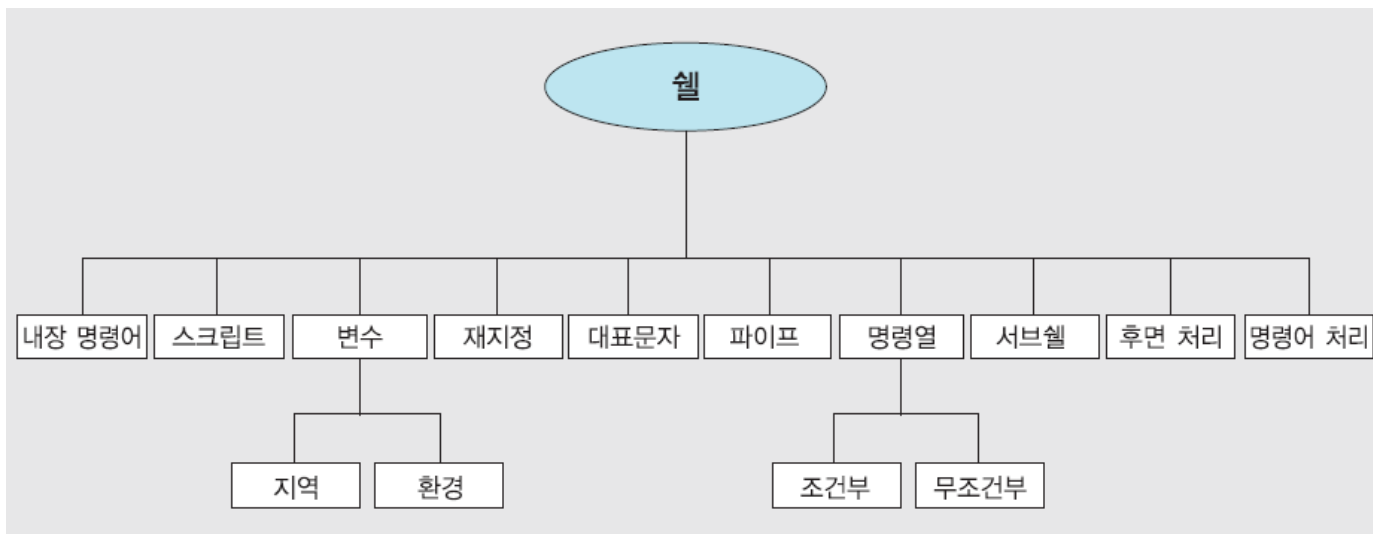
Shell의 종류

- 본 쉘(Bourne shell)
 - 벨연구소의 스티븐 본(Stephen Bourne)에 의해 개발됨
 - 유닉스에서 기본 쉘로 사용됨
- 콘 쉘(Korn shell)
 - 1980년대에는 역시 벨연구소에서 본 쉘을 확장해서 만듦.
- Bash(Bourne again shell)
 - GNU에서 본 쉘을 확장하여 개발한 쉘
 - 리눅스 및 맥 OS X에서 기본 쉘로 사용되면서 널리 보급됨
 - Bash 명령어의 구문은 본 쉘 명령어 구문을 확장함
- C 쉘(C shell)
 - 버클리대학의 빌 조이(Bill Joy)
 - 쉘의 핵심 기능 위에 C 언어의 특징을 많이 포함함
 - BSD 계열의 유닉스에서 많이 사용됨
 - 최근에 이를 개선한 tcsh이 개발되어 되어 사용됨



Shell의 기본 기능

- 명령어 처리
 - 사용자가 입력한 명령을 해석하고 적절한 프로그램을 실행
- 시작 파일
 - 로그인할 때 실행되어 사용자 별로 맞춤형 사용 환경 설정
- 스크립트
 - 셸 자체 내의 프로그래밍 기능



Start-up 파일

- 셸 마다 시작될 때 자동으로 실행되는 고유의 시작 파일
- 주로 사용자 환경을 설정하는 역할을 하며, 환경설정을 위해서 환경변수에 적절한 값을 설정한다.

-]\$ 환경변수명=문자열

```
]$ TERM=xterm
```

```
]$ echo $TERM
```

```
xterm
```

- export 명령어
\$ export TERM
- 환경 변수 확인
\$ env
HOSTNAME=CS2
HOST=CS2
TERM=xterm
SHELL=/bin/sh
GROUP=faculty
USER=chang
...



