Data Provided: None

The University Of Sheffield.

DEPARTMENT OF ELECTRONIC AND ELECTRICAL ENGINEERING

Autumn Semester 2011-12 (2.0 hours)

EEE6032 Operating Systems

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. a. **i)** Describe the types of exceptions commonly used in an operating system. What is the use of each type of exception? **(3)**

ii) Describe the mechanism by which a user process can execute a system call. In particular, explain the way in which a user process can pass parameters to a system call. **(4)**

iii) It is often said that the kernel must be "paranoid" about parameters passed to a system call (*i.e.* it must implement very extensive checking). Why is this so? **(3)**

b. Explain the operation of the *test-and-set* instruction. What is this instruction typically used for? **(5)**

c. In UNIX, describe how a *signal* is transmitted from one process to another. What is the role of the kernel in the delivery of a signal? How does the receiving process respond to the arrival of a signal? **(5)**

2. **a.** In the context of a real-time operating system, what is *priority inversion*? Why is it undesirable? How can priority inversion be prevented? **(5)**

b. Although deadlocking is a well-known problem, *livelocking* is also possible. What is livelocking and how can it be detected? **(5)**

c. Describe the *round-robin scheduling algorithm with multiple priority queues*. For what reasons would a process typically be promoted to higher queue, or relegated to a lower priority queue? **(5)**

d. Descibe an efficient mechanism by which an operating system manages processes which are blocked waiting on a semaphore. **(5)**

3. **a.** What is the dilemma surrounding the implementation of a threading system in a multi-kernel architecture operating system? **(4)**

b. Explain how two independent processes can communicate via shared memory. **(4)**

c. *Record locking* is a mechanism typically used in database systems whereby only one process can obtain the write-lock on a data file. Explain how record locking can also be used for process synchronisation. **(4)**

d. A program needs to read and independently perform calculations on data from two sources, *X* and *Y*. The times at which these data arrive are unpredictable. Why might it be desirable to split the program into two processes, one to handle only data from *X* and the other to handle only data from *Y*, rather than implementing a single process to handle both data streams. **(8)**

4. **a.** One memory allocation strategy that was used before virtual memory was the *buddy* system. Suppose a block of k bytes was requested by a program. This request was rounded up to the nearest value 2^n such that $k \geq 2^n$ for integer n. Then, if a block of memory of exactly 2^n was available, it was allocated. Otherwise, the largest free memory block $> 2^n$ was examined and split into two equal-sized *buddies*. If the buddies were larger than 2^n then one of the buddies was again split into two and the process repeated until finally a buddy was found which was $\geq 2^n$ bytes but could not be split further. This block was then allocated to fulfil the original request. What is the advantage of the buddy system for allocating memory? **(8)**

b. What is a virtual file system and what is its major advantage? **(4)**

c. **i)** What fundamental problem is a *journaling file system* designed to overcome? Outline the operation of a *journaling file system*. **(4)**

ii) What is the main disadvantage of a journaling file system? How can this be reduced and what is the trade-off involved? **(4)**

PIR/NLS