

Bash Tutorial

조영일

2011. 05 ~

목차

- 커서 이동
- 기본 명령
- 변수
- 제어문
- 함수
- Script File
- 입출력 리다이렉트
- 필터
- Job Control
- 설정파일

소개

- Unix Shell
 - Kernel을 둘러싸고 있는 껍데기
 - 사용자가 제어할 수 있는 명령행 해석기
 - MS 윈도우즈로 치면 DOS 환경 또는 커맨드 창

소개

- Bourne Shell
 - 1977, Stephen Bourne
 - /bin/sh
- BASH (Bourne Again Shell)
 - 1989, Brian Fox
 - /bin/bash

```
[root@localhost ~]# ping -q fa.wikipedia.org
PING text.pmtpa.wikimedia.org (208.80.152.2) 56(84) bytes of data.
```

```
^C
```

```
--- text.pmtpa.wikimedia.org ping statistics ---
```

```
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 540.528/540.528/540.528/0.000 ms
```

```
[root@localhost ~]# pwd
```

```
/root
```

```
[root@localhost ~]# cd /var
```

```
[root@localhost var]# ls -la
```

```
total 72
drwxr-xr-x. 18 root root 4096 Jul 30 22:43 .
drwxr-xr-x. 23 root root 4096 Sep 14 20:42 ..
drwxr-xr-x.  2 root root 4096 May 14 00:15 account
drwxr-xr-x. 11 root root 4096 Jul 31 22:26 cache
drwxr-xr-x.  3 root root 4096 May 18 16:03 db
drwxr-xr-x.  3 root root 4096 May 18 16:03 empty
drwxr-xr-x.  2 root root 4096 May 18 16:03 games
drwxrwx--T.  2 root gdm  4096 Jun  2 18:39 gdm
drwxr-xr-x. 38 root root 4096 May 18 16:03 lib
drwxr-xr-x.  2 root root 4096 May 18 16:03 local
lrwxrwxrwx.  1 root root    11 May 14 00:12 lock -> ../run/lock
drwxr-xr-x. 14 root root 4096 Sep 14 20:42 log
lrwxrwxrwx.  1 root root    10 Jul 30 22:43 mail -> spool/mail
drwxr-xr-x.  2 root root 4096 May 18 16:03 nis
drwxr-xr-x.  2 root root 4096 May 18 16:03 opt
drwxr-xr-x.  2 root root 4096 May 18 16:03 preserve
drwxr-xr-x.  2 root root 4096 Jul  1 22:11 report
lrwxrwxrwx.  1 root root     6 May 14 00:12 run -> ../run
drwxr-xr-x. 14 root root 4096 May 18 16:03 spool
drwxrwxrwt.  4 root root 4096 Sep 12 23:50 tmp
drwxr-xr-x.  2 root root 4096 May 18 16:03 yp
```

```
[root@localhost var]# yum search wiki
```

```
Loaded plugins: langpacks, presto, refresh-packagekit, remove-with-leaves
```

rpmfusion-free-updates		2.7 kB	00:00
rpmfusion-free-updates/primary_db		206 kB	00:04
rpmfusion-nonfree-updates		2.7 kB	00:00
updates/metalink		5.9 kB	00:00
updates		4.7 kB	00:00
updates/primary_db	73% [=====	62 kB/s	2.6 MB 00:15 ETA

커서 이동

- 문장의 맨 처음 / 끝
 - Ctrl-a / Ctrl-e
- 단어 단위 이동
 - Ctrl-*방향키*

화면 제어

- 화면 정리
 - Ctrl-l
- 스크롤 중단 및 재개
 - Ctrl-s / Ctrl-q

서버 접속 및 종료

- 접속
 - ssh hcon.nhnent.com
 - rlogin -i 아이디 호스트명
- 종료
 - logout / exit
- 조회
 - last

기본 명령

- pwd
 - 현재 디렉토리를 확인하는 명령

기본 명령

- id
 - 자신의 로그인 아이디를 확인하는 명령

기본 명령

- `uname`
 - 현재 접속해있는 서버의 이름/OS 정보를 확인하는 명령
 - `uname`
 - `uname -a`
- 비슷한 명령
 - `hostname`

