

KIT501 Assignment 2- UNIX Shell Programming

(10% of your KIT501 final result)

Due: 5pm on Thursday the 10th of October - Week 12

Introduction

This is an individual assignment. There are two (2) shell programming tasks in this assignment that use skills and knowledge you have learned from the unit's practical sessions. You are required to make a directory named **kit501a2** under your home directory (on *alacritas*) and use **kit501a2** as your working directory for this assignment.

Assignment submissions will be marked on the host *alacritas* as used in practicals. If you write and test your scripts elsewhere it's your responsibility to ensure that your scripts run correctly on the ICT Discipline's UNIX system (*alacritas*). If your scripts do not run on *alacritas*, they will receive zero marks immediately.

Task A

Write a shell script (to run on the *Bourne* shell) that can be used to remove some old C programs you no longer wish to keep. When the script is run with no arguments supplied, it picks up each C program from the current directory and lists the first 10 lines (hint: research the **head** command). It then prompts for deletion of the file (the user can then choose to delete or not to delete the file). When the user supplies arguments (C program file names) with the script, it works on those files only.

Your script for this task **must** be named **delC.sh**. In designing your script you should consider the following scenarios:

- There is no C program file stored under the current directory;
- There is at least one C program file stored under the current directory;
- The user-supplied arguments list contains names of existing C programs stored under the current directory. We can assume that all the C programs have been correctly named as *.c, and that there are no special characters (such as space) in any of these filenames;
- The user-supplied arguments list contains both existing and missing names of C programs stored under the current directory. Missing names refer to the files which no longer exist under the current directory.

Make sure your script is user-friendly and follows common sense (for example, when there is no C program stored under the current directory your script should display a message and then exit).

continued...

Task A – sample output

To illustrate the behaviour of the script, the following are sample outputs, noting the '\$' is the shell prompt.

```
$ ./delC.sh
This script removes C files which you no longer want to keep.
Here are the C file(s) under the current directory:
No C files found.
```

```
$ ./delC.sh
This script removes C files which you no longer want to keep.
Here are the C file(s) under the current directory:
f1.c f2.c

Displaying first 10 lines of f1.c:

/* A C program for handling arrays
   Written by John Jones, July 2009
   Updated April 2010
*/

#include <stdio.h>

int main()
{
    int x;

Delete file f1.c? (y/n):n (user input)
File f1.c NOT deleted.

Displaying first 10 lines of f2.c:

/* Another C program for handling arrays
   Written by John Jones, August 2009
   Updated May 2010
*/

#include <stdio.h>

int main()
{
    int x, y, z;

Delete file f2.c? (y/n): y (user input)
File f2.c deleted.
```

continued...

```
$ ./delC.sh f1.c
This script removes C files which you no longer want to keep.
The file(s) you want to delete is/are:
f1.c

Displaying first 10 lines of f1.c:

/* A C program for handling arrays
   Written by John Jones, July 2009
   Updated April 2010
*/

#include <stdio.h>

int main()
{
    int x;

Delete file f1.c? (y/n):y (user input)
File f1.c deleted.
```

```
$ ./delC.sh f2.c f3.c
This script removes C files which you no longer want to keep.
The file(s) you want to delete is/are:
f2.c f3.c

Displaying first 10 lines of f2.c:

/* Another C program for handling arrays
   Written by John Jones, August 2009
   Updated May 2010
*/

#include <stdio.h>

int main()
{
    int x, y, z;

Delete file f2.c? (y/n): y (user input)
File f2.c deleted.

Displaying first 10 lines of f3.c:

File f3.c does not exist.
```

```
$ ./delC.sh f3.c f4.c
This script removes C files which you no longer want to keep.
The file(s) you want to delete is/are:
f3.c f4.c

Displaying first 10 lines of f3.c:

File f3.c does not exist.

Displaying first 10 lines of f4.c:

File f4.c does not exist.
```

Task B

The second task is to write a shell script (to run on the Bourne shell) that allows a user to view, add, or delete a setting in a configuration file (**config.txt**) that contains settings in the form **variable=value**. The following is an example of such configuration file:

```
HOME=/home/abc
HOST=alacritas
HOSTTYPE=sun4
LOGNAME=abc
OSTYPE=linux
PATH=/usr/dt/bin:/usr/openwin/bin:/bin:.
PS1=$
PS2=>
SHELL=/usr/bin/tcsh
TZ=Australia/Tasmania
USER=abc
VENDOR=sun
EDITOR=joe
```

Your script for this task must be named **setting.sh**. For ease of use, your script must present a menu of operations that a user may choose from. After the user makes a selection and that the selected operation has been completed, the menu must be displayed again so that the user can make another selection. Validation check on user inputs is required (see the following sample output about this). In the beginning of your script you need to check to see whether the required configuration file (**config.txt**) actually exists under the current directory (if not, your script displays a message and then exits).

