

SHELL BASICS

안양대학교 컴퓨터공학과 하은용

내용

2

- 기초 명령어
- 셀 변수, 환경 변수

기초 명령어 (1)

3

□ echo [-ne] [문자열...]

- 지정한 문자열을 표준출력
- 여러 개의 문자열이 나열될 경우 출력되는 문자열 사이에 공백 한 개가 출력, 마지막에 newline 출력
- 쉘변수, 환경변수의 값 출력
- 옵션
 - n: 줄바꿈 문자 출력 안함, e: 문자열 중에 포함된 escape 문자 해석

```
$ echo I love linux
I love linux
$ echo -n I love linux
I love linux$
```

```
$ echo $PATH
/bin:/usr/bin:/usr/sbin:/usr/local/bin:.
$ echo $PWD
/home/mylinux
$ echo $SHELL
/bin/bash
```

셸변수

4

□ 변수란

- 셸(혹은 프로세스)이 실행될 때 참조되는 값을 저장한 변수

□ 변수의 유형

- 정의 주체에 따른 분류
 - 사용자 정의 변수(User-defined variables), 미리 정의된 변수(Predefined variables, Reserved variables)
- 전역변수(Global variables), 환경변수(Environment variables)
 - 현재 실행중인 셸뿐만 아니라 서브셸, 혹은 자식 프로세스에까지 상속되는(값이 유지되는) 셸 변수
- 지역변수(Local variables), 셸변수(Shell variables)
 - 현재 실행중인 셸에서만 참조됨, 서브셸에서는 값이 사라짐

변수의 정의

5

□ 변수 이름

- 문자, 숫자, _(언더바)로 구성, 첫 글자는 반드시 문자, 혹은 언더바
- 필수는 아니지만 통상적으로 대문자를 사용
- 변수에 저장되는 값은 모두 문자열로 취급

□ 변수의 정의

- 지역변수
\$ 변수명=값 \$ declare [변수명=값]
- 전역변수
\$ 변수명=값 \$ export 변수명=값
\$ export 변수명
- 정의되지 않은 변수의 초기값은 NULL

□ 변수 해제

- \$ unset 변수명

목록 확인, 읽기 전용, 기타

6

□ 지역변수

▣ \$ set \$ declare

□ 전역변수

▣ \$ env \$ printenv

□ 읽기전용 설정

▣ \$ readonly 변수명

```
$ OS_NAME ="Linux"
-bash: OS_NAME: command not found
$ OS_NAME= "Linux"
-bash: Linux: command not found
$ OS_NAME="Linux"
$ echo $OS_NAME
Linux
$
```

```
$ OS_NAME="Fedora Linux"
$ echo $OS_NAME
Fedora Linux
$ readonly OS_NAME
$ OS_NAME="Olleh Linux"
Bash: OS_NAME: readonly variable
$
```

예제

7

```
$ MY_NAME="Kil-Dong Hong"
$ echo $MY_NAME
Kil-Dong Hong
$ declare MYVAR="1"
$ echo $MYVAR
1
$ bash
$ echo $MY_NAME

$ echo $MYVAR

$ exit
$ echo $MY_NAME
Kil-Dong Hong
$ echo $MYVAR
1
$ unset MY_NAME
$ unset MYVAR
```

```
$ export MY_NAME="Kil-Dong Hong"
$ echo $MY_NAME
Kil-Dong Hong
$ MYVAR="1"
$ export MYVAR
$ echo $MYVAR
1
$ bash
$ echo $MY_NAME
Kil-Dong Hong
$ echo $MYVAR
1
$ exit
$ echo $MY_NAME
Kil-Dong Hong
$ echo $MYVAR
1
$ unset MY_NAME MYVAR
```

미리 정의된 변수(Reserved Variables) (1)