기본 명령

- echo
 - 문자열을 출력하는 명령
 - echo "hello world"
 - echo -n "hello world"; echo "java coffee"

기본 명령

- ls
 - 디렉토리와 파일의 목록을 출력하는 명령
 - 윈도우즈 cmd의 dir
- ls -l
- ls -alF
- ls /
- ls /home1/irteam

Globbering & Wildcard

- Glob pattern
 - 임의의 길이를 가지는 문자열
*
 - 임의의 한 글자
?
 - 글자 집합
[abcw-z]

Globbering & Wildcard

- `ls *`
- `ls .*`
- `ls [a-f]*.txt`
- `ls *.tx?`

파일 시스템

- File

- OS에서 데이터를 디스크와 같은 저장소에 저장해 둘 때의 단위 형태
- 넓은 의미에서는 일반 파일, 디렉토리와 각종 입출력 장치를 모두 파일로 간주

- Directory

- 파일을 담아두는 공간 (폴더)
- 디렉토리 안에 디렉토리를 담을 수 있기 때문에 트리 구조로 형상화

파일 시스템

- 디렉토리 이동
 - `cd /var/log`
- 사용자 홈 디렉토리
 - `cd ~/.vim`
- 특정 사용자의 홈 디렉토리
 - `ls ~nemon`

History

- 조회
 - history
- 12번 명령 재실행
 - !12
- 커맨드 prefix로 재실행
 - !find

History

- 바로 직전 명령 재실행
 - !!
- 바로 직전 명령의 마지막 argument
 - !\$

History

- 직전 명령 편집
 - echo "hello world"
 - ^hello^java

파일 다루는 기본 명령

- touch
 - 빈 파일 만들기
 - touch *파일*
- touch file1.txt

파일 다루는 기본 명령

- 디렉토리 관련 명령
 - mkdir *디렉토리*
 - rmdir *디렉토리*
 - 비어있는 디렉토리만 삭제

파일 다루는 기본 명령

- cp

- cp test1.txt test2.txt
- cp -r mydir1 mydir2

- mv

- mv test1.txt test2.txt
- mv mydir1 mydir2

- 대상파일이 이미 존재하는 디렉토리라면?

파일 다루는 기본 명령

- rm
 - rm -f test2.txt
 - rm -rf mydir2

파일 다루는 기본 명령

- file
 - 파일의 종류를 판단하는 명령
 - file *파일*

 - file /bin/gzip
 - file /etc/rpm/macros.jpckage

변수

- 사용자 변수
 - 보통 소문자로 명명
 - $x=1$
 - $x=\text{"hello world"}$
 - $x=\text{hello world}$

변수

- 값을 꺼내쓰기
 - echo x
 - echo \$x
 - echo \${x}
 - echo "\$x"

변수

- Full Quotation
 - single quote as literal
 - `x="java"; echo '$x is noun'`
- Partial Quotation
 - double quote as variable substitution
 - `x="java"; echo "$x is noun"`

Evaluation

- 명령 실행 결과를 문자열 변수로 대입
 - backquote(`) 또는 \$()로 둘러싸기
 - `date_str=`date +%Y%m%d```
 - `date_str=$(date +%Y%m%d)`

Evaluation

- 명령 문자열을 실행하기
 - eval *명령문자열*
 - cmd="ls -alF"
 - eval \$cmd

Evaluation

- Arithmetic
 - `a=13`
 - `b=$((a % 3))`
 - `i=$((2 ** 60))`
 - `echo $RANDOM`

Type

- read-only
 - declare `-r var1=1`
 - `var1=2`
- integer
 - declare `-i n=3`
 - `n="hello"`
 - `echo $n`

Type

- array
 - declare `-a arr=(one two three four five)`
 - `arr=3`
 - `echo ${arr[3]}`
- function
 - declare `-f fn1`
 - `function fn1() { echo $2; }`
 - `fn1 hello world`

환경 변수

- 환경(environment)
 - 프로세스를 둘러싼 주변 정보
 - env
- exporting
 - sub-shell에서 상속받아 사용할 수 있도록 변수를 공개
 - 사용자 변수를 환경 변수로 전환
 - export나 declare -x로 지정