Task B – sample output

Here is a sample output of your script. The \$ is the shell prompt. The items in *italics* are not part of the sample output. They are hints indicating how your script should behave.

```
$ ./setting.sh

*** MENU ***
1. Add a Setting
2. Delete a Setting
3. View a Setting
4. View All Settings
Q – Quit

CHOICE: 1 (user input)
Enter setting (format: ABCD=abcd): (user simply presses the Enter/Return key)
New setting not entered

Enter setting (format: ABCD=abcd): EDITOR (user input)
Invalid setting (A valid setting needs to contain a "=" sign)

Enter setting (format: ABCD=abcd): EDITOR= (user input)
The variable name of the setting is: EDITOR
The variable value of the setting is:
Invalid setting.
```

(Hint: To retrieve a variable name before the "=" sign, research the expr command's ability in handling strings)

Enter setting (format: ABCD=abcd): =vi *(user input)*

The variable name of the setting is:

The variable value of the setting is: vi

Invalid setting.

Enter setting (format: ABCD=abcd): 1EDITOR=vi *(user input)*

The variable name of the setting is: 1EDITOR

The variable value of the setting is: vi

Invalid setting. The first character of a variable name cannot be a digit.

Enter setting (format: ABCD=abcd): EDITOR=vi *(user input)*

The variable name of the setting is: EDITOR

The variable value of the setting is: vi

New setting added.

*** MENU ***

1. Add a Setting
 2. Delete a Setting
 3. View a Setting
 4. View All Settings
- Q - Quit

CHOICE: 1 *(user input)*

Enter setting (format: ABCD=abcd): USER=jchen *(user input)*

The variable name of the setting is: USER

The variable value of the setting is: jchen

Variable exists. Changing the values of existing variables is not allowed.

*** MENU ***

1. Add a Setting
 2. Delete a Setting
 3. View a Setting
 4. View All Settings
- Q - Quit

CHOICE: 2 *(user input)*

Enter variable name: EDTOR *(user input)*

Variable does not exist. *(Your script needs to check whether a variable exists or not)*

*** MENU ***

1. Add a Setting
 2. Delete a Setting
 3. View a Setting
 4. View All Settings
- Q - Quit

CHOICE: 2 *(user input)*

Enter variable name: EDITOR *(user input)*

EDITOR=vi

Delete this setting (y/n)? y *(user input)*

Setting deleted *(However, if user's answer is n here, then the setting stays)*

*** MENU ***

1. Add a Setting
 2. Delete a Setting
 3. View a Setting
 4. View All Settings
- Q - Quit

CHOICE: 3 *(user input)*

Enter variable name: USER1 *(user input)*

Variable does not exist. *(Your script needs to check whether a variable exists or not)*

*** MENU ***

1. Add a Setting
 2. Delete a Setting
 3. View a Setting
 4. View All Settings
- Q - Quit

CHOICE: 3 *(user input)*

Enter variable name: USER *(user input)*

USER=abc

Requested setting displayed above.

*** MENU ***

1. Add a Setting
 2. Delete a Setting
 3. View a Setting
 4. View All Settings
- Q - Quit

CHOICE: 4 *(user input)*

HOME=/home/abc

HOST=alacritas

HOSTTYPE=sun4

LOGNAME=abc

OSTYPE=linux

PATH=/usr/dt/bin:/usr/openwin/bin:/bin:.

PS1=\$

PS2=>

SHELL=/usr/bin/tcsh

TZ=Australia/Tasmania

USER=abc

VENDOR=sun

*** MENU ***

1. Add a Setting
 2. Delete a Setting
 3. View a Setting
 4. View All Settings
- Q - Quit

CHOICE: 5 *(user input)*

Invalid choice.

*** MENU ***

1. Add a Setting

```
2. Delete a Setting
3. View a Setting
4. View All Settings
Q - quit
```

CHOICE: q (*user input*)

(The running script is terminated. The shell prompt is displayed)

Script requirements

You must:

- Include **your name, student ID**, and a **brief introduction** of what the script does in all your shell scripts, as a comment in the beginning of each script.
- Make your scripts run on the **Bourne** shell, regardless of which shell the user of your scripts is currently on.
- Add in-line comments to help other people understand your scripts.
- Use "\n" where appropriate to make the output of your scripts more readable.
- Note that your script structure and layout are also important as they will be marked as part of the assessment process.

