Assessment Pro Forma

1. Module number	CSN08101
2. Module title	Systems & Services
3. Module leader	Dr S Li
4. Tutor with responsibility for this Assessment	*Jim Jackson*
5. Assessment	*Coursework – see attached*
6. Weighting	*40%*
7. Size and/or time limits for assessment	*6 hours, 30 minutes.*
8. Deadline of submission	*Week 12* *5pm* *26th November 2015*
9. Arrangements for submission	*Script files must be submitted via Moodle.*
10. Assessment Regulations **All assessments are subject to the University Regulations.**	*No exemptions.*
11. The requirements for the assessment	*See Attached.*
12. Special instructions	*None.*
13. Return of work	*Feedback is provided during the demonstration session.*
14. Assessment criteria	*See Attached.*

CSN08112 Computer Systems (Bridging)

Operating Systems Coursework

The purpose of this coursework is to introduce you to the functions performed by an operating system and, in particular, to give you experience of the associated command language and the filestore facilities.

On completion of this coursework you will be able to :

(a) Use the filestore facilities provided by a Linux-based system and particularly, the file attributes and directory structures.
(b) Use the command language associated with a Linux-based system, including some of the commonly used commands and the commonly encountered structures.
(c) Appreciate the strategy associated with the testing of systems software, and the need to build robust products.
(d) Understand and implement 'best practice' housekeeping procedures to allow for user/system misuse or mistakes.

SCENARIO

An on-line news service has amassed a very large number of image files which the business now wishes to archive. At the moment these images are stored on various types of flash memory device.
With no company standard for storing such images, these memory devices have been used by different staff who have created their own sub-directory structures according to their own preference.

Unfortunately they have also used the devices to store other materials including copies and edited versions of the original images.
Therefore, image files may be found in any of the several sub-directories on the flash drive and may even be duplicated in a different folder.
It can be assumed that a duplicate file will always have the same name as the original.

It is also quite likely that the same file name has been used for two completely different images on occasion.

You have been asked to develop a Linux bash shell script which will record the absolute pathnames of any duplicate images files found in the flash drive and save these paths to a text file called duplicates.txt in a photo archive directory.
The script will also reliably copy original images from the flash drive to the archive directory without creating any duplicates. It is also absolutely imperative for the company's business that no image is ever lost by being omitted or over-written in the single photo archive directory.

The script will need to function with flash drives potentially containing tens of thousands of images and should therefore be reasonably efficient in its operation.

SPECIFICATION

The **ph**oto **ar**chiving script must be named "phar" and should be located in the **~/my-applications/bin** directory.

It will be used as:

phar flash_pathname archive_dir_path

Examples:

```
$ phar  /mnt/sdb1  ~/my-documents/photoarch
$ phar  /mnt/sdb1  ../imagesarc
$ phar  sdb1  /root/images/archive
```

The first example above would copy all the relevant image files from **/mnt/sdb1,** and any sub-directories within this directory, to the directory **~/my-documents/photoarch**.

1. The script should be efficient, robust and cope with possible user error.
    - If the photo archive directory does not already exist, the script must create it.
    - If the image source directory does not exist, the script should report an error and terminate.
    - If the user omits either of the two arguments then the script should provide a helpful error message as in:-

      "Usage : phar image_path archive_path"

2. Only original image files with file names in the form IMG_dddd.JPG are to be archived. (where “dddd” can be any 4 digit number).

3. No sub-directories are allowed inside the photo archive directory, only files i.e. it is a flat one-level structure.

4. When copying an image file to the archive directory (e.g. IMG_0001.JPG), if there is already a file with that name in the archive then the second file should be called IMG_0001.JPG.JPG to avoid overwriting the first file, unless the two files are identical in which case the absolute pathname of one is recorded in the *duplicates.txt* file and the file is not copied.

IMPLEMENTATION

Students can implement the required function either as a single script file or by using multiple scripts to achieve the result, provided “bash” shell script programming is used.
For example you might create a “safe copy” script first which copies a single image to a destination directory but only after checking for file name conflicts. Similarly you may wish to copy all the image files to the destination directory before running a second script to check for, and remove, duplicates.

Regardless of how the script(s) are constructed the user must still be able to perform the archive using:-

phar *image_path archive_path*

TESTING

The script functionality should be tested rigorously before submission. The script should work with any appropriate source and destination directories and any reasonable quantity of image files.

Example test image files will be provided for help with testing.
Students can extract an example set of directory and image files from a zipped archive, available from the file repository using a web browser.

SUBMISSION

The submission should include :

An electronic copy of the shell script(s) (documented with internal #comments). Submitted to Moodle

(65%)

A demonstration of the scripts by the students showing all the features working.

(35%)

Students should be prepared to explain their script implementation at the demonstration.
Demonstrations can be carried out in the lab following the Moodle submission.