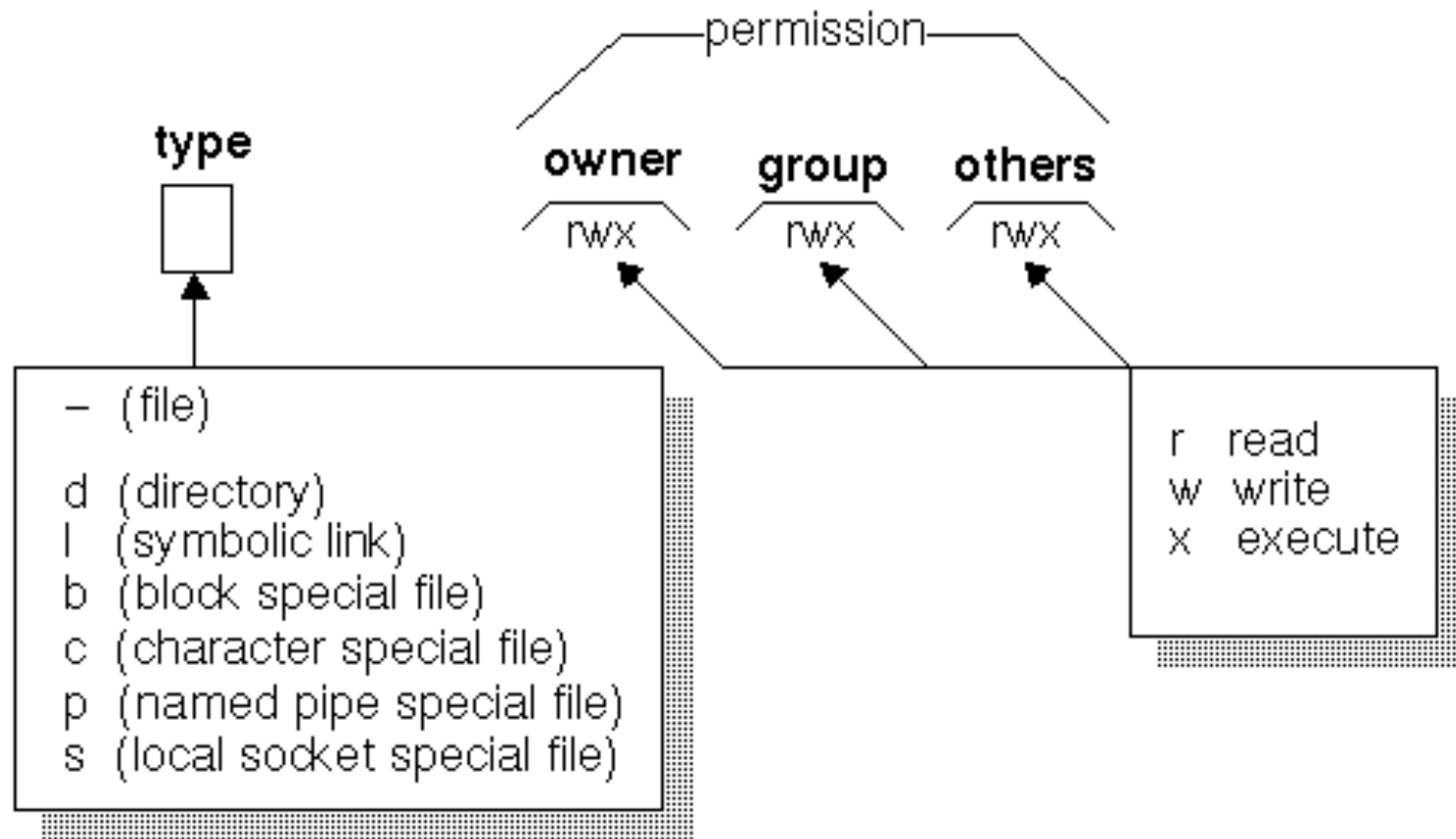
환경 변수

- var1=2
 - bash
 - echo \$var1
 - exit
- export var1=2
 - bash
 - echo \$var1
 - exit

