

CPT264 – UNIX (Linux) Systems Administration and Programming (USAP), Study period 3, 2016.

Assignment 1

Due: 9th October 2016, 9 pm.

Overall Description

An important part of the management of a Linux installation is to use shell scripts in combination with the right choice of system programs. Many of the skills you will learn in this assignment are using programs you would use for systems maintenance. The main limitation will be that your script will not have super user access and therefore you should not expect your program to be running as root.

For this assignment, you will be required to report on various aspects of the state of the system via a menu-driven interface. You should complete each component in the order specified and ideally during the week specified as each part of the system is designed to map to material covered during that week of the course.

All scripts written for this assignment should be written for the with appropriate script preamble (the shebang/ hashbang). All your scripts should have appropriate permissions to be runnable by a normal user.

It is up to you where and how you develop your scripts but they must execute in the University-provided environments of Saturn, titan and Jupiter and they must pass testing with shellcheck (www.shellcheck.net). If you scripts fail with either of these there will be **substantial** penalties (a fail mark).

It is recommended that you maintain a git repository that contains the current state of your assignment at all times. All scripts submitted must be your own work – submitting someone else's work as your own is plagiarism. All cases of plagiarism will be pursued.

You should submit a single .zip file / tarball to blackboard that contains all the scripts developed for this program.

Late Submission Policy

A standard penalty of 10% of the marks available for each day late will be applied. We will not accept assignments that are over 5 days late.

Applications for extensions

Extensions must be requested no later than 24 hours before the deadline. Extensions will not be provided simply for starting late or struggling with the material. You need to show adverse circumstances that have impacted on your ability to complete the assessment. Please see the following document for further details:

<http://www1.rmit.edu.au/students/assessment/extension>

Academic staff can give an extension for a maximum of seven days. All longer extensions must be requested from special considerations: <http://www1.rmit.edu.au/students/specialconsideration>

Assessed Requirements

Coding standards (20%)

For this assignment we expect you to conform to good coding standards. For example you should have variables at the start of each script specifying the path to any binaries used, all variables should be upper case and have meaningful names. You should also make use of functions wherever necessary to make your scripts more readable, and you should explain what your program does with adequate comments.

You should also provide a README file that explains how to use your program and any issues you had in developing your scripts.

Part 1: Week 3 (20%)

For this part you should write a script called “basic” that accepts the following arguments. Please note that for these requirements you don’t need to filter anything down – it is sufficient to just call the correct program to perform the task:

```
./basic freemem
```

displays the amount of free memory available on the system.

```
./basic diskfree
```

displays the amount of disk space free on each partition on the system. No error messages from the system should be displayed and the output from this command should be “paged” so that the user can scroll through the results.

```
./basic quota
```

displays the disk quota(s) for the current user

```
./basic groups
```

displays the groups that the current user is a member of

```
./basic history
```

displays a history of the commands that have been entered into the current shell. The results of this command should be “paged” so that the user can scroll through the results.

Please note that you may need to do additional research to discover which program on the system can provide the required information. All error messages from the programs you call should be ignored but if the command did not succeed you should provide your own error messages.

Part 2: Week 4 (20%)

For this part you should write a script called “advanced” that accepts the following arguments. A high distinction attempt at this section is expected to use “getopts”:

```
./advanced -p
```

prints the number of cpus available in the system.

```
./advanced -o
```

outputs the maximum number of open files allowed per user

```
./advanced -b wildcard destination
```

searches for all files that match the wildcard specified and copies them to the destination specified. Your script should test for whether the destination directory exists and if it does not, create it. Hint: use ‘find’ for this requirement. Example:

```
./advanced -b \*.txt mytextfiles
```

would copy all .txt files that live under the current directory to a directory called mytextfiles.

Part 3: Week 5 (20%) – Parsing the /etc/passwd file

For this requirement, you should write a script called “userman” that uses standard programs to retrieve the following information:

```
./userman userid user
```

displays the user id (a number) for the user specified

```
./userman groupid user
```

displays the default groupid for the user specified

```
./userman comment user
```

displays the comment field (GECOS field) for the username specified

```
./userman home user
```

displays the home directory of the user specified

```
./userman shell user
```

displays the default login shell for the user specified

Note: in all cases, “user” is a username that exists on the system and so you should be validating that the user exists before outputting the required data.

Please note that in the real world we would never do this. There are far more robust programs to use for manipulating of passwd file data. I am much more interested in your ability to parse tabulated data here than whether this is the right or wrong thing to do.

Part 4: week 6 (20%) – Menu system

Create a menu system script called “sysman” that will allow interactive input from the user and then call the appropriate script to perform the function. The following are the menus to be displayed for the main menu, the basic menu, the advanced menu and the passwd menu:

The main menu:

```
#####
#
#  Welcome to System Manager
#  -----
#
#  Please select a menu item:
#
#  1) basic menu options
#  2) advanced menu options
#  3) passwd file options
#  4) quit
#
#  Please enter a choice:
#####
```

The basic menu:

```
#####
#  Basic Tasks Menu
#  -----
#
#  1) Show free memory
#  2) Show free disk space
#  3) Show disk quota
#  4) Show the groups I am a member of
#  5) Show the history of commands for bash
#  6) Quit this menu
#
#  Please enter an option:
#####
```

```
#####
#  Advanced Tasks Menu
#
#  1) Print the number of CPUs available on the system
#  2) Print the maximum number of open files allowed per user
#  3) Backup all files that match a criterion
#  4) Quit this menu
#
#  Please enter your choice:
#####
```

The passwd menu:

```
#####
#  Passwd Tasks Menu
#
#  1) Print the userid of a specified user
#  2) Print the groupid of a specified user
#  3) Print the comment field for a specified user
#  4) Print the home directory of a specified user
#  5) Print the default shell of a specified user
#  6) Exit this menu
#
#  Please enter your selection:
#####
```

Please note that userid and groupid are the numbers, not the names for the user / group.

Please note that in some cases you will need to add additional prompts to read in input from the user.

Files to Submit

You should submit 4 script files:

- basic
- advanced
- userman
- sysman

But these should be compressed into a single archive. I will accept either a zip file or a tar ball. No other submissions will be accepted.

Question and Clarifications

All queries you have about implementation of this assignment, include inconsistencies you discover in the specification must be raised with your instructor on blackboard.

Please verify all assumptions you make as otherwise you risk a loss of marks.