KF4005: Operating systems fundamentals
Course Work 2017-18

Module tutor: David Kendall

1 Dates and mechanisms for assessment submission and feedback

Date of hand out to students: 27th February 2018

Mechanism to be used to disseminate to students: eLP

Date and Time of Submission by Student:

- Assessed lab: In your usual lab session in the week commencing 23rd April 2018
- Report: 23.59 on 3rd May 2018

Mechanism for Submission of Work by Student:

- The assessed lab deliverables should be submitted via eLP using the assignment link as follows: `Assessment->Assessed Lab`.
- The report should be submitted via eLP using the Turnitin link as follows: `Assessment->Report`.

Date by which Work, Feedback and Marks will be returned to Students:
Within 20 working days of report submission date.

Mechanism(s) for return of assignment work, feedback and marks to students:
email and appointment on request

2 Assignment brief

2.1 Overview

2.2 Assessed lab session (Weighting: 50%)

You will be required to attend an assessed lab session in week 12. You must attend in the lab session that is scheduled in your personal timetable, as provided by the University. This should be regarded as a formal University requirement. Failure to attend at the correct time may result in a mark of zero for the assessed lab component of the assignment.

At the completion of the assessed lab session you must upload your deliverables to the eLP using the link `Assessment->Assessed Lab`. Failure to upload your deliverables correctly may result in a mark of zero for the assessed lab component of the assignment.

Mounting and gathering information from a disk image

Write a `bash` shell script, called `disk_image_analysis` that performs the following actions:

1. Given the name of a disk image file and the name of a directory to act as a mount point, mount the specified disk image to the mount point. The script should take reasonable precautions against failure. For example, it should ensure that

   (a) the disk image file exists

   (b) either the directory for the mount point exists and is empty, or is created by the script if it does not exist

   Your script should allow the user to bypass the mounting of the disk image file. This functionality should be accessible through the use of a command line option.

2. Extract file metadata from the `/bin` and `/sbin` directories and output the information as a list of lines of semi-colon separated data. This makes it easy to include the data into a database for later analysis. The metadata required is as follows: last access date and time; last modification date and time; creation date and time; permissions; user and group ids; file size and file name. Your script should allow the user to specify different directories

for analysis, using a command line option. Again the script should take reasonable precautions against failure.

3. Create an SQL database, called `KF4005AL`, on your host machine.
4. Create a table, called `filedata`, in the `KF4005AL` database. The table schema for `filedata` should be suitable for storing the file metadata extracted from the disk image file.
5. Load the file metadata into the `filedata` table.
6. Use an SQL SELECT command on the `filedata` table to output the file name, file permissions, user id, group id and last access date of all records, in descending order of last access date.

Notes

1. A disk image file will be made available so that you can test your script during development.

2. A *different* disk image file will be provided for use at the assessed lab session.

3. At the assessed lab session, you will be required to enter some simple commands to prepare and run your script and to clean up after yourself. You will be assessed on your use of these commands too.

4. An example of what will be required at the assessed lab will be provided in advance of the lab so that you can practice (see http://www.hesabu.net/kf4005/L11.html).

5. **Important — Deliverables** Before leaving your assessed lab session you must ensure that the assessed deliverables have been prepared exactly as specified below and uploaded to Blackboard (eLP) using the link `Assessment -> Assessed Lab`.

   - Create a new directory called `deliverables`.
   - Add the following files to this directory
     - Your disk image analysis script, called `disk_image_analysis.sh`
     - The text file of extracted file metadata that was loaded into your `filedata` table. This text file should be called `filedata.txt`.

  - The text file containing the output from your SQL SELECT statement. This text file should be called `sqlout.txt`.
  - The edited history of the bash commands used in practical exercises. This text file should be called `bashinput.txt`.
  - The edited output of the bash commands used in the practical exercises. This text file should be called `bashoutput.txt`.

- Use the `tar` command to create a gzipped, tar file of the `deliverables` directory. The gzipped, tar file name should be called `deliverables.tgz`.
- Upload `deliverables.tgz` to Blackboard using the link `Assessment -> Assessed Lab`.

