

# **Coursework Assessment**

**Purpose**

This coursework is intended to help you develop your understanding of shell scripting in both a Windows and Unix environment.

**Organisation**

You should undertake this coursework in groups consisting of two students. (Please note that there are no exceptions to this maximum of TWO, so please do not ask!) You should choose your own group member. You will not be organised into groups by your tutor.

If you do not successfully complete this coursework at the first opportunity and are required to resubmit it at the second opportunity, you may undertake it as an individual exercise if you prefer or if you are unable to find a group member.

You will have an opportunity to gain some initial experience of shell scripting during the lab sessions of the module. However, you will need to spend a considerable amount of time, outside of lab sessions, conducting additional research in preparation for this coursework.

**Grading**

The coursework counts for 50% of the overall mark and it has three components (Two scripts, Coursework report and lab sessions).Lab sessions are marked and counts towards the coursework result. The following table gives the distribution of marks:

|  |  |  |
| --- | --- | --- |
| Components | Coursework mark | Overall mark (Module) |
| Each script (Demo) | 30% | 15% |
| Report (Submission) | 20% | 10% |
| Lab Sessions (Weekly) | 20% | 10% |
| Exam (January) |  | 50% |

**Resources**

The following resources are available to you to help you prepare for this coursework:

* The Knoppix CD, which was given at the first lab session you attend.
* Any PC running Windows 10. Versions of Windows from XP onwards provide support for a number of scripting languages as standard. Additional scripting languages can be installed by downloading them, free of charge, from the Internet.
* There is also an USB-based Knoppix, which you created in the labs using the CD.
* The PCs in the laboratories KD.2.14, KD.2.15 and KD.2.28Erun Windows 10 with Windows Sub-system Linux (Ubuntu) installed and will be available for use outside the scheduled laboratory sessions. Access is provided on a first-come, first-served basis. Please check the timetable on the door of the laboratory for details.

Any queries regarding this coursework should be directed via e-mail or Moodle to the module leader.

**Deliverables**

You are required to program TWO shell scripts, one which executes in a Unix environment and one which executes in a Windows environment.

Your scripts must satisfy the following requirements:

1. They must perform an operating system routine, which is based ona useful system administration function. Some examples of useful system administration functions are given below.
2. It must be possible to run them and hence demonstrate them using the facilities available in the labs used by this module. Alternatively, you may use your own hardware (e.g. personal laptop) and software provided that they are both legal (e.g. licensed) and you are able to bring them to the University for the purposes of demonstration.
3. They may be written in whatever shell scripting language you choose provided that they comply with point 2 above, and that you do not use the same scripting language for both the Unix and Windows scripts. There are a significant number of scripting languages available for Unix including Bash, Perl and Python and for Windows including Powershell, VBScript, JScript and the traditional batch file language. In addition, there are Windows versions of many of the scripting languages that are available for Unix including Perl and Python. Whilst you are free to choose any shell scripting language you like, the module leader RECOMMENDS that you use Bash for your Unix script and a batch file for Windows. These are the simplest scripting languages for beginners.
4. You may use examples of scripts that you find in printed documentation and on the Web as the starting point for your own scripts. If you do so, you must include a copy of the original script in your submitted documentation (failure to do so will lead to a mark of zero and charges of plagiarism) along with details of its source. You must also be in a position, when you demonstrate your script to your tutor, to explain how you have enhanced/changed the original script.

You will be required to demonstrate your work towards the end of the module at a time to be specified by the module leader via Moodle.

**Report Submission**

Before the demonstration session, you will need to submit your report online by 21.00 on Monday16th December 2019.

You **must** submit the report in Microsoft word format to the Turnitin submission link on Moodle.

Failure to fulfil the submission requirements above will lead to you being awarded a mark of zero for this coursework.

Your report should include:

* Copies of your two scripts (for Windows and Unix), with **annotations**
* Copies of any and all scripts which you used as a basis for your own with details of how you have changed the originals.
* An**individual evaluation** (between one to two pages of A4) of your contribution to the work of the group including:
	+ The total number of hours spent working on the coursework
	+ An assessment of whether the amount of time that you spent on the coursework was adequate
	+ An assessment of the extent to which you have satisfied the requirements of this coursework
	+ A discussion of what you would do differently if you were given a second opportunity to undertake this coursework.
* References section including a list of in-text references.
* A bibliography containing details of ALL sources of information that you have used whilst preparing this coursework.

***Each group submits one report only, but each member of the group has towrite an individual evaluation at the end of the report.***

***You must indicate your student number at the beginning your evaluation.***

**Examples of useful system administration scripts**

The list below contains examples of useful system administration scripts. It is not meant to be an exhaustive list and you are encouraged to think up your ideas – solving your own problems (and coding them) is often more interesting than trying to solve someone else’s problem! In addition to the list below, you will be given access to examples of useful system administration scripts (some of which were written by previous students) via Moodle.