Note on Plagiarism

Plagiarism is a very serious matter, and ignorance of this is not an excuse. If you submit work, claiming it to be your own, and it is not original work of your own this is cheating. This includes (but is not limited to) submitting code written by others or paying individuals or companies to do the work for you. Plagiarism can result in academic penalties both in relation to your assignment, and also on your permanent university record.

Assignment Submission

The assignment is due on Thursday the 10th of October (Week 12) at 5 pm.

You must submit a single folder named **kit501a2** which contains the following three (3) files, electronically to the **kit501submit** folder which will be created for you in your **alacritas** home account:

```
delC.sh, setting.sh, config.txt
```

(Please make sure that your **kit501a2** folder only contains the three files as listed here, when you are ready to submit them)

To submit your assignment, follow these two steps:

1. Ensure that both your **kit501a2** folder and your **kit501submit** folder are stored right under your home directory;
2. Run the following commands to copy your **kit501a2** folder into your **kit501submit** folder (the \$ is the shell prompt):

| | |
|--------------------------------|---|
| \$ cd | <i>(This takes you back to your home directory from anywhere)</i> |
| \$ cp -r kit501a2 kit501submit | <i>(This copies your assignment folder into your kit501submit folder. The option -r is important)</i> |
| \$ ls -l kit501submit | <i>(This verifies that your assignment folder is now in your kit501submit folder)</i> |
| \$ ls -l kit501submit/kit501a2 | <i>(This verifies that your assignment files are now in your kit501submit folder)</i> |

IMPORTANT NOTE: The `kit501submit` folder will be created for you automatically close to the submission time – **do not create it yourself** as this may cause your assignment to not be submitted correctly. If the `kit501submit` folder does not exist, please visit the ICT discipline’s Help Desk immediately. **Without a `kit501submit` folder, you will NOT be able to submit your assignment 2.**

You must also submit a signed cover sheet to the KIT501 assignment box located at the ICT discipline’s Help Desk or Reception. **Assignments without a signed coversheet will NOT be marked.**

It is your responsibility to ensure that your assignment has been successfully submitted to the correct folder on *alacritas*. If you require assistance with this, please visit the ICT discipline’s Help Desk.

If your assignment is late then you should submit your files to the `kit501late` folder. The late folder will be created automatically after the due date and will be available for one week only.

Need Help?

You are encouraged to seek assistance from your lecturer after you have seriously thought about the assignment. Please note that we can provide general advice, however, we will not help you write any code, nor will we help you debug.

The marking scheme is attached on the next page

Appendix: KIT501 Assignment 2 Marking Scheme

Script delC.sh (for each item there are only 3 possible marks: 100% or 50% or 0%)

| | Mark | Out of |
|--|------|--------|
| Execution of the shell script Script runs as expected (2). Script runs but not as expected (1). Script does not run (0) | | 2 |
| Display a message then exit when there is no C file under current directory | | 2 |
| Pick up each C file from current directory | | 2 |
| List the first 10 lines then prompt for deletion | | 2 |
| When user supplies C file names as arguments work on those files only | | 2 |
| Can handle invalid file names contained in user-supplied arguments list | | 2 |

Script setting.sh (for each item there are only 3 possible marks: 100% or 50% or 0%)

| | Mark | Out of |
|--|------|--------|
| Execution of the shell script Script runs as expected (2). Script runs but not as expected (1). Script does not run (0) | | 2 |
| Correctly allow user to remove settings | | 2 |
| Correctly allow user to view a setting | | 2 |
| Correctly allow user to view all settings | | 2 |
| Checking the existence of the configuration file | | 2 |
| Checking the existence of the variable name for deleting or viewing a setting | | 2 |
| Correctly allow user to add new settings | | 2 |
| Add new settings - Changing the values of existing variables not allowed | | 2 |
| Add new settings - verify user input contains a "=" sign | | 2 |
| Add new settings - retrieve variable name and variable value of a new setting | | 2 |
| Add new settings - first character of a variable name cannot be a digit | | 2 |

Others (for each item there are only 3 possible marks: 100% or 50% or 0%)

| | Mark | Out of |
|--|------|--------|
| Shell scripts structure and layout Clear and tidy (2). Somewhat messy but understandable (1). Messy (0) | | 2 |
| Appropriate comments | | 2 |
| Include name, ID, and brief introduction in all scripts | | 2 |

Assignment 2 Total:

/40