File Permission

```
[tethys]:[8:39am]:[/home/rnejdl/DOCS] > ls -al
total 1825
drwxr-xr-x  2 rnejdl  rnejdl    512 Sep 16 10:27 .
drwxr-xr-x 23 rnejdl  rnejdl   4096 Sep 17 21:09 ..
-rw-----  1 rnejdl  sarah   208896 May  3 23:09 DSLProvo.doc
-rwxr--r--  1 rnejdl  rnejdl  307543 Sep 11 17:37 IPAddressing-Overview.pdf
-rwxr--r--  1 rnejdl  rnejdl  412989 Aug 28 18:08 Routers-DSL-Flowpoint.pdf
-rwxr--r--  1 rnejdl  rnejdl   76810 Sep  6 11:42 SWOT analysis.vsd
-rwxr--r--  1 rnejdl  rnejdl   20480 Sep  6 11:41 Social Styles.doc
-rwxr--r--  1 rnejdl  rnejdl   25600 Sep 15 17:43 Acceptable Use Policy.doc
-rwxr--r--  1 rnejdl  rnejdl   28160 Sep  6 11:42 bibliography leadership.doc
-rw-----  1 rnejdl  rnejdl  705536 Sep 12 23:45 provo-final.doc
-rwxr--r--  1 rnejdl  rnejdl   23552 Sep  6 11:41 question types.doc
[tethys]:[8:39am]:[/home/rnejdl/DOCS] >
```

File Permission



File Permission

Octal	String	Owner			Group			Other		
		read	write	execute	read	write	execute	read	write	execute
777	rwX rwx rwx	0	0	0	0	0	0	0	0	0
755	rwX r-X r-X	0	0	0	0		0	0		0
744	rwX r-- r--	0	0	0	0			0		
644	rw- r-- r--	0	0		0			0		
321	-wX -w- --X		0	0		0				0

File Permission

- `vi test1.sh`
- `ls -al !$`
- `chmod 0644 !$`
- `ls -al !$`

- `chmod u+x test1.sh`

File Permission

- umask
 - 파일에 부여되는 permission의 초기값에서 제외할 권한 지정
 - umask 0027
 - touch test2.txt
- 소유자 변경하기
 - chown *소유자아이디 파일*

제어문

- Boolean expression
 - `&&`, `||`, `!`, `-a`, `-0`
 - `((0 && 1)) && echo true || echo false`
 - `((0 || 1)) && echo true || echo false`

제어문

- Comparison operator
 - integer
 - -eq -ne -gt -ge -lt -le
 - string
 - < <= > >= == = != -z -n
 - **zero length? null?**

제어문

- File Test

- -e

- exist?

- -s

- size not zero?

- -f / -d / -p / -L

- file, directory, pipe, symlink?

- -r / -w / -x / -u / -g / -k

- permission? (setuid, setgid, sticky)

- -nt / -ot

- newer than, older than?

제어문

- if

- if [조건]; then
 - 실행문
- fi

- x=3
- if [\$x -eq 3]; then
 - echo "x is 3"
- fi

제어문

- 한 줄 쓰기
 - `x=3; if [$x -eq 3]; then echo "x is 3"; fi`

제어문

- 주의사항
 - -eq 연산자는 정수 연산자
 - ==, = 연산자는 문자열 연산자
 - 변수가 정의되지 않았거나 공백을 포함할 수 있으므로 quotation 필요
 - x="he llo"
 - if ["\$x" == "he llo"]; then echo \$x; fi

제어문

- while

- while [조건]; do
 - 실행문
- done

- i=3
- while [\$i -lt 10]; do
 - echo \$i; i=\$((i + 1));
- done

제어문

- 무한루프
 - while ;; do
 - date
 - sleep 5
 - done

제어문

- for

- for 변수명 in 리스트; do
- 실행문
- done

- for dir in *; do
 - [-d "\$dir"] &&
(echo "\$dir"; cd "\$dir";
ls | wc -l)
- done

