KIT501 ICT System Fundamentals
Semester 2, 2017

Assignment Two – UNIX Shell Programming

(10% of your KIT501 final result)

Due at 3pm Thursday Week 12, 12 October 2017

There are three (3) shell programming tasks in this assignment. This is an individual assignment. 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 by the KIT501 tutors on `alacritas`. If you write and test your scripts elsewhere it's your responsibility to ensure that your scripts run correctly on the School UNIX system (`alacritas`).

Task A
Write a shell script (to run on the Bourne shell) to copy all the files from the directories named `dir1` and `dir2` into a new directory named `dir3`.

(1). In the beginning of your script you need to check that the directories `dir1` and `dir2` exist under the current directory (if not, your script displays an error message and then exits).

(2). All the three directory names are supplied as arguments, and the new directory `dir3` must not currently exist (ie, the creation of `dir3` is part of the script).

(3). The script first copies all the files from `dir1` into `dir3`.

(4). The script then copies every file from `dir2` to `dir3` subject to these conditions: if the file from `dir2` is not already present in `dir3`, or if the `dir2` file is newer than the same `dir3` file it will overwrite its namesake in `dir3`.

(5). Remove all the temporary files (if any) at the end of your script.

(6). Your script must generate an output similar to the sample output shown in the top box of page 2.

Hint: You should note that the `cp` command without any option does not preserve certain attributes (such as ownership and timestamps).

Your script for this task must be named **`cp2.sh`**. Make sure that your script is user-friendly and follows common sense. The following is a sample output of the script. The $ is the shell prompt.

```
$ ./cp2.sh  dir1  dir2  dir3
These files from dir1 copied into dir3:
a.c  b.c  file1.c

These new file(s) from dir2 copied into dir3:
file2.c

These file(s) from dir2 copied into dir3 and overwrite(s) their
namesakes in dir3:
a.c
```

The following sample outputs verify the above result:

```
$ ls
cp2.sh dir1  dir2  dir3

$ ls -l dir1
total 12
-rw-r--r-- 1 sxu staff  9 Aug  2 12:52 a.c
-rw-r--r-- 1 sxu staff  9 Aug  2 12:52 b.c
-rw-r--r-- 1 sxu staff  9 Aug  2 12:52 file1.c

$ ls -l dir2
total 12
-rw-r--r-- 1 sxu staff  9 Aug  2 12:53 a.c
-rw-r--r-- 1 sxu staff  9 Aug  2 12:51 b.c
-rw-r--r-- 1 sxu staff  9 Aug  2 12:53 file2.c

$ ls -l dir3
total 16
-rw-r--r-- 1 sxu staff  9 Aug  2 12:53 a.c
-rw-r--r-- 1 sxu staff  9 Aug  2 12:52 b.c
-rw-r--r-- 1 sxu staff  9 Aug  2 12:52 file1.c
-rw-r--r-- 1 sxu staff  9 Aug  2 12:53 file2.c
```

(Continued next page)

Task B

Write a shell script (to run on the Bourne shell) that runs an infinite loop to monitor which users on a *denial list* have logged into the UNIX system more than once.

(1). The denial list must be stored in a text file named `user.deny` which lists which users are not allowed multiple logins, specified by one username per line.

(2). In the beginning of the script you need to check that the `user.deny` file exists under the current directory, and if not, your script displays a message to say so and then exits.

(3). Every 3 seconds, the script must display a warning message to report users on the denial list who have logged in multiple times. Display the user's full name instead of his/her username in the warning message.

(4). It's unnecessary for your script to handle the situation where a user logs in and then logs off immediately (i.e. within a 3 second interval).

