UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR09047 OPERATING SYSTEMS

Monday 13 th August 2018

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: S.Rogers, A. Donaldson, S. Kalvala

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. (a) Briefly explain what thrashing is. [*2 marks*]

   (b) Briefly explain the two types of fragmentations in memory allocation. [*4 marks*]

   (c) Explain the working mechanism of Least Recently Used (LRU) Clock (also known as Second Chance) page replacement algorithm. [*4 marks*]

   (d) If a virtual address is 27 bits long, what is the size of the space that can be addressed? If a 64TB (terabytes) address space is desired, at least how long is a virtual address? [*2 marks*]

   (e) Assume a single-level page table where each page table entry occupies 4 bytes. A virtual address is 32 bits long and a page comprises 4KB (212 bytes). Would the number of page table entries in the page table be determined by the number of pages being used by a process? Explain why. [*4 marks*]

   (f) Why are multi-level page tables often used instead of ordinary (single-level) page tables? What is the added cost associated with using multi-level page tables? [*4 marks*]

   (g) Suppose there is a machine with 32-bit addresses and a two-level page table (in memory) such that the first 10 bits of an address is an index into the first level page table and the next 10 bits are an index into a second level page table. Suppose also that each entry in the page tables is 4 bytes long. How much space is occupied in memory by the page tables for a process that has 64MB of actual virtual address space allocated contiguously. Briefly show your work. [*5 marks*]

2. (a) Briefly explain what a superblock is. [*2 marks*]

   (b) Provide one advantage and one disadvantage of file allocation based on i-nodes. [*4 marks*]

   (c) Suppose you have a file system where the block size is 2KB, a disk address is 32 bits, and an i-node contains the disk addresses (pointers) of: (1) the first 12 blocks of a file, (2) a single indirect block, (3) a double indirect block and (4) a triple indirect block. Note that in answering the following questions, you do not need to simplify arithmetic expressions.

      i. What is the largest file that can be represented by an i-node? [*4 marks*]

      ii. Consider storing a 10 MB file using the file system described above. How many direct addresses (pointers) are used in referencing this file? [*3 marks*]

      iii. How much of the 10 MB file in ii. will be referenced by only direct pointers? [*3 marks*]

      iv. How much of the 10 MB file in ii. will be referenced through only the single-indirect pointer? [*3 marks*]

      v. How much of the 10 MB file in ii. will be referenced through only the double-indirect pointer? [*3 marks*]

      vi. How much of the 10 MB file in ii. will be referenced through only the triple-indirect pointer? [*3 marks*]

3. (a) What is the difference between a blocked process and a ready process? [*2 marks*]

(b) Using the following test and set() function, implement blocking locks.

```
bool test_and_set(bool *flag) {
            bool old = *flag;
            *flag = True;
            return old;
}
```

In other words, write Acquire() and Release() functions which avoid busy-waiting. You are provided with a wait queue and two functions (enqueue process() and dequeue process()) that can be used to manipulate the wait queue. Specifically, enqueue_process() puts a calling process into the wait queue and lets it go to sleep, whereas dequeue_process() takes one waiting process out of the wait queue and places it onto the front of the ready queue. [*6 marks*]

(c) List four necessary conditions for deadlock. Explain each condition in one or two sentences. [*4 marks*]

(d) Five silent philosophers sit at a round table with bowls of spaghetti. Forks are placed between each pair of adjacent philosophers. Each philosopher must alternately think and eat. However, a philosopher can only eat spaghetti when he has both left and right forks. Each fork can be held by only one philosopher and so a philosopher can use the fork only if it is not being used by another philosopher. After he finishes eating, he needs to put down both forks so they become available to others. A philosopher can take the fork on his right or the one on his left as they become available, but cannot start eating before getting both of them. Eating is not limited by the remaining amounts of spaghetti or stomach space; an infinite supply is assumed.

i. Depending on how the philosophers pick up forks, they may need to eternally wait for each other to release a fork. Provide one example of when such a situation can occur. [*4 marks*]

ii. Discuss a solution (or strategy) to prevent deadlock. [*6 marks*]

(e) In a batch system using shortest-job-first (SJF) scheduling, what is the average turnaround time for a set of four jobs whose running times are 13, 3, 10 and 15 minutes? Show your work. [*3 marks*]