The University Of Sheffield.

Data Provided: None

DEPARTMENT OF ELECTRONIC AND ELECTRICAL ENGINEERING

Autumn Semester 2008-2009 (2 hours)

Operating Systems 6

Answer **THREE** questions. **No marks will be awarded for solutions to a fourth question.** Solutions will be considered in the order that they are presented in the answer book. Trial answers will be ignored if they are clearly crossed out. **The numbers given after each section of a question indicate the relative weighting of that section.**

1. Why is it necessary for a multi-programmed operating system to provide facilities for inter-process communication (IPC)? **(4)**

How can a simple file model be generalised to pass messages between processes in a pipe model? What is the fundamental limitation of IPC by pipes? **(6)**

Many operating systems provide memory-mapped files as a possible IPC mechanism (as well as other IPC methods). What are the advantages and disadvantages of memory-mapped files for IPC? **(5)**

For transferring data between processes, which mechanism would you expect to be faster: pipes or memory-mapped files? Justify your answer. **(1)**

In addition to pipes and memory-mapped files, the UNIX operating system provides *signals* as a low-level IPC mechanism. Explain how two user processes can use signals to communicate. What are the limitations of using signals for IPC? **(4)**

2. Describe the typical multi-thread library/API function to create a thread, paying particular attention to the parameters and their meaning **(4)**

Multi-threading systems are implemented at either user level of at kernel level. What does this mean? What are the advantages and disadvantages of each implementation approach? **(4)**

Does the POSIX `pthreads` package use kernel-level or user-level thread management? **(2)**

In a user-level multi-threading library, is there one stack per process, or one stack per thread? Explain your answer. **(2)**

In a kernel-level multi-threading system, there are two processes, A and B. Each process has two threads, A1 and A2, and B1 and B2, respectively. If thread A1 is currently running, what considerations need to be taken into account by the kernel's thread scheduler when deciding to switch to thread A2 or to one of the threads in process B (B1 or B2)? How can this be addressed in practice? **(4)**

In some older user-level multi-threading libraries, switching between threads was co- **(4)**

operative, even though the underlying kernel was fully pre-emptive. What does this mean and what are the programming implications of using such a library? Can you identify the major flaw in such a system?

3. **a.** Outline the principal characteristics of deadlock. **(4)**

A set of four processes, $\{p_1, p_2, p_3, p_4\}$ share three reusable resources, $\{R_1, R_2, R_3\}$. There are 3 instances of R_1, two instances of R_2 and two instances of R_3.

- p_1 holds one instance of R_1 and is requesting one instance of R_2
- p_2 holds two instances of R_2, and is requesting one instance of R_1 and one instance of R_3
- p_3 holds one instance of R_1 and is requesting one instance of R_2
- p_4 holds two instances of R_3 and is requesting one instance of R_1

Determine which, if any, of the four processes are deadlocked. Explain each logical deduction. **(8)**

b. Describe the round robin scheduling algorithm with multiple level queues. What happens if the duration of the timeslice is increased to an arbitrarily large duration? **(4)**

c. Describe the hardware mechanism by which the process scheduler of a multi-programmed operating system is invoked. How can this mechanism be used to vary relative process priorities? **(4)**

4. **a.** Outline how a virtual address is translated into a physical address in a basic virtual memory system. What is the principal performance penalty of this scheme compared to processes directly accessing physical memory? How can this disadvantage be reduced? **(8)**

In practice, many operating systems use multiple-level page tables where a Level 1 page table contains the addresses of Level 2 page tables which contain the actual address mappings. Typically, for a 32-bit (4GB) virtual address space, the Level 1 table contains 4096 entries (pointers) to Level 2 tables, each of which contain 1024 entries. What is the advantage of this multi-level page translation method? Apart from added complexity, what do you expect to be the greatest disadvantage? **(4)**

b. The least recently used (LRU) algorithm is the most popular page replacement algorithm in virtual memory systems. Why is an exact implementation of the LRU algorithm impractical? Describe a simple approximation to the LRU page replacement algorithm which has acceptable performance. **(4)**

c. In connection with virtual memory, what is a ‘dirty’ (or ‘write’) bit used for? How can this be used in a page replacement algorithm to improve the efficiency of a virtual memory system? **(4)**

PIR/NLS