Assignment 1 - Shell scripting

itws2 instructors 2012-Spring, IIIT HYDERABAD

January 22, 2012

Contents

Do NOT copy codes or reports. Copying in ANY form will be heavily penalized.

1 Submission Instructions

Please follow these instructions **exactly** as outlined.

1.1 File naming

- You must submit all problems from Levels 1,2 and 3.
- Name the files according to the question and Level. For problem `1.1` (first problem of Level 1) name the file as `1-1.sh`, problem `2.2` as `2-2.sh` and so on.
- Keep all your program files in a single directory named `codes` (make **NO** sub-directories).

1.2 Report

- You must submit a report for this assignment. The report will be an `org` file. It must have the following
  - Title of the report (Mention the assignment number)
  - Name and Roll Number
  - For each question you solve:
    - Describe in about 2-3 lines how you solved the problem. Be brief. *Verbosity will be penalized.*
    - Include a link to the code using the `file:` method in org.
    - Include a sample input and output in your report using the `#begin_example` method in org.
- Export the org file to a `.html` file and rename this file to `index.html`
- Keep all your `.org, .html` files in a `report` directory

1.3 Directory structure

- All codes should be in a `codes` directory. The files should be named as mentioned above.
- All report related material should be in a `report` directory. The directory must also contain your report in HTML format in a file called `index.html`
- Your final directory structure should resemble

```
.
|-- codes
|   |-- 1-1.sh
|   |-- 1-2.sh
|    ...
|-- report
|    |-- index.html
|    |-- index.org
```

1.4 Deadline

- The deadline for the assignment is 23rd Jan, 2012. Please do not ask for an extension.

2 Practice Problems

These problems need not be submitted.

2.1 Single line commands

Write commands that use a combination of linux commands, the `|` operator to achieve the following. Do not use `;` or `&` operators.
e.g. `ls ; cat a.txt` is **not** valid.
`ls | grep a.txt` is valid.

1. Print a list of all files (in current directory and sub-directories) that contain the word "iiit". What can you do so that you will ignore the case of the letters in the word, i.e. even files with words like iIiT, IIIT,IIit etc will be reported ?
2. Display the total number of files (**not directories, only files**) in the current directory.

3. Display the 10 largest files (**not directories, only files**) in the current directory.

4. Convert all occurences of character 'a' in a string to 'e'.

2.2 Concepts

1. Find which processor the system is using. e.g. Intel Atom, Intel Core i5 etc.

2. How do you create a hidden file in Linux ? How do you list **all** (including hidden) files ?

3. Find if firefox is running on the system and terminate the process.

4. Delete only the contents of a file, and not the file itself (use redirection operators).

5. Compile the file addnos.c and generate ' `a.out` '. Run it as ' `./a.out &` '. Check if the program is running. Will it ever finish executing ? Why ?

3 Level 1 Problems

3.1 List contents of `.tar`, `.tar.gz` files

Write a script that will list the contents (files, directories etc.) in a `.tar` or `.tar.gz` file. Accept the filename as a commandline argument. You can assume that the substring `.tar` or `.tar.gz` does not occur anywhere else in the filename, except as an extension.

```
Input
$> ./1-1.sh file.tar

Sample Output
a.txt
b.txt
new_folder
...
img_02.jpg
```

i.e. list of files and directories in file.tar

```
Input
$> ./1-1.sh file.txt

Output
error: given file is not a tar file
```

3.2 Processes without a terminal

Write a single line program to find processes which are not attached to a terminal. List (process name, pid) pairs.

```
Input
$> ./1-2.sh
Sample output
/usr/bin/firefox 2137
/usr/bin/thunderbird 3451
....
/bin/evince 6792
```

4 Level 2 Problems

4.1 Simulating `echo -n`

You need to write a shell script that simulates the functionality of `echo -n` (i.e. given an argument, the command just prints it on `stdout` without a `\n`).

```
Sample Input
$prompt> ./2-1.sh this is testing

Output
this is testing$prompt>
```

Note: There is no `\n` in the output

4.2 Process with maximum memory

Write a single line command to find the process consuming the maximum memory. List the (process name, pid) pair.

```
$>./2-2.sh
Sample output
/usr/bin/firefox 3421
```

4.3 List of logged in users

Display only the names of users logged into the system.

```
$>./2-3.sh
Sample output
user1
user23
```

4.4 Using Grep

When we execute ‘ `ps -elf | grep firefox` ‘ the output shows the process ‘ `grep` ‘ too. Write a script so that this does not happen.

```
Input
$>./2-4.sh firefox

Sample Output
0 S user 2950    1  2 80 0 - 141365 poll_s 18:35 ? 00:03:03 /usr/bin/firefox
0 S user 3349 2950  0 80 0 - 13167  poll_s 19:32 ? 00:00:00 /usr/firefox/plugins
0 S user 3351 2950  0 80 0 - 13169  poll_s 19:32 ? 00:00:00 /usr/firefox/plugin/totem
```

Note

```
0 S user 4381 3090 0 80 0 - 832 pipe_w 20:31 pts/3 00:00:00 grep firefox
```

is **not** present.

5 Level 3 Problems

5.1 Largest file finder

Write a shell script to find the largest file in a directory and its immediate sub-directories (1 level deep only). Take the directory name as a command-line argument

```
.
|-- assign4.pdf
|-- assign5.pdf
|-- Book
|   |-- a.rtf
|   |-- b.jpg
|   |-- Notes
|   |   |-- a.txt
|   |   |-- b.txt
|   |-- Images
|   |   |-- 1.jpg
```

With reference to the directory structure, running the code in the current directory will search for files in the current directory and the Book directory (**not** the Notes or Images directory). Only files `assign4.pdf, assign5.pdf, a.rtf, b.jpg` will be considered while finding the largest file.

```
$>./3-1.sh /home/user/itws2
Sample Output
/home/user/itws2/Book/b.jpg
```

5.2 Emacs cleanup

You know that `emacs` makes files like `a.txt~` once you open files like `a.txt`. After a period of time, these files occupy unnecessary disk space. Write a shell script which removes all such files from a directory. The directory name is taken as a command line argument. Your output should be the list of files which you removed.

```
Input
$>./3-2.sh /home/user/work/itws2/assignment

Output
Removed files
a.txt~
assign.org~
```

5.3 Youtube downloader

Write a shell script to download youtube videos as an flv file. The exact problem scenario is as follows. Assume you are on a Linux system which

has both firefox and Adobe Flash (Flash Version >= 10.2) installed. Your firefox is already open and has a single youtube video playing in it. Your script should be able to save this video as a flv file. Furthermore, the name of this output flv file should be taken as a command line argument.

```
Sample Execution
$>./3-3.sh myvideo.flv

Sample Output
Video myvideo.flv saved successfully.
```

(The current directory now contains a file called `myvideo.flv` which has the required youtube video)

5.3.1 Installing Flash Plugin

Depending on the OS version, download the appropriate file. You can use "uname -i" command to find which version of the OS you are running.

- i686/i386 : 32 Bit Flash Plugin
- x86_64 : 64 Bit Flash Plugin

Then copy the file to `~/.mozilla/plugins/`. Create the appropriate folders if they don't exist This should install flash plugin for firefox. To confirm, start firefox and type `about:plugins` in the Address Bar and check if Shockwave Flash is listed there.

5.3.2 Checking Flash Plugin Version

Visit the link http://kb2.adobe.com/cps/155/tn_15507.html