UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR09015 OPERATING SYSTEMS

Wednesday 1st May 2013

14:30 to 16:30

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

CALCULATORS MAY NOT BE USED IN THIS EXAMINATION

Year 3 Courses

Convener: K. Kalorkoti
External Examiners: K. Eder, T. Field

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. This question concerns the interaction of scheduling and resource control.

   (a) i. Explain what it means for a process to be *blocked* in scheduling. [*2 marks*]

   ii. Outline **briefly** how a simple pre-emptive scheduling system with a fixed quantum and static or dynamic priorities works. [*6 marks*]

   (b) Consider a single-processor system in which processes are pre-emptively scheduled with a quantum of 1 unit, and are assigned priorities on creation which do not change. Suppose that three processes are created with the following properties:

   - P_1 starts at time 0, has priority 1 (low), and needs to execute for 5 quanta to complete its work.
   - P_2 starts at time 3, has priority 3 (high), and needs 4 quanta to complete its work.
   - P_3 starts at time 4, has priority 2 (medium), and needs 10 quanta to complete its work.

   All processes run without invoking any blocking procedures.

   Draw a diagram to show which processes are executing at which times, and note the finishing time of each process. [*4 marks*]

   (c) Now consider a system as in the previous part, but with the following additional behaviour:

   - After executing for 1 quantum, P_1 requests exclusive access to a resource R. Once it obtains R, it executes for 3 quanta before releasing it and continuing with its remaining quantum of execution.
   - After executing for 1 quantum, P_3 requests exclusive access to R. Having obtained it, it executes for 1 quantum before releasing it and continuing.

   When a process is waiting for a resource that is not available, it is blocked.

   Draw a similar diagram, noting the finishing times. Why might this be considered a serious problem? [*6 marks*]

   (d) The problem seen in the previous part is known as 'priority inversion'. One proposed solution is 'priority inheritance': if a process holds a resource, its priority is temporarily raised to that of the highest priority process waiting for the resource.

   i. Show how this avoids the problem in the situation of the previous part. [*3 marks*]

   ii. Discuss any disadvantages you can see with this solution. [*4 marks*]

2. This question concerns the implementation of mutual exclusion.

   (a) Define the mutual exclusion problem, and give the standard criteria to be satisfied by a successful solution. [*8 marks*]

   (b) State Peterson's algorithm. Show that it satisfies the key mutual exclusion criterion, and say which of the other criteria from part (a) it satisfies. [*8 marks*]

   (c) The following algorithm for mutual exclusion is due to Donald Knuth. In this code, the two processes are identified by `i` being `0` or `1`; `flag[]` is an array of two global flag variables, each of which can have the value `CRIT` (in the critical section), `REQUEST` (asking to enter the critical section) or `OUT` (outside the critical section). The global variable `turn` is initialized to either `0` or `1`.

```
 1   j = 1 - i; // j is the id of the other process
 2   repeat
 3     flag[i] = REQUEST;
 4     // busy-wait while it's not our turn and the other
 5     // process is either in or requesting
 6     while ( turn != i && flag[j] != OUT )
 7       { }
 8     // may be safe to enter
 9     flag[i] = CRIT;
10     // but if the other process is still in critical
11     // go round again
12   until ( flag[j] != CRIT );
13
14   /* critical section here */
15
16   /* yield the turn to the other process */
17   turn = j;
18   flag[i] = OUT;
```

      i. By a similar technique to that used for Peterson's algorithm, or otherwise, show that this algorithm does satisfy mutual exclusion. [*5 marks*]

      ii. A distinctive feature of this algorithm is the use of three flag states instead of two. Suppose that the `REQUEST` on line 3 were replaced by `CRIT`. What would happen? [*4 marks*]

3. This question is about files and file systems.

   (a) Outline the traditional Unix file system, including definitions of *inode*, *directory*, and *metadata*. You need not consider disk blocking. [*7 marks*]

   (b) Explain how a Unix file can have more than one name ('hard links'). What happens when a hard link is deleted? [*5 marks*]

   (c) A 'symbolic link' or 'symlink' is a special file which contains the name of a file. When a user program opens a symlink, the OS reads the name from the symlink, and opens that file instead.

   How much of the Unix file metadata makes sense for a symlink? For each item you discuss, give a brief justification of why it does or does not make sense. [*6 marks*]

   (d) A famous security problem on Unix is the 'symlink race', which works as follows.

   > Suppose that a privileged program `foobar` writes some sensitive information to a file `/tmp/foo`. If the file already exists, `foobar` checks the ownership and permissions before writing; if the file does not exist, `foobar` creates it with the correct permissions, using the `open(...,O_CREAT)` system call that opens a file, creating it if necessary.
   > Now suppose that an attacker manages to create a symlink `/tmp/foo` pointing to `/etc/passwd`, *after* `foobar` has checked for the existence of `/tmp/foo`, but *before* `foobar` calls `open()`. Then the privileged program will overwrite `/etc/passwd` (or any other file of the attacker's choice).

   The correct solution to the problem is for all such privileged programs to be re-written so that they avoid the race (which can be done on Unix filesystems). However, this is often considered unrealistic, and it is suggested that it is better to add general restrictions to the operating system that will prevent most instances of the problem.

   What suggestions can you make for changes to the treatment of `open()` or of symlinks that might help? Discuss both the advantages and disadvantages of any suggestions you make. (You may get some ideas by reconsidering your answer to part (c), but this is not the only approach.) [*7 marks*]