제어문

- switch-case

- case 변수 in
- 케이스1)
 - 실행문;;
- 케이스2)
 - 실행문;;
- *)
 - 실행문;;
- esac

- a=3
- case "\$a" in
- [0-9])
 - echo integer;;
- [[:upper:]])
 - echo upper;;
- *) echo unknown;;
- esac

제어문

- break
- continue

입출력 리다이렉트

- 표준 입력(standard input) – 0번
 - 키보드 입력이 프로세스로 들어가는 통로
- 표준 출력(standard output) – 1번
 - 프로세스가 출력한 내용이 화면으로 나오는 통로
- 표준 에러(standard error) – 2번
 - 프로세스에서 발생한 에러가 화면으로 나오는 통로

입출력 리다이렉트

- input redirect
 - `sort < in.txt`
- output redirect
 - `ls /lkjfslfjds > out.txt`
 - `ls /lkjfslfjds > out.txt 2> err.txt`

입출력 리다이렉트

- merge(append)
 - `ls /lkjfslfjds > allresult.txt 2>&1`
 - `ls /lkjfslfjds 2>&1 | sort`

필터

- 프로세스의 출력을 다른 프로세스의 입력으로 전달
- 파이프(|) 기호를 이용

- ls | head
- cat test.txt | sort | uniq -c | sort -n

필터

- 특정 단계에서 발생하는 에러도 잡고 싶다면?
 - `find /etc | grep 허가`
 - `find /etc 2>&1 | grep 허가`

파일 내용 보기

- `cat test1.txt`
- `more test1.txt`
- `grep bash .bashrc | less -eMR`

파일 내용 보기

- 파일의 앞쪽 일부만 보기
 - `head -10 test1.txt`
- 파일의 뒤쪽 일부만 보기
 - `tail -5 test1.txt`
 - `tail -f /var/log/lastlog`

파일에서 문자열 찾기

- 찾기

- `grep 문자열 파일`
- `grep "export" /etc/profile`
- `grep -r "export" /etc`

- 잘라내서 쓰기

- `cut -d 구분자 -f 필드번호 파일`
- `grep irteamro /etc/passwd | cut -d: -f1,3-4,7`

Alias

- 별명

- alias

- alias grep="egrep"

- alias diffs="diff -ignore-all-space"

- alias l="ssh hcon.nhnent.com"

함수

- function과 subroutine을 구분하지 않음
 - `func1() { echo "hello, $1"; }`
 - `func1 "world"`

 - `function func2 { return $((($1 + $2)); }`
 - `func2 3 4`
 - `echo $?`

Script File

- shebang
 - #!
 - 어떤 인터프리터 프로그램을 사용할 것인가를 지정
 - #!/bin/bash
 - #!/usr/bin/env python
 - 권장하는 방식
 - #!/bin/awk -f
 - 옵션을 붙여야 할 경우, env를 사용할 수 없음

Script File

- 파일 작성 (test.sh)
 - `#!/bin/bash`
 - `echo "hello world"`
- `bash test.sh`
- `chmod a+x test.sh`
- `./test.sh`

Script File

- command-line arguments
 - \$1, \$2, ...
 - \$#
- exit status
 - \$?
- function의 매개변수와 리턴값도 동일하게 처리

Job Control

- Foreground job
 - 기본적으로는 프로그램은 foreground에서 실행됨
 - 한 명령의 실행이 끝날 때까지 다음 명령은 기다려야 함
- Background job
 - &를 붙여서 실행하면 background에서 실행됨
 - 다음 명령을 바로 실행하거나 프롬프트가 뜸

Job Control

- sleep 500 &
- jobs
- fg % 1
- Ctrl-z
- bg
- wait

사용자 설정

- Bourne shell
 - ~/.profile
- BASH
 - login shell
 - .bashrc
 - non-login shell
 - .bash_profile

추천 설정

- export TERM=xterm-256color
- export LC_ALL=ko_KR.utf8
- export LANG=ko_KR.utf8
- export GREP_OPTIONS='--color=always'
- export CLICOLOR=1
- export LSCOLORS=GxFxCxDxBxegegabagaced
- export PS1="[\u@\h] \W\\$ "
- export LESS='-EMR'
- export EDITOR=vim
- export PAGER="less \$LESS"
- export PATH=/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/bin:/sbin:\$HOME/bin