* Creation of user accounts by reading users’ names from a file
* Deletion/amendment of users’ accounts
* Backing up of important system files
* Archiving of users’ files
* Deletion of temporary files
* Easy-to-use menu-driven front end to a number of system commands
* Displaying of important system information e.g. CPU, memory and disk usage
* Comparing files and displaying differences
* Displaying directory information e.g. number of files, subdirectories and total size of a specified directory

**Assessment criteria**

The formal assessment of this coursework is based upon a combination of the documentation and demonstration. If you fail to submit the documentation by the deadline, you will not be allowed to demonstrate your work and will be awarded a mark of zero for this coursework.

During the demonstration, you will be asked to show your tutors both of the shell scripts that you have written. The scripts must be identical to those in your documentation. Alternatives will not be accepted. The purpose of the demonstration is to ensure that your documentation contains the results of your own work and that you fully understand its contents.

***Each script will be marked against the following criteria:***

*Level of usefulness*

* Does the script perform some useful and realistic system administration function?
* Does it automate tasks which would otherwise involve running a series of separate commands/programs?

*Level of complexity*

* Is the script simple or does it demonstrate complex functionality?
* Does it demonstrate a good understanding of the features and capabilities of the scripting language?
* Was the script written from scratch or was it adapted from an example found?
* If the latter, to what extent was it adapted?

*Level of functionality*

* Is the script fully functional (e.g. it is fully working and does exactly what you intended), is it partially working or is it non-functional?

*Level of understanding*

* Can you explain fully and clearly the code contained within your script and its purpose?
* Regardless of the assessment criteria above, you will not be given credit for a script unless you can adequately explain it.

***The report that you submit will be marked against the following criteria:***

*Quality of evaluation*

* Do the individual evaluations address all the necessary points?
* How well are the points addressed?
* Is the evaluation realistic?

The table below gives you an indication of the relationship between the mark awarded for this coursework and the level of your achievement. Please note that the actual mark awarded will depend upon the quality of your evaluation as well as the quality of your scripts.

|  |  |
| --- | --- |
| Level of achievement | Mark awarded |
| Your scripts were written by someone else. You have properly acknowledged the source of your scripts, have been able to demonstrate them working and you can explain their function and operation fully. | 0 - 39% |
| You have taken someone else’s scripts but have made some basic enhancements. You have properly acknowledged the source of your scripts, have been able to demonstrate them working and you can explain their function and operation fully. | 40 - 49% |
| Either you have taken someone else’s scripts but have made significant enhancements. You have properly acknowledged the source of your scripts, have been able to demonstrate them working and you can explain their function and operation fully.Or your scripts are original pieces of work. They are non-trivial, are at least partially functional and you are able to explain their operation fully. | 50 – 59% |
| Your scripts are original pieces of work. They are moderately complex, mostly functional and you are able to explain their operation fully | 60 – 69% |
| Your scripts are original pieces of work. They are complex, fully functional and you are able to explain their operation fully. | 70%+ |

**Getting started**

You have a total of 60 hours to complete this coursework. Your fellow group member also has 60 hours. How you prepare this coursework and divide up your time and responsibilities is entirely a matter for you and your partner to agree. However, you might find the steps outlined below useful:

1. Make sure that you carry out the lab exercises for Sessions 1 to 10. You will find it difficult to write scripts if you do not understand the syntax of your chosen scripting language and the system commands that your scripts incorporate. See the recommended reading for Sessions 1 – 10 for ways in which you can extend your knowledge beyond the material covered in the lab sessions.

*(Suggested time: 10 hours)*

1. Download examples of scripts. There are huge numbers of them freely available on the Internet. (Try typing batch file examples in Google!). Try running the downloaded scripts and work out what they do and how they do it. (A note of caution: Please check the scripts before running them in case they trash your disk or send your personal data to someone on the Internet!) Reading other people’s code is often a good way to teach yourself how (and how not) to program.

*(Suggested time: 20 hours)*

1. Code, test and debug your scripts. Be sure to test your scripts on the machine that you will you use to demonstrate them. Do not assume that the demonstration environment is identical to the development environment.

*(Suggested time: 20 hours)*

1. Prepare your documentation. The documentation should be of a high standard. Do not forget that each group member has to submit an individual report. The only similarity between the reports will be the scripts, which means that the report will need to be the individual students own piece of work.

*(Suggested time: 5 hours)*

1. Prepare yourself for the demonstration. Make sure that you can install your scripts and run them on your demonstration machine. Make sure that you fully understand both the function and operation of your scripts. You will be asked questions during the demonstration.

*(Suggested time: 5 hours)*