

Computer Programming Batch & Shell Programming 9th Lecture

김현철

컴퓨터공학부

서울대학교

Slide Credits

- 엄현상 교수님
 - 서울대학교 컴퓨터공학부
 - Computer Programming, 2007 봄학기



순서

- Bash Shell Programming
 - Quoting 보충
 - Looping
 - Per-Case Execution
 - Array
 - Example
 - Miscellaneous
 - Q&A
-

Quoting 보충

- Examples of Quoting Rules

person=hatter

Expression	Value
<code>\$person</code>	<code>hatter</code>
<code>"\$person" [or] "'\$person'" [or] "\$person "</code>	<code>hatter</code>
<code>W\$person</code>	<code>\$person</code>
<code>'\$person'</code>	<code>\$person</code>
<code>"\$person" [or] W'\$personW'</code>	<code>'hatter'</code>
<code>W"\$personW" [or] "W"\$personW"" [or] W"\$person"W"</code>	<code>"hatter"</code>
<code>W'\$person'W'</code>	<code>'\$person'</code>
<code>"\$person"</code>	<code>"\$person"</code>

Looping

Double Quotes around \$@

```
for arg in "$@"; do # correct
    echo $arg
done
for arg in "$*"; do # wrong
    echo $arg
done
```

■ Result

```
martini:~$ loop.sh 1 "2 2" 3
1
2 2
3
1 2 2 3
```

Looping (계속)

```
martini:~$ ls  
backup.C backup.c backup1.C backup1.c
```

```
martini:~$ more for.sh  
for cfilename in *$1*u?.[cC]; do  
    echo $cfilename  
done
```

■ Results

```
martini:~$ for.sh a  
backup.C  
backup.c
```

```
martini:~$ for.sh c  
backup.C  
backup.c
```

Looping (계속)

let i=i+1 # bash only

```
i=0
while [ $i -lt 3 ]; do
  echo $i
  i=`expr $i + 1`
done
```

■ Result

```
martini:~$ while.sh
0
1
2
```

Per-Case Execution

```
echo Enter your command \(who, list, or quit\)
while read command; do
  case $command in
    who) who; echo Done with running who;;
    list) ls; echo Done with running ls;;
    quit) break;;
    *) echo Enter who, list, or quit;;
  esac
done
```

■ Result

```
martini:~$ case.sh
Enter your command (who, list, or quit)
who
net001 pts/2 Mar 31 13:26 ... (omitted)
Done with running who
list
...
Done with running ls
date
Enter who, list, or quit
quit
martini:~$
```

Array (bash Only)

- Variable Containing Multiple Values
 - No Maximum Limit to the Size
 - No Requirements That Member Variables Be Indexed or Assigned Contiguously
 - Zero-Based
-

Array (계속)

```
a1=(one two three)
echo ${a1[*]}
echo a1[*]
echo ${a1[@]}
echo a1[@]
echo ${a1[0]}
echo a1[0]
echo ${a1[3]}
unset a1[3]
unset a1[*]
echo ${a1[0]}
```

```
# unset
i=0
echo $i
unset i
echo $i
-----
0
```

■ Result

```
one two three
a1[*]
one two three
a1[@]
one
a1[0]
```

Bash Script Example (계속)

```
i=0
status=0
if [ $# -lt 1 ]; then
    echo Usage: $0 [-b base1 base2...] [-c expo1 expo2...]
    exit 1
fi
for arg in "$@"; do
    if [ -n "`echo $arg | grep '-'" ]; then
        if [ $arg = -b ]; then
            status=1
        elif [ $arg = -c ]; then
            status=2
        fi
    else
        case $status in
            1 ) base[$i]=$arg; i=`expr $i + 1` ;;
            2 ) exponent[$i]=$arg; i=`expr $i + 1` ;;
            * ) echo Usage: $0 [-b base1 base2...] [-c expo1 expo2...]; exit 1 ;;
        esac
    fi
done
# to be continued on the next slide
```

Bash Script Example (계속)

```
if ! [ -d OUTPUT ]; then
  mkdir OUTPUT
  if [ $? -gt 0 ]; then # if the exit status of the previous command indicates a failure
    echo Remove/Rename the OUTPUT File
    exit 1
  fi
fi
for i in ${base[@]}; do
  for j in ${exponent[@]}; do
    if ! [ -e ./OUTPUT/output.$i.$j ]; then
      ./pow $i $j >> ./OUTPUT/output.$i.$j
    fi
  done
done
```

■ Execution

```
martini:~$powsim.sh -b 3 4 5 -c 12 14 16
```

Miscellaneous

- Stream Editor: *sed*
 - Global Search and Replace
 - Character Translator: *tr*
 - Regular Expressions
 - for Editors (Including *sed*) and *grep*
 - Option Interpretation: *getopts*
 - Functions
 - Signals
 - ...
-