UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR09047 OPERATING SYSTEMS

Monday 7$^{\underline{\text{th}}}$ August 2017

14:30 to 16:30

INSTRUCTIONS TO CANDIDATES

Answer any TWO of the three questions. If more than two questions are answered, only QUESTION 1 and QUESTION 2 will be marked.

All questions carry equal weight.

CALCULATORS MAY NOT BE USED IN THIS EXAMINATION

Year 3 Courses

Convener: C. Stirling
External Examiners: A. Cohn, A. Donaldson, S. Kalvala

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. (a) When servicing requests for data on a disk, the simplest thing to do is to service the request queue in the order of arrival, moving the head appropriately. If the current position of the head is known, the more sophisticated algorithms known by the acronyms SSTF, C-SCAN, C-LOOK can be used.

   i. Describe briefly each of the three mentioned algorithms. [*6 marks*]

   ii. Assume a slow single-platter disk with the following characteristics. It has 200 tracks (number 1 at the inside, 200 at the outside). Assume that the time taken to move the head from standstill at track m to standstill at track $m + n$ (or $m - n$) is approximately $(2 + n/10)$ ms. At the current time, the head is stationary at track 99, after moving there from an inner track. The disk request queue contains requests for tracks 54 (front), 57, 38, 89, 159, 37, 160, 25 and 183 (back).

      A. For each of the three mentioned algorithms, show the sequence of moves made to service the request queue, and calculate approximately the time taken to service the whole queue. Ignore the time taken to read the data from the track. [*9 marks*]

      B. What does this exercise suggest to you about the relative merits of disk scheduling algorithms? How would you attempt to gain more confidence in your suggestion? [*4 marks*]

   iii. Suppose that a program is being used to compare two files block by block, which happen to be located at the extreme inside and outside of the disk respectively. Does the disk scheduling algorithm matter? What if the OS does read-ahead? [*6 marks*]

2. (a) List, with one sentence definitions, some of the criteria that might be used to evaluate process scheduling algorithms. [*5 marks*]

   (b) What is a pre-emptive process scheduler? Describe the first come first served, shortest job first, round-robin (RR) and feedback/priority queue approaches. [*6 marks*]

   (c) Suppose that the usual description of the RR algorithm is modified so that a single process can have two (or more) independent entries in the algorithm's data structures, so that a process can be doubly, triply, etc. scheduled.

      i. What is the effect of having two entries for the same process? Does it depend on how they are placed? [*3 marks*]

      ii. Under what circumstances would you consider it appropriate to give a process multiple entries? [*3 marks*]

   (d) In a scheduling system known as preemptive priority scheduling, the priority of each process is a (positive or negative) number that changes dynamically, at a rate α when it is on the ready queue, and at a rate β when it is actually running. Processes are initialized with priority 0, and reset to priority 0 when they return from a blocked state. (Positive α means increasing priority.) What happens if the algorithm is run with the following settings? Explain your answers.

      i. $\beta > \alpha > 0$ [*4 marks*]

      ii. $\beta < \alpha < 0$ [*4 marks*]

3. (a) Give brief (one sentence) definitions of the following key terms in memory management: virtual memory, page table, TLB, secondary storage. [*4 marks*]

   (b) What happens to virtual memory when a Unix process invokes fork()? [*4 marks*]

   (c) Define the mutual exclusion problem, and give the standard criteria to be satisfied by a successful solution. [*7 marks*]

   (d) A semaphore is a commonly used abstraction for mutual exclusion. Explain how a basic binary semaphore is used, specifying the methods it provides to its users. [*4 marks*]

   (e) Show how to implement a binary semaphore using an atomic test-and-set primitive. Use either clear English, C or pseudo-code . [*6 marks*]