Institute of Technology Blanchardstown
Institiúid Teicneolaíochta Baile Bhlainséir

INSTITUTE OF TECHNOLOGY BLANCHARDSTOWN

Year	Year 2
Semester	Semester 1
Date of Examination	Tuesday 10 January 2012
Time of Examination	12.30pm – 2.30pm

Prog Code	BN002	**Prog Title**	Higher Certificate in Science in Computing in information Technology	**Module Code**	COMP H2028
Prog Code	BN013	**Prog Title**	Bachelor of Science in Computing in Information Technology	**Module Code**	COMP H2028
Prog Code	BN104	**Prog Title**	Bachelor of Science (Honours) in Computing	**Module Code**	COMP H2028

Module Title	Operating Systems (Client)

Internal Examiner(s): **Dr. Kevin Farrell**

External Examiner(s): **Dr. Richard Studdert**

Instructions to candidates:

1. **Question One in Section A is COMPULSORY.**
2. **Candidates should attempt ALL parts of Question One in Section A**
3. **Candidates should attempt ONE question from Section B, and ONE question from Section C**
4. **This paper is worth 100 marks.**
5. **Question One is worth 40 marks, and all other questions are worth 30 marks each.**

DO NOT TURN OVER THIS PAGE UNTIL YOU ARE TOLD TO DO SO

SECTION A: COMPULSORY QUESTION

Question 1: Answer *ALL* parts of this question *(4 marks for each part)*

a) Distinguish between the concept of a *soft real-time operating system* and a *hard real-time operating system.*

b) With reference to memory management, consider a paging system with the *page table* stored in memory.

   i. If a memory reference takes **200 ns**, how long does a paged memory reference take?

   ii. Consider the addition of a *translation look-aside buffer (TLB).* If **85%** of all page-table references are found in the *TLB*, and if finding a page-table entry in the *TLB* takes **zero time** (if the entry is there), what is the *effective access time (EAT)*?

c) Explain the concept of *multiprogramming* in an operating system.

d) Briefly describe **two** possible rules of thumb when determining the *time quantum* in a *round robin* scheduling scheme.

e) List **four** objectives of a good *process scheduling policy* for an operating system.

f) Describe the difference between a *process* and a *thread.* Give **one** advantage of threads over processes.

Question 1 is continued on the next page

g) Why is it considered bad practice to log directly into a Linux GUI-based system as *root*?

h) Consider the situation where a user downloads the source-code *tarball* of an application they wish to install on their Linux system. Using the `tar` command, the user has extracted the files from the *tarball* into their home directory, and now wishes to install the application. List the steps that the user must type to *configure*, *compile* and *install* the software on their system. For each step, provide the command prompt symbol to indicate whether an ordinary user or root should type the command.

i) Consider the situation where a user has just purchased a **third** *Serial-ATA* hard disk drive for their Linux system. They install the disk, boot up their computer, log in as an ordinary user and run a *konsole* terminal window. Provide the commands, required by the user, to run a command-line partitioning tool which will allow them to start the procedure of partitioning the new disk. (You **do not** have to provide the partitioning tool options for actually creating partitions).

j) In relation to Linux, distinguish between an *absolute pathname* and a *relative pathname*.

(40 Marks Total)

SECTION B: ANSWER QUESTION 2 or QUESTION 3

Question 2

a) Distinguish between *user mode* and *kernel mode* in an operating system.

(4 marks)

b) Explain by what mechanism, and for what reasons, the switch from *user mode* to *kernel mode* takes place.

(8 marks)

c) Describe, with the aid of a diagram, the *UNIX System V Release 4* Process State Model, indicating the transitions between states, and the reasons for those transitions.

(14 marks)

d) Briefly explain why the *UNIX System V Release 4* operating system is not suitable for *real-time* processing.

(4 marks)

(30 Marks Total)

Question 3

a) In relation to memory management, explain the following terms:
logical address, physical address, process loading, swapping

(4 marks)

b) Describe a simple method which allows memory to be addressed as a set of *pages* each containing a fixed number of *displacements.*

(4 marks)

c) Explain the term *page replacement*, and why it is needed.

(4 marks)

d) Describe any **two** of the following page replacement algorithms: *Least Recently Used (LRU)*, *Not Recently Used (NRU)*, *First-In First-Out (FIFO)* and *Clock.*

(8 marks)

e) (i) In relation to *virtual memory* based on paging, explain the constraints regarding storing the *page table* of every process in *real memory.*

(6 marks)

(ii) In relation to *virtual memory* based on *paging*, briefly explain how the location of the *page table* of the currently running process is located.

(4 marks)

(30 Marks Total)

SECTION C: ANSWER QUESTION 4 or QUESTION 5

Question 4

a) Consider the situation of a Linux System Administrator who has forgotten the *root* password. Propose a solution to this problem.

(4 marks)

b) Explain in detail, the purpose and contents of the `/etc/group` file.

(8 marks)

c) With reference to the Linux Operating System, describe the `/etc/passwd` and the `/etc/shadow` files under the following headings:

i. Their purpose.
ii. The type of information they contain.
iii. The differences between them.

(6 marks)

d) How many fields are used on each line in both the `/etc/passwd` and the `/etc/shadow` files? Describe the purpose of the different fields in each case.

(12 marks)

(30 Marks Total)

Question 5

a) Distinguish between *automatic* and *manual* booting in Linux.

(2 marks)

b) List and describe the **six** distinct phases involved in bootstrapping a Linux Operating System.

(12 marks)

c) Briefly describe **six** tasks, which are generally performed by *start-up scripts* during the bootstrapping of a Linux Operating System.

(6 marks)

d) With reference to Linux:

(i) Explain the concept of a *run-level*.

(2 marks)

(ii) State how many *run-levels* are supported in Linux, and how many are actually defined.

(2 marks)

(iii) Describe each of the defined *run-levels*.

(6 marks)

(30 Marks Total)