시작 파일(start-up file)

- 본 셸

`/etc/profile`

`~/.profile`

- bash

`/etc/profile`

`/etc/bashrc`

`~/.bash_profile`

`~/.bashrc`

- C 셸

`/etc/.login`

`~/.login`

`~/.cshrc`



Shell 명령어 종류

- 내장 명령어

- 셸 내에 내장되어 있는 명령어

```
]$ echo -n 명령줄인수*
```

```
]$ cd 디렉터리
```

- 유틸리티 프로그램

- 명령어를 위한 실행 파일이 별도로 존재하는 명령어

```
]$ ls /bin/ls
```



입출력 재지정 및 파이프

- 출력 재지정
\$ 명령어 > 파일
- 출력 추가
\$ 명령어 >> 파일
- 입력 재지정
\$ 명령어 < 파일
- 문서 내 입력
\$ 명령어 << 단어
...
단어
- 파이프
\$ 명령어1 | 명령어2



복합 명령어

- 명령어 열 (command sequence)
\$ 명령어1; ... ; 명령어_n
\$ date; who; pwd
- 명령어 그룹 (command group)
\$ (명령어1; ... ; 명령어_n)
\$ date; who; pwd > out1.txt
\$ (date; who; pwd) > out2.txt



조건 명령어

- \$ 명령어1 && 명령어2

```
]$ gcc myprog.c && ./a.out
```

- \$ 명령어1 || 명령어2

```
]$ gcc myprog.c || echo "컴파일 실패"
```



파일 이름 대치

- 대표문자를 이용한 파일 이름 대치
 - 대표문자를 이용하여 한 번에 여러 파일들을 나타냄
 - 명령어 실행 전에 대표문자가 나타내는 파일 이름들로 먼저 대치하고 실행

```
]$ gcc *.c
```

```
]$ gcc a.c b.c test.c
```

- 대표문자
 - * 빈 스트링을 포함하여 임의의 스트링을 나타냄
 - ? 임의의 한 문자를 나타냄
 - [...] 대괄호 사이의 문자 중 하나를 나타내며 부분범위 사용 가능
- ```
]$ ls *.txt
```
- ```
]$ ls [ac]*
```



명령어 대치(command substitution)

- 명령어를 실행할 때 다른 명령어의 실행 결과를 이용
 - 역따옴표(`)로 에워싼 다른 명령어 부분은 그 명령어의 실행 결과로 대치된 후에 실행

```
]$ echo 현재 시간은 `date`
```

```
]$ echo 현재 디렉터리 내의 파일의 개수 : `ls | wc -w`
```

```
현재 디렉터리 내의 파일의 개수 : 32
```



따옴표 사용

- 따옴표를 이용하여 대치 기능을 제한

```
$ echo 3 * 4 = 12
```

```
3 cat.csh count.csh grade.csh invite.csh menu.csh test.sh =  
12
```

```
$ echo "3 * 4 = 12"
```

```
3 * 4 = 12
```

```
$ echo '3 * 4 = 12'
```

```
3 * 4 = 12
```

```
$ name=나가수
```

```
$ echo '내 이름은 $name 현재 시간은 `date`'
```

```
내 이름은 $name 현재 시간은 `date`
```

```
$ echo "내 이름은 $name 현재 시간은 `date`"
```

```
내 이름은 나가수 현재 시간은 2011년 12월 8일 목요일 오후 03시 43분  
12초
```



따옴표 사용

- 정리

1. 작은따옴표(')는 파일이름 대치, 변수 대치, 명령어 대치를 모두 제한한다.
2. 큰따옴표(")는 파일이름 대치만 제한한다.
3. 따옴표가 중첩되면 밖에 따옴표가 효력을 갖는다.



셸 스크립트 파일 작성

- 셸 스크립트 종류

1. 첫 번째 줄에 사용할 셸을 `#!`경로명 형태로 지정

```
#!/bin/csh
```

```
#!/bin/ksh
```

```
#!/bin/bash
```

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

2. 단지 첫 번째 줄이 `#`으로 시작되면 c 셸 스크립트로 간주
3. 그 외는 본 셸 스크립트로 간주한다.



