Lab Assignment – Terminal Part 2

CS380

Lab Assignment

Details: A shell script is a sequence of shell commands written in an executable script file. Executing this file instructs the shell to execute all commands in the order of their appearance in the script file. Go through the following pages in this assignment to get started with shell programming. And then complete the below assignment.

Assignment: Write a shell script that displays various system parameters by using shell commands like *who*, *whoami*, *date*, *hostname*, etc.

Submit your completed assignment through Moodle. You should submit (1) your shell script & (2) a screenshot of the output from running your shell script in Terminal.

Assignment submission deadline is provided on Moodle. Penalties will be imposed on late submissions.

Tip for screenshots in Ubuntu: https://help.ubuntu.com/stable/ubuntu-help/screen-shot-record.html

Shell Programs

(These pages are intended to help you get started with Shell programming.)

- Creating and Executing Shell Programs
- emacs Text Editor
- Adding Comments

How to Create and Execute a Shell Program

- Use a text editor such as emacs or vi to create a new file
- Enter a “shebang” (#!) indicating which shell (sh, bash, csh, ...) should execute the program
- Enter shell command lines (and optionally, shell control structures for branching and looping)
- Save the new file and exit the text editor

- Turn on execute permission for your new file
- Make sure the new file is in a directory where the shell looks for commands (PATH variable)

- Invoke the shell program by using the new file name as a command name

Text Editors

vi: visual text editor (wysiwyg) compared to older line-oriented editors (ex and ed)

“moded” editor ... need to use a command to allow adding text to a file

vim: vi improved

has both a command line interface and a graphical user interface

emacs: text editor known for being customizable and extensible

nice interface to R, LaTex, C/C++

“non-moded” editor ... entered text becomes part of file ...

control sequences are used as editing commands

aquamacs: “a modern editor based on emacs that makes Mac users feel at home”

*** here we briefly illustrate basic emacs, which is available on both Linux and Mac OS X

resources for learning emacs

- interactive tutorial: within emacs, use `<CTRL>h t`
- manual: http://www.gnu.org/software/emacs/manual/
- aquamacs: http://aquamacs.org/

Basic emacs Text Editing Commands

enter emacs to edit existing file	`emacs <file.existing>`
enter emacs to create a new file	`emacs <file.new>`
save file	`<CTRL>x <CTRL>s`
exit emacs	`<CTRL>x <CTRL>c`
move cursor one character forward	`<CTRL>f`
move cursor one character backward	`<CTRL>b`
move cursor to next line	`<CTRL>n`
move cursor to previous line	`<CTRL>p`
delete current line	`<CTRL>k`
delete current character	`<CTRL>d` or `<Delete>` or `<Backspace>`
undo last edit	`<CTRL>u`
access help	`<CTRL>h`
access emacs interactive tutorial	`<CTRL>h t`

Creating and Executing a New Shell Program

```
$ emacs myprog
 #!/bin/bash
 echo hello
 date
 who am i
 echo have a good day
 <CTRL>x <CTRL>s
 <CTRL>x <CTRL>c
$ chmod +x myprog
$ echo ${PATH}
/usr/local/bin:/bin:/usr/bin
$ pwd
/u/dkoffman/unix
$ PATH=${PATH}:/u/dkoffman/unix
$ myprog
 hello
 Thu Apr 10 13:00:46 EDT 2014
 dkoffman  pts/80   2014-04-10  12:59  (abc-xyz-princeton.edu)
 have a good day
$
```

Comments

# starts a comment

<CR> ends a comment

```
$ cat wdata_le_part_scan
#
# Output consists of the first 4 lines
# of all wdata_le_part_[a-z][a-z] files
# in the current directory.
#
# Output is placed in a single file
# called wdata_le_part_scan.out
# in the current directory.
#
#!/bin/bash
head -4 wdata_le_part_[a-z][a-z] > wdata_le_part_scan.out
$
```