Final Exam in Operating Systems

Department of Information Technology
Division of Computer Systems
Uppsala University
2010-01-12

Lecturer: Ivan Christoff
Time: 08 – 13

Rules:
1. Do not solve more than one problem per sheet.
2. Make sure to motivate your answers.
3. Write legibly.
4. Write only on one side of each sheet.
5. Write your name on each sheet.

Good Luck!

Problem 1 (10 points) Answer the following questions:

1. What is thrashing? How can thrashing be avoided?
2. Explain the terms time-sharing and multiprogramming.
3. Explain the terms mutual exclusion, critical section and starvation.
4. What is the difference between access lists (for objects) and capability lists (for domains)?
5. Describe two different techniques for dealing with recovery from deadlock.

Problem 2 (5 points) Consider the processes $C_1, ..., C_5$ given below.

<table>
<thead>
<tr>
<th>Process</th>
<th>Burst time</th>
<th>Arrival time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>$C_2$</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>$C_3$</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>$C_4$</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$C_5$</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Compute (and compare) the average waiting time when the processes are scheduled with Preemptive Shortest Job First and with Round Robin (RR). For RR you should use the time quantum which gives the shortest average waiting time.

When computing the average waiting time you must take the context switch time into consideration. Assume that a context switch costs 1 time unit.

Problem 3 (3 points) Assume that we have a memory access time of 50 ns, a page-fault service time of 0.5 ms, and that one of 10000 accesses leads to a page-fault. Compute the efficient access time. If there is a requirement that the efficient access time is not greater than 75 ns, how often can we “allow” pages-faults to occur?
Problem 4 (3 points) Consider the RAG given below.

\[ P = \{P_1, P_2, P_3\}, \ R = \{R_1, R_2, R_3\} \]
\[ E = \{P_1 \rightarrow R_1, P_2 \rightarrow R_2, R_1 \rightarrow P_2, R_1 \rightarrow P_3, R_2 \rightarrow P_3, R_3 \rightarrow P_1, R_3 \rightarrow P_2, R_3 \rightarrow P_3\} \]

Answer the following questions:
1. Is the system in a safe state?
2. Assume that we add the requests \( P_1 \rightarrow R_2 \) and \( P_2 \rightarrow R_1 \) to E. Is the new system in a safe state?

Problem 5 (3 points) What is a multilevel page-table? When is it used?

Problem 6 (8 points) Consider an instance of the Readers and Writers problem. The program given below shows the structure of the Reader/Writer processes (the program is by no means complete).

```
program Readers-writers;
var mutex, wrt : semaphore;
    readcount : integer;

procedure Writer;
begin
    repeat
        writing is performed;
    until false
end;

procedure Reader;
begin
    repeat
        reading is performed;
    until false
end;
```

Assume that we are studying the second readers-writers problem, i.e. if a writer is waiting to access the data, no new readers may start reading.

1. Sketch a solution with semaphores (using the above given definition). How should the semaphores be initialized?
2. Does your solution avoid starvation?

Problem 7 (4 points) Describe how the working-set model works. When is it used?

Problem 8 (4 points) Consider the reference string: 4, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0. Simulate page-replacement according to FIFO, LRU and the optimal algorithm, assuming a system with 3 free frames.