8

□ 미리 정의된 변수란

- ▣ 특별한 목적으로 사용되는 미리 정의된 변수로 일반적으로 특정 값이 미리 할당

□ 종류

변수명	설명	변수명	설명
HOME	현재 사용자 홈 디렉터리	HOSTNAME	현재 호스트의 이름
PATH	셸이 명령어를 찾을 때 참조하는 디렉터리 경로	HOSTTYPE	현재 호스트의 종류
PS1	일차 명령프롬프트 모양	IGNOREEOF	셸프롬프트상에서 EOF를 입력했을 때 logout되는 시점의 EOF 입력 횟수
PS2	이차 명령프롬프트 모양	LANG	Locale 설정
BASH	Bash의 설치 경로	LINENO	현재 수행중인 스크립트, 혹은 함수에서 줄 번호
BASH_VERSION	Bash의 버전	MACHTYPE	시스템유형 CPU-COMPANY-SYSTEM형식
GROUPS	현재 사용자의 그룹 ID	OSTYPE	운영체제 유형
HISTCMD	다음 명령의 history 번호	PWD	현재 작업 디렉터리
HISTFILE	기본 history 파일명	UID	현재 사용자의 real user ID
HISTFILESIZE	History 파일에 저장되는 명령 최대수	USER	현재 사용자의 ID
HISTSIZE	History 목록이 기억하는 명령 최대수		

미리 정의된 변수(Reserved Variables) (2)

□ 특수 변수

- ▣ \$\$ - 현재 프로세스의 PID
- ▣ \$? - 최근 실행한 명령의 종료 값
- ▣ \$0 - 프로그램에 전달된 첫 번째 인자
- ▣ \$1 - 프로그램에 전달된 두 번째 인자
- ▣ \$* - 명령인자 전체를 하나의 문자열로 간주한 값
- ▣ \$! - 마지막으로 실행된 백그라운드 프로세스의 PID

쿼팅 (1)

10

□ 쿼팅 (Quoting) 이란

- ▣ 특수 문자가 포함된 문자열이 쉘에 의해 특수문자가 재해석되거나 확장되는 것을 방지하기 위해 문자열을 따옴표로 묶는 것을 의미

□ 한 문자의 쿼팅

- ▣ 이스케이프 문자 '\ ' 사용

```
$ echo Hello      World
Hello World
$ echo Hello\ \ \ World
Hello      World
$ echo ||
-bash: syntax error near unexpected token `|'
$ echo \|\|
||
$ touch This is a variable
$ touch This\ is\ a\ variable
```

```
$ echo Hello\
> World
Hello
World
$ mkdir cat\'s\ ear
$ ls v*
$ ls v\*
$ ls [a-z]ns
$ ls \[a\-z\]ns
$
```

쿼팅 (2)

11

□ 문자열의 쿼팅

- 큰따옴표("부분 인용"): \$, `(backquote), \ (이스케이프)를 제외한 모든 특수 문자들을 보존
- 작은따옴표('완전 인용'): 작은따옴표를 제외한 모든 문자들을 보존

```
$ echo Hello      World
Hello World
$ echo "Hello      World"
Hello      World
$ echo |      |
-bash: syntax error near unexp ~
$ echo "|      |"
|      |
$ touch This is a variable
$ touch "This is a variable"
$ echo "Current working dir: `pwd`"
Current working dir: /home/mylinux
$ echo "My home is $HOME"
My home is /home/mylinux
```

```
$ echo "Hello
> World"
Hello
World
$ echo "Hello \
> World"
Hello World
$ echo "My home directory is $HOME"
My home directory is /home/mylinux
$ echo 'My home directory is $HOME'
My home directory is $HOME
$
```

쿼팅 (3)

12

□ 이스케이프 문자와 문자열 쿼팅

- ▣ echo 명령에서 -e 옵션을 사용했을 경우 몇몇 문자는 특별하게 해석됨
 - \n: 줄바꿈, \t: 탭, \0xx: 8진수 0xx 값의 문자

```
$ echo "Hello\nWorld"
Hello\nWorld
$ echo -e "Hello\nWorld"
Hello
World
$
```

요약

13

- echo
- declare
- export
- set
- unset
- env
- printenv
- readonly
- 환경변수
- 쉘변수
- 쿼팅