UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR09015 OPERATING SYSTEMS

Friday 14th August 2015

09:30 to 11:30

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

CALCULATORS MAY NOT BE USED IN THIS EXAMINATION

Year 3 Courses

Convener: S. Viglas
External Examiners: A. Cohn, T. Field

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. (a) State three advantages of loadable kernel modules. [*3 marks*]

   (b) Briefly explain context switching and process scheduling briefly. [*4 marks*]

   (c) Briefly explain what a race condition is. [*2 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.

   Now suppose there are two types of dining philosophers. One type always picks up his left fork first and the other type always picks up his right fork first - call these a lefty and a righty. Each type executes consecutive "wait"s on their forks (left followed by right for lefties, and the other way around for righties), eats, then put down the forks in the reverse order of the waits (right followed by left for lefties, and the other way around for the righties). Explain why every seating arrangement of lefties and righties, with at least one of each, avoids deadlock. [*8 marks*]

   (e) For each set of threads below, determine if a deadlock state can be reached. If so, fix the deadlock; if not, give sufficient proof the deadlock cannot occur. Assume that `mutex_a` should be acquired before read and write on variable a, `mutex_b` for variable b, etc.

```
     Thread 1              Thread 2
     lock(mutex_a)         lock(mutex_b)
     lock(mutex_b)         lock(mutex_c)
     b=a                   c=b
 i.  unlock(mutex_a)       unlock(mutex_b)
     lock(mutex_c)         lock(mutex_a)
     b=b+c                 a=c*2
     unlock(mutex_c)       unlock(mutex_a)
     unlock(mutex_b)       unlock(mutex_c)
```

[*4 marks*]

```
     Thread 1              Thread 2
     lock(mutex_h)         lock(mutex_g)
     lock(mutex_i)         lock(mutex_i)
     lock(mutex_j)         i=g+g
     j=i+h                 unlock(mutex_i)
 ii. unlock(mutex_j)       unlock(mutex_g)
     unlock(mutex_h)
     lock(mutex_g)
     g=g+i
     unlock(mutex_g)
     unlock(mutex_i)
```

[*4 marks*]

2. (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 (2^{12} 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*]

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

(b) Provide one advantage and 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]*