

Shell scripting

Purpose

- Putting commands in a file
- Special purpose code
- Runs in interpreted mode
- Rarely used when speed is concerned

Basic Commands

- ls
- cat
- cp
- mv
- date
- who
- more
- echo
- expr
- cd
- head
- tail
- rm
- wc
- pwd
- mkdir
- ps
- chmod
- man
- grep
- sort
- diff
- touch
- find
- nohup
- stty
- sleep

Basic Commands

- ls
- cat
- cp
- mv
- date
- who
- more
- echo
- expr
- cd
- head
- tail
- rm
- wc
- pwd
- mkdir
- ps
- chmod
- man
- grep
- sort
- diff
- touch
- find
- nohup
- stty
- sleep

• *command optional-arguments optional-filenames*

Shell scripts

- Program starts with one of the followings

- `#!/bin/sh`
- `#!/bin/tcsh`
- `#!/bin/bash`

- Structure of a program

```
#!/bin/bash  
# this is comment  
Body of program
```

- Execution

- `bash scriptfile`
- `chmod +x scriptfile`
- `./scriptfile`

Example: Hello world

```
$> cat firstprog  
#!/bin/bash  
echo "Hello world!"
```

```
$> bash firstprog  
Hello world!
```

Example: Variable

```
$> cat secondprog  
#!/bin/bash  
name="Shivi"  
echo "Hello $name"
```

```
$> bash secondprog  
Hello Shivi
```

Example: Command line argument

```
$> cat scriptfile  
#!/bin/bash  
name=$1  
echo "Hello $name"
```

```
$> bash scriptfile Vikram  
Hello Vikram
```


Example: Command line argument (cont)

```
$> cat scriptfile
#!/bin/bash
echo "There are $# parameters."
echo "The parameters are $@"
echo "The script name is $0"
echo "The first parameter is $1"
echo "The second parameter is $2"
```

```
$> bash scriptfile a b c
There are 3 parameters.
The parameters are a b c
The script name is scriptfile
The first parameter is a
The second parameter is b
```

Example: I/O

```
$> prog > outfile
```

- Output of prog will be stored in outfile

```
$> prog >> outfile
```

- Output of prog will be appended to outfile

```
$> prog < infile
```

- prog will read input from file infile

```
$> prog1 | prog2
```

- Output of prog1 will be used as input for prog2
 - `who | wc -l`
 - `ls -l | grep ".txt$"`

Reading input

- Example:

```
echo "Please, enter your firstname and lastname"
```

```
read FN LN
```

```
echo "Hi! $LN, $FN !"
```

Conditional statement: if-then-else

- Example: if-then

```
if [ "xyz" = "abc" ]; then
    statements
fi
```

- Example: if-then-else

```
var1="xyz"; var2="abc";
if [ "$var1" = "$var2" ]; then
    statements
else
    statements
fi
```

Loop

- Syntax of 'for' loop

```
for var in list
do
    statements
done
```

- Example

```
list="1 2 3 4"
for var in $list
do
    echo $var
done
```

Output:

```
1
2
3
4
```

Local variable & function

- Example:

```
HELLO=Hello
function myfunc() {
    local HELLO=World
    echo $HELLO
}
echo $HELLO
myfunc
echo $HELLO
```

Output:

```
Hello
World
Hello
```

Function with parameters

- Example:

```
function myfunc2(){
    exit
}
function myfunc1() {
    echo $1
}
myfunc1 Hello
myfunc1 World
myfunc2
echo myfunc1
```

Output:

```
Hello
World
```

Arithmetic operation

- Example

- `echo $[45*10]`

450

- `echo $((55*10))`

550

- `xyz=`echo $((55*10))` ; echo $xyz`

550

sed

- `$> cat infile`

```
abcd efgh  
IIT Patna  
Mtech CSE students  
Bihta  
Bihar
```

- `$> sed '/IIT/p' infile`

- Output:

```
abcd efgh  
IIT Patna  
IIT Patna  
Mtech CSE students  
Bihta  
Bihar
```

sed (contd)

- `$> cat infile`

```
abcd efgh  
IIT Patna  
Mtech CSE students  
Bihta  
Bihar
```

- `$> sed -n '/IIT/p' infile`

- Output:

```
IIT Patna
```

- Same as `grep`

sed (contd)

- `$> cat infile`

```
abcd efgh  
IIT Patna  
Mtech CSE students  
Bihta  
Bihar
```

- `$> sed '/IIT/d' infile`

- Output:

```
abcd efgh  
Mtech CSE students  
Bihta  
Bihar
```

- Deleted the line containing the pattern

sed (contd)

- `$> cat infile`

```
abcd efgh  
IIT Patna  
Mtech CSE students  
Bihta  
Bihar
```

- `$> sed 's/IIT/I_I_T/g' infile`

- Output:

```
abcd efgh  
I_I_T Patna  
Mtech CSE students  
Bihta  
Bihar
```

- Original file remains the same

sed (contd)

- `$> cat infile`

```
abcd efgh  
IIT Patna  
Mtech CSE students  
Bihta  
Bihar
```

- `$> sed -i 's/IIT/I_I_T/g' infile`

- There will be no output in the terminal

- `$> cat infile`

```
abcd efgh  
I_I_T Patna  
Mtech CSE students  
Bihta  
Bihar
```

awk

- `$> cat infile`

```
name S1 S2 S3
abc 45 90 89
cde 33 47 98
lkm 97 76 52
pqr 76 20 02
jkl 78 88 88
```

- `$> awk '{print $1,$2}' infile`

- Output

```
name S1
abc 45
cde 33
lkm 97
pqr 76
jkl 78
```

awk (contd)

- `$> cat infile`

```
name S1 S2 S3
abc 45 90 89
cde 33 47 98
lkm 97 76 52
pqr 76 20 02
jkl 78 88 88
```

- `$> awk '{print $1","$2","$3","$4}'
infile`

- Output

```
name,S1,S2,S3
abc,45,90,89
cde,33,47,98
lkm,97,76,52
pqr,76,20,02
jkl,78,88,88
```

awk (contd)

- `$> cat infile`

```
name S1 S2 S3
abc 45 90 89
cde 33 47 98
lkm 97 76 52
pqr 76 20 02
jkl 78 88 88
```

- `$> awk 'BEGIN{s=0} {s=s+$2}
END{print s}' infile`

- Output
329

awk (contd)

- `$> cat infile`

```
name S1 S2 S3
abc 45 90 89
cde 33 47 98
lkm 97 76 52
pqr 76 20 02
jkl 78 88 88
```

- `$> awk 'BEGIN{s=0}`

```
{if(NR!=1){s=s+$2}}
END{print s}' infile
```

- Output
329

awk (contd)

- `$> cat infile`

```
name S1 S2 S3
abc 45 90 89
cde 33 47 98
lkm 97 76 52
pqr 76 20 02
jkl 78 88 88
```

- `$> awk '`

```
{if(NR==1){print $0,"Total"}
else{print $0,$2+$3+$4}}
' infile
```

- Output

```
name S1 S2 S3 Total
abc 45 90 89 224
cde 33 47 98 178
lkm 97 76 52 225
pqr 76 20 02 98
jkl 78 88 88 254
```