UNIX SHELL PROGRAMMING

SHELL SCRIPTING

PREETHI.P

Department of Computer Science and Engineering

UNIX SHELL PROGRAMMING

SHELL SCRIPTING

PREETHI.P

Department of Computer Science and Engineering

UNIX SHELL PROGRAMMING

SHELL VARIABLES

Shell supports variables that are useful both in the command line and shell scripts.

Ex: Variable Assignments

```
count=5
echo $count
total=$count
```

Variable names are Case sensitive and begin with a letter and can contain numerals and _

Shell variables are not typed ☺

SHELL VARIABLES

unset: Variables can be removed

readonly: protected from reassignment

Both are internal commands

Unix system and software packages are in UPPERCASE

Note: C shell uses the set statement to set variables

Ex: set c=1 or set c = 1

SHELL VARIABLES

Effects of Quoting and Escaping:

msg = hello\ student\'s\ hope\ you\ are\ doing\ good

msg = "hello\ student\'s\ hope\ you\ are\ doing\ good"

echo pay is \$100

echo 'pay is $100'

UNIX SHELL PROGRAMMING

SHELL VARIABLES

Where to use:

Setting Pathnames: if pathnames are used many times, then assign it to a variable.

Ex: path = '/home/student'

Ex: mydir =`pwd`; echo $mydir

Ex: text = abc; ext =.c

Ex: filename=$text$ext

Ex: filename=$text'.c'

UNIX SHELL PROGRAMMING

SHELL VARIABLES

Where to use:

Setting Pathnames: if pathnames are used many times, then assign it to a variable.

Ex: path = '/home/student'

Ex: mydir =`pwd`; echo $mydir

Ex: text = abc; ext =.c

Ex: filename=$text$ext

Ex: filename=$text'.c'

UNIX SHELL PROGRAMMING

SHELL SCRIPTS

Group of commands have to be executed – they are stored in file and the file itself is shell script/program.

Runs in interpretive mode, each statement is loaded into memory when it is to be executed.

Note: .sh is not mandatory

Shell scripts are executed in a separate child shell process.

You can provide a special interpreter line in the first line of the script to specify a different shell for your script.

UNIX SHELL PROGRAMMING

SHELL SCRIPTS

First Shell : EX1.sh

```
#!/bin/sh
# script.sh: Sample shell script
echo “Today’s date : `date`”
echo “this months calender :”
cal `date “+%m”`
echo “my shell : $SHELL”
```

Execution : EX1.sh or sh ex1.sh

SHELL SCRIPTS

Making Scripts Interactive with read command:

read name

the script pauses to take input from the keyboard. The entered value is stored in variable name.

Second shell:

```
#!/bin/sh
echo “enter the pattern to be searched”
read pat
echo “enter the file name”
read filename
grep “$pat” $filename
```

SHELL SCRIPTS

Command Line Arguments

to run shell scripts in interactive mode command line arguments are passes with redirection and pipeline.

Positional parameters: Arguments are assigned to these Special variables.

First argument is assigned to $1, second $2……

In addition to these , we have some more

$*- Complete set of positional parameters as a single string

$# - Number of arguments specified.

SHELL SCRIPTS

Command Line Arguments

Third Example: ex3.sh

```
#!/bin/sh
# comment
echo “program : $0
the number of argument $#
the arguments are $*”
```

EX: ex3.sh sales emp.txt

Note : in Bash ./ is prefixed to the script name

SHELL SCRIPTS

Command Line Arguments

Few more positional parameters:

“$@”

$?

$$

$!

SHELL SCRIPTS

Command exit and its status

exit 0 : used when everything runs fine

exit 1: when something goes wrong

Ex: grep sales emp.txt ;echo $?

0

Ex: grep students emp.txt; echo $?

1

1 as it was a failure finding students in emp.txt

Logical operators && and || conditional Execution

&& : The && delimits two commands; command 2 is executed only when command 1 succeeds.

EX: grep ‘sales’ emp.txt && echo “pattern sales found ”

|| : The || plays a inverse role; command 2 is executed only when command 1 fails.

Ex: grep “student” emp.txt || echo “pattern not found”

grep “student” emp.txt || exit 2

UNIX SHELL PROGRAMMING

SHELL SCRIPTS

The If conditional

```
if command is successful
 then
execute commands
else
execute commands
if

if command is successful
then
execute commands
if
```

SHELL SCRIPTS

The If conditional

```
if command is successful
 then
execute commands
elif command is successful
then
else
execute commands
if
```

THANK YOU

PREETHI.P

Assistant Professor , Department of Computer Science

preethip@pes.edu