셸 스크립트 작성 및 실행

- 에디터를 사용하여 스크립트 파일을 작성한다.
- chmod를 이용하여 실행 모드로 변경한다.
- 스크립트 이름을 타입핑하여 실행한다.

```
spbe2013@ubuntu:~$ cd #
spbe2013@ubuntu:~$ cd Documents/
spbe2013@ubuntu:~/Documents$ vi shell_ex1.sh
spbe2013@ubuntu:~/Documents$ chmod 755 shell_ex1.sh
spbe2013@ubuntu:~/Documents$ ./shell_ex1.sh
Date :
Wed Sep 11 04:38:27 PDT 2013
user :
spbe2013 tty7          2013-09-11 04:06 (:0)
spbe2013 pts/0         2013-09-11 04:10 (:0)
system status :
  04:38:27 up 32 min,  2 users,  load average: 0.14, 0.07, 0.06
spbe2013@ubuntu:~/Documents$
```



실습 : shell_ex1.sh

- 파일 내용

```
#!/bin/bash

echo Date :
date

echo user :
who

echo system status :
uptime
```

- 결과

```
spbe2013@ubuntu:~/Documents$ ./shell_ex1.sh
Date :
Wed Sep 11 04:44:32 PDT 2013
user :
spbe2013 tty7          2013-09-11 04:06 (:0)
spbe2013 pts/0         2013-09-11 04:10 (:0)
system status :
 04:44:32 up 38 min,  2 users,  load average: 0.00, 0.03, 0.05
spbe2013@ubuntu:~/Documents$
```



실습 : shell_ex2.sh

- 파일 내용

```
#!/bin/bash  
  
echo "Start"  
sleep 2  
echo "Finish"
```

- 결과

```
spbe2013@ubuntu:~/Documents$ ./shell_ex2.sh  
Start  
Finish  
spbe2013@ubuntu:~/Documents$
```



실습 : shell_ex3.sh

- 파일 내용

```
#!/bin/bash

test=/

echo -1-----
ls /
echo -2-----
ls $test
echo -3-----
ls '$test'
echo -4-----
ls "$test"
echo -5-----
echo ls $test
echo -6-----
exec ls /
```



실습 : shell_ex3.sh cont'd

- 결과

```
spbe2013@ubuntu:~/Documents$ ./shell_ex3.sh
-1-----
bin    dev    initrd.img    lost+found    opt    run    srv    usr
boot   etc    initrd.img.old media          proc   sbin    sys    var
cdrom  home   lib           mnt           root   selinux tmp    vmlinuz
-2-----
bin    dev    initrd.img    lost+found    opt    run    srv    usr
boot   etc    initrd.img.old media          proc   sbin    sys    var
cdrom  home   lib           mnt           root   selinux tmp    vmlinuz
-3-----
ls: cannot access $test: No such file or directory
-4-----
bin    dev    initrd.img    lost+found    opt    run    srv    usr
boot   etc    initrd.img.old media          proc   sbin    sys    var
cdrom  home   lib           mnt           root   selinux tmp    vmlinuz
-5-----
ls /
-6-----
bin    dev    initrd.img    lost+found    opt    run    srv    usr
boot   etc    initrd.img.old media          proc   sbin    sys    var
cdrom  home   lib           mnt           root   selinux tmp    vmlinuz
spbe2013@ubuntu:~/Documents$
```



실습 : shell_ex4.sh

- 파일 내용

```
#!/bin/bash

echo ---- Redirect example ----
echo "(1)"
echo "a" > test
cat test
echo "(2)"
echo "b" > test
cat test
echo "(3)"
echo "c" >> test
cat test

echo ---- command sequence ----
date; who; pwd
```

- 결과

```
spbe2013@ubuntu:~/Documents$ ./shell_ex4.sh
---- Redirect example ----
(1)
a
(2)
b
(3)
b
c
---- command sequence ----
Wed Sep 11 05:11:34 PDT 2013
spbe2013 tty7          2013-09-11 04:06 (:0)
spbe2013 pts/0         2013-09-11 04:10 (:0)
/home/spbe2013/Documents
spbe2013@ubuntu:~/Documents$
```



실습 : shell_ex5.sh

- 자신의 홈 디렉토리를 압축하여 백업하는 쉘스크립트를 작성하라.
 - Backup 파일 이름 : 사용자이름_현재날짜_현재시간.tar.gz
 - ex) cse2018_20151007_153030.tar.gz
 - 사용자 이름은 현재 로그인된 계정을 이용한다.
 - \$whoami
 - 날짜와 시간 출력 (날짜_시분초)
 - \$date +%Y%m%d_%H%M%S

