UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

Faculty of Science and Engineering
Department of Computer Science and Information Systems

End-of-Semester Exam

Academic Year:	2008/2009	**Semester:** Autumn	
Module Title:	Operating Systems	**Module Code:** CS4023	
Exam Duration:	2½ Hours	**Total Marks:** 70 (1 mark is equal to 1% of the final grade)	
Lecturer:	Dr. N. S. Nikolov		

Instructions to Candidates:

All questions should be answered. In Q9, choose to answer either Option 1 or Option 2.
Keep the answers to questions Q1-Q7 as brief as possible (typically, no more than 2-5 sentences each).
State clearly any assumptions you make.

Please write **ALL** answers in the answer booklet.

Questions

Q1. Explain briefly the difference between the concepts of *batch operating system*, *multiprogramming operating system* and *multitasking operating system*. **(5 marks)**

Q2. Give a short definition of *process*. How does the operating system prevent a process from monopolizing a processor? List the five main states of a process and draw a state-transition diagram. **(5 marks)**

Q3. What are the two fundamental models of interprocess communication (IPC)? What are the major pros and cons in using one of them instead of the other? **(5 marks)**

Q4. How can the kernel and user mode of operation of a CPU be used by an operating system for protecting resources from malicious behaviour? What is the role of system calls and how are they typically implemented? **(5 marks)**

Q5. Give a short definition of *thread*. What key advantage would you get by running a multithreaded application on a multiprocessor system over running it on a single-processor system? **(5 marks)**

Q6. In a file-system interface, explain the purpose of the **`open()`** operation. **(5 marks)**

Q7. What will be printed on the screen by the following code and why? **(5 marks)**

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int x;
int main()
{
   pid_t pid = fork();
   x = 5;
   if (pid == 0) x = x + 10;
   x = x + 5;
   printf("x = %d ", x);
   exit(0);
}
```

Q8. Consider an operating system with a single ready queue. Suppose that processes P_1, P_2, P_3 and P_4 have been admitted into the ready queue in this order and all four of them are already in the queue when the CPU scheduler decides which process is the next to run: **(9 marks)**

Process	**CPU Burst Time**
P_1	5.0
P_2	4.0
P_3	2.0
P_4	1.0

a. **(4 marks)** If you want to minimise the waiting time for process P_2 would you better use the *First-Come First-Served* or the *Shortest-Job First* (non preemptive) CPU scheduling algorithm?

b. **(5 marks)** What time-quantum will minimize the turnaround time for process P_3 if the Round-Robin algorithm is used for CPU scheduling?

Q9. Answer either Option 1 or Option 2. **(10 marks)**

Option 1.

Two concurrently running threads T_0 and T_1 share variables which they both modify in a section of their code marked as *critical section*. The following synchronization protocol has a flaw. Explain what it is and suggest how it can be fixed.

```
//Shared variables
boolean t1WantsToEnter = false;
boolean t2WantsToEnter = false;

//Thread T1:
…
while (!done) {
  t1WantsToEnter = true;
  while (t2WantsToEnter);
  // START OF CRITICAL SECTION
  …
  // END OF CRITICAL SECTION
  t1WantsToEnter = false;
}

//Thread T2:
…
while (!done) {
  t2WantsToEnter = true;
  while (t1WantsToEnter);
  // START OF CRITICAL SECTION
  …
  // END OF CRITICAL SECTION
  t2WantsToEnter = false;
}
```

Option 2.

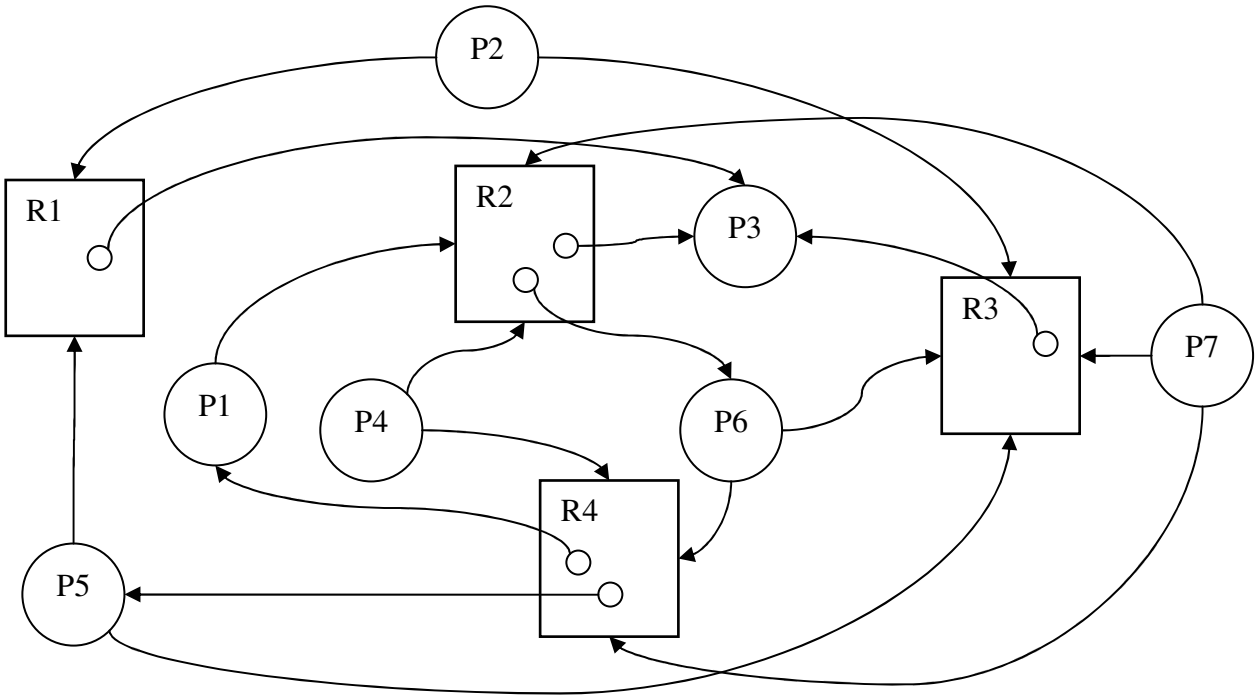
Consider two concurrently running threads which are synchronized with the use of a binary semaphore S. Consider the following implementation of the semaphore operation **`wait()`**:

```
wait(S)
{
  while(S <= 0);
  S--;
}
```

Write the corresponding version of the **`signal()`** operation. Show with an example that if the **`wait()`** and **`signal()`** semaphore operations are not executed atomically, then mutual exclusion may be violated.

Q10. Deadlock Detection: **(8 marks)**

In the following resource allocation graph, the large circles represent processes, the boxes are classes of identical resources, and the small circles are the resources. Is there circular wait in the system? Can we say that the system is deadlocked or not?



Q11. A system supports virtual memory with demand paging. **(8 marks)**
Consider the following string of page references by a particular process P_0.

1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5, 4, 2, 3

Assuming P_0 has been allocated *three* frames in the physical memory:

a. **(4 marks)** Compute the number of page faults if the Least-Recently-Used (LRU) page replacement algorithm is used.
b. **(4 marks)** Compute the minimum number of page faults.

End of Exam