UNIVERSITY OF ABERDEEN SESSION 2016/2017

Examination in CS3026 – Operating Systems

12 Dec 2016 Time: 15.00pm – 17.00pm

Candidates are not permitted to leave the Examination Room during the first or last half hours of the examination.

Answer TWO out of the three questions.

Each question is worth 25 marks; the marks for each part of a question are shown in brackets.

1. (a) Processes and Threads allow concurrent execution of programs. Describe (i) the relationship between processes and threads, and (ii) the benefits and disadvantages of threads vs. processes.

[5]

(b) Describe a thread management approach that does not require a mode switch. What are the advantages and disadvantages?

[4]

(c) Describe and name the three basic states of a process: when it is waiting to gain access to the CPU, being executed on the CPU, or waiting for I/O. Also describe the state transitions a process may go through during its execution (you can use a drawing to show the transitions).

[5]

(d) What is the so-called "Critical Section Problem"? What are the three basic requirements for any solution to the critical section problem? Why are these requirements needed?

[5]

(e) Consider the following code fragment for solving the "Dining Philosophers" problem: 5 processes compete for 5 resources. The implementation shown for a process i may lead to a deadlock – explain why this is the case. Extend the code with a solution that avoids deadlock and starvation.

```
Semaphore fork[5]; init(fork,{1,1,1,1,1};
Process i
while(TRUE)
{
   philosopher_thinks();
   wait(fork[i];
   wait(fork[(i+1) % 5]);
   philosopher_eats() ;
   signal(fork[(i+1) % 5]);
   signal(fork[i];
}
```

[4]

(f) Both the concept of a semaphore and a monitor can be used to guarantee mutual exclusion between processes. However, there is a difference in terms of how they have to be used in programs – what is the difference?

[2]

2. (a) Explain how the Clock Algorithm implements an efficient paging mechanism for virtual memory.

[6]

(b) Explain the problem of fragmentation in the context of allocating memory for processes. What kind of fragmentation can occur when paging is used?

[4]

(c) Explain what the Principle of Locality is, and why it is important for the efficient management of virtual memory.

[2]

(d) Explain what a Working Set is and how it is related to the Principle of Locality. The operating system may monitor the size of the Working Set – what can be concluded from this information?

[4]

(e) A “Translation Lookaside Buffer” (TLB) is used to support virtual memory management. Explain what it is used for and why. Provide an explanation for the concepts “TLB miss” and “TLB hit” and what they mean for a virtual memory management. What has to be done with the TLB buffer when a context switch occurs?

[5]

(f) In order to manage files on a large hard disk, i-nodes are used. We assume that a disk block is of size 4KB, and that an i-node contains 12 direct index entries, one single indirect index entry, one double indirect index entry and one triple indirect index entry. We also assume that a 32-bit (4 bytes) format is used for block numbers stored as index entries. What is the largest possible size of a file, such an i-node can address with its direct and indirect indices?

[4]

3. (a) Explain the difference between preemptive and non-preemptive scheduling. Which type of scheduling is better suited for interactive systems and why? For the scheduling strategy “Round Robin” (RR) and for “Shortest Job First” SJF, point out whether they are a pre-emptive or non-pre-emptive scheduling policy.

[5]

(b) Consider the scheduling of processes P1, P2, P3 and P4. These processes have the following arrival times, where they become ready for execution: for P1 = 0, for P2 = 4, for P3 = 5, and for P4 = 7 time units. We also assume that these processes will have the following CPU execution times after their arrival: for P1 = 7, for P2 = 4, for P3 = 1, and for P4 = 4 time units.

Process	Arrival Time	Execution Time
P1	0	7
P2	4	5
P3	5	4
P4	6	3

Consider the use of a Shortest-Job-First (SJF) policy:
(i) show in what sequence the processes will be scheduled, and
(ii) calculate the average waiting time for this batch of processes.

[4]

(c) Point out a problem that may occur when a priority scheduling policy, such as SJF, is used. What can be done to counteract this problem? Which scheduling policy does not have this particular problem?

[4]

(d) Explain the concept of Lottery Scheduling. How can it be used to prioritise processes? How does it differ from "Fair Share Scheduling"?

[5]

(e) With Round-Robin scheduling, each process gets a fixed unit of CPU time (a time quantum). If the time quantum is 20ms and there are 10 processes waiting to be scheduled for execution – what is the maximum time a process can expect to wait?

[3]

(f) Let's assume that a computing system has to handle three video streams at the same time:

- Stream 1 is handled by process A: every 30 msec a frame arrives to be processed by process A, process A needs 20 msec CPU time to decode a frame
- Stream 2 is handled by process B: every 40 msec, a frame arrives to be processed by process B, process B needs 20 msec to decode a frame
- Stream 3 is handled by process C: every 50 msec, a frame arrives to be processed by process B, process C needs 20 msec to decode a frame

Given the processing times for each process, is the performance of the computer system good enough to schedule all three streams without delay? In your answer, explain (a) how you can calculate whether a computer system's performance is good enough to handle a real-time scheduling task, and (b) provide an answer for this particular example.

[4]

END OF PAPER