사용자 설정

- 이미 실행 중인 shell에 적용하려면?
 - `~/.bashrc`
 - `source ~/.bashrc`

자주 사용하는 명령

파일 경로 다루기

- 디렉토리이름 구하기
 - `dirname /usr/local/share/doc/foo.txt`
- 파일이름 구하기
 - `basename /usr/local/share/doc/foo.txt`

화면 청소하기

- clear

실행가능한 프로그램 확인

- which
 - which lsof
 - PATH 환경변수에 포함되어야 탐색 가능

로케일 관련 명령

- locale

- locale
- locale -a

- iconv

- iconv -f CP949 -t UTF-8 *완성형파일* > *UTF8파일*
 - -c 옵션 필수
- iconv -l

정렬과 카운팅

- 정렬
 - sort *파일*
- 인접한 행의 중복 제거
 - uniq *파일*
- 행/열/문자 수 세기
 - wc *파일*

정렬과 카운팅

- 자주 쓰는 조합

- `cut -d: -f7 /etc/passwd | sort -u | wc -l`

- `grep -v "^#" /etc/sysctl.conf | cut -d'.' -f1 | sort | uniq -c | sort -n`

파일 탐색

- find

- find *디렉토리 목록* -name 이름패턴 -type 파일유형 ...
- find /usr/lib64
- find /usr/lib64 -name "lib*.a" -ls
- find /usr/lib64 -type d -name "py*"
- find /usr/lib64 \(-name "python*" -o -name "pgsql*" \) -ls

입력을 개별 처리하기

- for 반복문은 공백이 포함된 입력을 처리하기 곤란함
- xargs를 이용하여 행 단위로 처리하기
 - xargs 명령
 - xargs -I % 명령 "%"
- find /etc/init | xargs wc -l
- find /etc/init | xargs -I % wc -l "%"

다른 사용자로 변경

- su
 - 다른 사용자로 변경하기
 - su irteam
 - su -l irteamsu
- sudo
 - rlogin -l irteamsu 호스트명
 - sudo apps/httpd/bin/apachectl start

접속 비밀번호 변경하기

- passwd
- kpasswd

원격 제어

- rsh - *로그인ID* *호스트명* "*명령*"
- rcp, scp
 - rcp myfile.tgz irteam@*호스트명*:~/경로
- ftp, sftp, rsync

파일 묶기

- 여러 파일을 하나로 묶거나 압축하기
 - tar **cvfz** mypack.tar.gz src/
 - tar **tvfz** mypack.tar.gz
 - tar **xvfz** !\$
- bzip2 포맷은 **z** 대신 **j** 옵션 사용

파일 압축

- zip 포맷
 - `zip -r mypack.zip src/`
 - `unzip -l mypack.zip`
 - `unzip -x !$`

프로세스 관련 명령

- 프로세스 목록 조회
 - ps aux / pstree
- 프로세스 종료
 - kill 프로세스ID / pkill 프로세스명
- 프로세스 이름으로 프로세스ID 알아내기
 - pidof 프로세스명

시스템 관련 명령

- 시스템 자원 사용현황 조회
 - top
- 디렉토리의 크기 조회
 - du -s
- 디스크 전체 사용현황 조회
 - df -k

네트워크 관련 명령

- ifconfig => ip addr
- ping
 - ping 10.161.128.1
- nslookup => getent hosts
 - getent hosts hangame.com

매뉴얼 페이지

- man wc
- man 2 open
- man 3 opendir

명령 조합

- 현재 디렉토리의 서브 디렉토리를 디스크 사용량 순으로 정렬해서 보려면?
- 파일의 130번 행 하나만 출력하려면?
- 특정 디렉토리 이하의 어딘가에 있는 모든 텍스트 파일을 찾아서 그 파일의 최근 수정 시각을 보려면?

명령 조합

- `du -ks * | sort -rn`
- `head -130 test1.txt | tail -1`
- `find /usr -type f | xargs file | grep text | cut -d:
-f1 | xargs -I % ls -l "%"`

Reference

- <http://tldp.org/LDP/abs/html/index.html>



Thank You

for watching my presentation