Your script for this task must be named **`mlog.sh`**. The following is a sample output of the script (It is OK if the script leaves behind temporary files when it is finally interrupted). The $ is the shell prompt.

```
$ ./mlog.sh

No user on the user.deny list has multiple logins

No user on the user.deny list has multiple logins

The user John Smith (on the denial list) has logged in more than
once!

The user Adam Jones (on the denial list) has logged in more than
once!

No user on the user.deny list has multiple logins

No user on the user.deny list has multiple logins

The user David Monks (on the denial list) has logged in more than
once!

The user Nick Andrews (on the denial list) has logged in more than
once!

… … … … … … … … … … … … … … … … …
```

Cut
three files

Task C

Dominion Consulting in Sydney needs a shell script (to run on the Bourne shell) to maintain its employee records file, which contains the following information about each employee: telephone number (8 digits, first digit non-zero), family name (alphabetic characters), first name (alphabetic characters), department number (2 digits), and job title (alphabetic characters). This script should let users **add**, **delete**, **search** for, and **display** specific employee information.

(1). Create a text file named `records` containing the following records with fields delimited by colons:

```
95671660:Jones:Sarah:45:sales manager
93272658:Smith:John:43:technical manager
98781987:Williams:Nick:35:computer officer
99893878:Brown:Sarah:12:electrician
95673456:Couch:David:26:chef
95437869:Anderson:Sarah:19:CEO
```

(2). In the beginning of your script you need to check to see whether the required text file (`records`) actually exists under the current directory (if not, your script displays a message and then exits).

(3). Your script (must be named **`menu.sh`**) will present a menu of operations that a user may choose from. Among other tasks, these operations automate the following four activities:

1. Displaying all current employee records on the screen.
2. Searching for and displaying specific employee record(s) (search all fields, ignoring case)
3. Adding new records to the `records` file.
4. Deleting records from the `records` file.

(4). Your script must produce the following menu:

```
Dominion Consulting Employees Info Main Menu
============================================
1 - Print All Current Records
2 - Search for Specific Record(s)
3 - Add New Records
4 - Delete Records
Q - Quit
```

(5). After a 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.

(6). You must validate the employee details when adding a new record.

(7). A valid family name, first name and job title must be alphabetic characters and/or spaces. The first digit of a valid phone number (of eight digits) must not be zero. Each telephone number must be unique. A valid department number must only contain 2 digits.

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.

```
$ ./menu.sh

Dominion Consulting Employees Info Main Menu
============================================
1 - Print All Current Records
2 - Search for Specific Record(s)
3 - Add New Records
4 - Delete Records
Q - Quit

Your Selection: 1(user input)
95671660:Jones:Sarah:45:sales manager
93272658:Smith:John:43:technical manager
98781987:Williams:Nick:35:computer officer
99893878:Brown:Sarah:12:electrician
95673456:Couch:David:26:chef
95437869:Anderson:Sarah:19:CEO

Press Enter to continue... (user presses Enter here)

Dominion Consulting Employees Info Main Menu
============================================
1 - Print All Current Records
2 - Search for Specific Record(s)
3 - Add New Records
4 - Delete Records
Q - Quit

Your Selection: 5(user input)
Invalid selection

Press Enter to continue... (user presses Enter here)

Dominion Consulting Employees Info Main Menu
============================================
1 - Print All Current Records
2 - Search for Specific Record(s)
3 - Add New Records
4 - Delete Records
```

```
Q - Quit

Your Selection: 2(user input)
Enter keyword: Sarah (user input)
95671660:Jones:Sarah:45:sales manager
99893878:Brown:Sarah:12:electrician
95437869:Anderson:Sarah:19:CEO

Press Enter to continue... (user presses Enter here)

Dominion Consulting Employees Info Main Menu
============================================
1 - Print All Current Records
2 - Search for Specific Record(s)
3 - Add New Records
4 - Delete Records
Q - Quit

Your Selection: 2 (user input)
Enter keyword: Monks (user input)
Monks not found

Press Enter to continue... (user presses Enter here)

Dominion Consulting Employees Info Main Menu
============================================
1 - Print All Current Records
2 - Search for Specific Record(s)
3 - Add New Records
4 - Delete Records
Q - Quit

Your Selection: 2(user input)
Enter keyword: (user simply presses Enter without typing in anything)
Keyword not entered

Press Enter to continue... (user presses Enter here)

Dominion Consulting Employees Info Main Menu
============================================
1 - Print All Current Records
2 - Search for Specific Record(s)
3 - Add New Records
4 - Delete Records
Q - Quit

Your Selection: 3(user input)
Add New Employee Record
Phone Number (xxxxxxxx): (user simply presses Enter without typing in anything)
Phone number not entered
Phone Number (xxxxxxxx): 00123456 (user input)
Invalid phone number
Phone Number (xxxxxxxx): 95437869 (user input)
Phone number exists
Phone Number (xxxxxxxx): 99887766 (user input)

Family Name: abc123 (user input)
Family name can contain only alphabetic characters and spaces
```

```
Family Name: Warren (user input)

Given Name: Tony5 (user input)
Given name can contain only alphabetic characters and spaces
Given Name: Tony (user input)

Department Number: 123 (user input)
A valid department number contains 2 digits
Department Number: 23 (user input)

Job Title: accountant1 (user input)
Job title can contain only alphabetic characters and spaces
Job Title: accountant (user input)

Adding new employee record to the records file …
New record saved (display this only after saving record is successful)

Add another? (y)es or (n)o: n (user input)
```