6. **Marking** Marks will be awarded for the assessed lab as follows

- Practical exercises involving the use of `bash`, including set up, preparation, running and clean up after the disk image analysis *(20 marks)*
- Satisfaction of functional requirements of the disk image analysis script *(10 marks)*
- Quality of the disk image analysis script as judged by the following criteria:
  - Use of functions - functional decomposition
  - Use of meaningful variable and function names
  - Use of layout that aids the understanding of the reader
  - Use of bash shell script idioms: parameter expansion, command substitution etc.
  - Use of SQL

  (10 marks)

2.3 OS theory and concepts (Weighting: 50%)

You are required to submit a document that satisfies the requirements below. Number your answers carefully, using precisely the numbering scheme used in the specification of requirements below. Failure to number your answers correctly may lead to loss of marks.

1. *Process management*

   A typical behaviour of a user of the Unix operating system is to enter a command such as `ps -eF | grep named` into a terminal emulator. This command is interpreted by the shell to create two processes that communicate via a pipe to execute the required programs.

   (a) Explain briefly what is meant by the following: *terminal emulator*, *shell*, *program*, *process*, *pipe.*

   (5 marks)

   (b) Describe the mechanisms of process creation and program execution that might be used in the execution of the command `ps -eF | grep named`. Draw a diagram to illustrate the relationship between the processes that are created.

   (5 marks)

   (c) Explain how a *pipe* can be created and used to allow communication between the processes. Explain clearly which processes can read and write to/from the pipe. Draw a diagram to illustrate your answer.

   (5 marks)

   (d) Describe briefly what happens to each process after it has completed its execution. How does a parent process discover the termination status of its children? Consider both successful and unsuccessful termination.

   (5 marks)

2. *Memory management*

   It is important to illustrate your answers to this question with diagrams, when appropriate.

   (a) Describe the typical *memory model* of a user process (task), running under the control of an operating system.

   (5 marks)

(b) Discuss the importance of *memory protection* in ensuring the reliability of a multi-tasking system. Describe a simple *hardware* mechanism that supports the implementation of memory protection.

(5 marks)

(c) What is *address binding* and when can it be done? Make sure that your answer clearly distinguishes between *symbolic*, *relocatable* and *absolute* addresses.

(5 marks)

(d) Explain the distinction between *logical* and *physical* addresses in the context of OS memory management. Describe a simple *hardware* mechanism that supports the mapping of logical to physical addresses.

(5 marks)

3 Further information

Learning Outcomes assessed in this assessment:

Knowledge & Understanding:

1. Demonstrate knowledge of the fundamentals of an operating system, including its architecture and the implementation of its services.

Intellectual / Professional skills & abilities:

2. Apply scripting languages and tools to the monitoring, investigation and administration of an operating system and its resources.

3. Design and utilise simple SQL queries.

Personal Values and Attributes:

4. Communicate the results of work / study accurately and reliably.

Assessment Criteria/Mark Scheme: The coursework consists of

1. an assessed laboratory session leading to the submission of a specified set of deliverables (50%)
2. a report on OS theory and concepts (50%)

More detailed marks allocation is provided in the assignment brief.

Referencing Style: Harvard

Expected size of the submission: Your report should be about 1200 words in total. There is no fixed penalty for exceeding this limit but unnecessary verbosity, irrelevance and 'padding' make it difficult for the marker to identify relevant material and may lead to some loss of marks.

Assignment weighting: 100%

Academic Integrity Statement: You must adhere to the university regulations on academic conduct. Formal inquiry proceedings will be instigated if there is any suspicion of plagiarism or any other form of misconduct in your work. Refer to the University's Assessment Regulations for Northumbria Awards if you are unclear as to the meaning of these terms. The latest copy is available on the University website.

Failure to submit: The University requires all students to submit assessed coursework by the deadline stated in the assessment brief. Where coursework is submitted without approval after the published hand-in deadline, penalties will be applied as defined in the University Policy on the Late Submission of Work.
https://www.northumbria.ac.uk/static/5007/arpdf/lateappr

Anonymous marking: University policy requires that work be marked anonymously. In order to facilitate this we request that only your student number is included on work submitted for summative assessment.