



UNIVERSITY of LIMERICK

O L L S C O I L L U I M N I G H

COLLEGE of INFORMATICS and ELECTRONICS

Department of Computer Science

and Information Systems

End-of-Semester Assessment Paper

Academic Year: **2005/06**

Module Title: **Operating Systems Overview**

Duration of Exam: **2½ Hours**

Lecturer: **J Sturdy**

Semester:

Module Code:

Percent of Total Marks:

Paper marked out of :

Semester 1

CS4145 and CS4023

100

100

Instructions to Candidates

- Answer any 5 questions.
- Each question is worth 20 marks.

1. Explain the concept of “process address space”, and the differences between processes and threads. When would you use processes, and when would you use threads, in designing an application?

What are the conceptual differences between paging and segmentation? You should mention any ways that they have an effect on writing programs to run on systems using them. Does a segmented architecture have any particular implications for a multi-tasking environment?

Explain what “swapping” is, and when it occurs, both in the loose sense of the word, and in an older sense that is now little-used. Why is the old type of swapping less useful than it was in more primitive systems?

2. What interrupt-driven operating system action links memory and concurrency? Under what circumstances does it occur? What are the different kinds of this event, and how does the operating system handle each of them, and what data structures does it use and modify in doing so? Your answer should include explanations of the flag bits in the data structures concerned.

Which system calls (functions within the operating system that can be called from ordinary programs) are made more efficient by things done using this mechanism, and how? Explain the actions done by these system calls, both with and without this assistance.

3. Explain what software is executed when any kind of interrupt occurs. You should mention its effect on data structures such as process lists and memory management information.

Some kinds of interrupt may be propagated to user processes in the form of signals. What are signals, and what are the operations that a process can do concerning them, either as they happen, or in advance? How could you use signals in an application?

4. What are files? How are they identified on a Linux file system?

What are directories? How are they related to files?

What are file systems? What factors do you have to consider in setting up a filing system on a storage device?

Explain how these three concepts fit together. Your answer should include musings on the following questions:

- Is it possible for a file to contain another file?
- Is it possible for a directory to contain another directory?
- Are directories a kind of file?
- Is it possible for a file system to contain another file system?
- Does reflecting on “Are directories a kind of file?” affect your answer to “Is it possible for a directory to contain another directory?”?

Why do operating systems such as Linux and Unix present hardware devices (such as USB interfaces, the keyboard, disk drives, etc) as part of the file system?

5. How does “Buffered input/output” make the operation of a computer system more efficient?

Explain how you would implement input/output buffering in a concurrent Operating System, explaining the data structures and algorithms used. You should take into account the behaviour of physical devices such as disk drives and removable flash drives.

Are there any circumstances in which it is better not to buffer I/O, or in which is it better for a program to take specific control of the buffering? If so, what are they, and what control is needed? Give arguments for or against (or, for a really balanced answer, both for and against) the view that if the operating system were doing its job properly, this would not be needed.

6. Some operating systems are provided with scripting languages, and use of such languages is often regarded as part of “systems programming”. Are the scripting languages really part of the operating system? If so, why, and if not so, why are they often treated as if they are?

Explain how operating systems may be configured and customized for the uses and users of specific installations (such as web servers, mail servers, and personal desktops). You should mention what kinds of things can be configured, how the configuration information is held, how it is applied, and two approaches to how it may be changed, with arguments for and against each approach, as seen both by novice or casual users, and by “industrial-strength” users such as IT professionals.

7. Although email is an application (or group of applications), it is often treated as a facility provided by the operating system.

List, with brief notes, the main arguments for and against the operating system handling various aspects of email. *(There is no single right answer to this part of the question. It will be marked in terms of the understanding of the topic that you display in your arguments for or against the statement.)*

List and describe the programs, processes, files and protocols involved in processing email.

8. Why is it important that only operating system code can be executed when the CPU is in supervisor mode? What is the mechanism that enforces this?

Explain the concepts of user-IDs, and how they apply to files and to processes. How does their use by processes determine their use by filing systems, and vice versa?

What is a “root user-ID”? Is it necessary to have such a user, or are there alternatives? Is it desirable to have one, or not to have one, or does this depend on the circumstances? *(There is no single right answer to the last part of this question. It will be marked in terms of the understanding of the topic that you display in your arguments for or against the statement.)*