(*Note: if the user answer is y here then the above procedure is repeated*)

```
Dominion Consulting Employees Info Main Menu
============================================
1 - Print All Current Records
2 - Search for Specific Record(s)
3 - Add New Records
4 - Delete Records
Q - Quit

Your Selection: 4(user input)
Delete Employee Record

Enter a phone number: 05671660 (user input)
Invalid phone number
Enter a phone number: 99999999 (user input)
Phone number not found
Enter a phone number: 95671660 (user input)
95671660:Jones:Sarah:45:sales manager

Confirm deletion: (y)es or (n)o: y (user input)

Record deleted.  (This message is displayed only after this record has been successfully deleted
from the records file)
```

(*If the user answer is n then the main menu is displayed again. Then if the user enters Q or q then the script is terminated.*)

(Continued next page)

For All Your Scripts
You must

- Include your **full 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.

Submitting Your Assignment
You must submit a single folder named `kit501a2` which contains the following five (5) files, electronically to the `kit501submit` folder which has been created for you in your alacritas account:

`cp2.sh, mlog.sh, user.deny, menu.sh, records`
(Please make sure that your `kit501a2` folder only contains the five files as listed here, when you are ready to submit them)

You can use a Windows PC (in one of our school's Windows PC labs) to submit your assignment. This can be done by simply selecting your assignment folder and copy it: In Windows, right-click on your `kit501a2` folder and choose copy from the pop-up context menu. Navigate to the `kit501submit` folder (which is located in your alacritas account [M: drive in the school computer lab]), right-click on it, and choose paste from the pop-up menu. You should look in the `kit501submit` folder to verify that your assignment files are there.

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 School Help Desk.

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

Please note: 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 School 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.

(See next page for the assignment marking scheme)

Appendix: KIT501 Assignment Two Marking Scheme

Script cp2.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**
Include full name, ID, and brief introduction of what the script does		**2**
Appropriate in-line comments		**2**
Check that dir1 and dir2 currently exist and that dir3 does not currently exist.		**2**
Copy all files from dir1 into dir3		**2**
Copy new files from dir2 into dir3		**2**
Copy newer files from dir2 to dir3 and overwrite their namesakes		**2**

Script mlog.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**
Include name, ID, and brief introduction of what the script does		**2**
Appropriate in-line comments		**2**
Check the existence of the file named user.deny		**2**
Correctly set up an infinite loop		**2**
Correctly display "No user on the user.deny list has multiple logins" when this is the case		**2**
Correctly report the full names of the users (on the denial list) with multiple logins		**2**

Script menu.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**
Include name, ID, and brief introduction of what the script does		**2**
Appropriate in-line comments		**2**
Correctly set up a loop to repeatedly display the menu		**2**
Correctly display all records		**2**
Correctly search for specific records		**2**
Correctly add new records		**2**
Validate each filed of a new record when adding new records		**2**
Correctly delete records		**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 (**4**). Somewhat messy but understandable (**2**). Messy (**0**)		**4**

Assignment Total: /50