Lecturer George Ajam

Nov 1st . 2012

Network Dept. / College of IT

Shell Scripting

- What is shell?
- Basic
- Syntax

Why Shell?

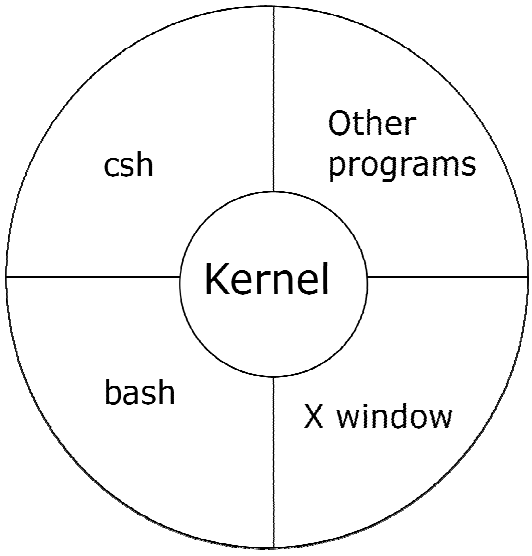
- The commercial UNIX used Korn Shell
- For Linux, the Bash is the default
- Why Shell?
  - For routing jobs, such as system administration, without writing programs
  - However, the shell script is not efficient, therefore, can be used for prototyping the ideas
- For example,

ls –al | more (better format of listing directory)

man bash | col –b | lpr (print man page of man)

What is Shell?

- Shell is the interface between end user and the Linux system, similar to the commands in Windows
- Bash is installed as in /bin/sh
- Check the version **$ /bin/bash --version**



Pipe and Redirection

- Redirection (< or >)

ls –l > lsoutput.txt (save output to lsoutput.txt)

ps >> lsoutput.txt (append to lsoutput.txt)

more < killout.txt (use killout.txt as parameter to more)

kill -l 1234 > killouterr.txt 2 >&1 (redirect to the same file)

kill -l 1234 >/dev/null 2 >&1 (ignore std output)

- Pipe (|)
  - Process are executed *concurrently*

ps | sort | more

ps –xo comm | sort | uniq | grep –v sh | more

$cat mydata.txt | sort | uniq | > mydata.txt (generates an empty file !)

Shell as a Language

- We can write a script containing many shell commands
- Interactive Program:
  - grep files with POSIX string and print it

for file in *

> do

> if grep –l POSIX $file

> then

> more $file

- **fi**
- **done**

Posix

There is a file with POSIX in it

- ‘*’ is wildcard

more `grep –l POSIX *`

more $(grep –l POSIX *)

more –l POSIX * | more

Writing a Script

- Use text editor to generate the “first” file

```
#!/bin/sh
# first
# this file looks for the files containing POSIX
# and print it
for file in *
do
  if grep –q POSIX $file
  then
    echo $file
  fi
done
exit 0
```

$ /bin/sh first

$ chmod +x first

$./first (make sure . is include in PATH parameter)

Simple Bash Script

Our objective in this simple script is to write a Hello World example:

We need first to use an editor of any type to write our script, so you may choose between (vi, nano or gedit), (let me choose vi **mytest**) mytest will represent the file name to be created. When I use vi I should press the letter ***i in order*** to start writing (**i**nserting) to the editor! We'll get to know more about vi soon.

```
#! /bin/bash
# This is a sample script of hello world

echo "Hello World"
```

Line 1: specifies which shell should be used to interpret the commands in the script.
Line 2: is a comment (has no effect when the script is executed).
Line 3: displays a message ***Hello World***.

To run the script:

- Method 1:
  - Make the script executable: chmod u+x **mytest**
  - Try to run the script by typing the command: **mytest**
  - You will get the error message: command not found
  - Remember, a unix/Linux system will only only look for commands or scripts in the directories in your search path. So the system looks for the "mytest" command in the directories /usr/bin and /bin, doesn't find it and returns the error message.

- Run the script with the command:
  ./mytest
  which means: run **mytest** from the current directory then the result would be

  Hello World

- Method 2:
  - If you are getting error messages when you run the script, try to trace the lines as they execute using the command: bash -v mytest
  - As the script executes, each line is displayed on the screen so that you know exactly what your script is doing.

A very simple backup script

```
#!/bin/bash
tar -cZf /var/my-backup.tgz /home/me/
```

Home Work

Please write to George.ejaam@gmail.com after you try the simple backup script; what's going to happen? Your answer should contain explanations details from the help of what is clearly used. THIS HOME WORK ID DUE TILL 7-11-2012 10:00 PM Iraq's time, don't email it later than the deadline please.