UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

COLLEGE *of* INFORMATICS *and* ELECTRONICS
Department of Computer Science and Information Systems

End-of-Semester Exam

Academic Year:	2007/2008	**Semester:** Autumn
Module Title:	Operating Systems	**Module Code:** CS4023
Exam Duration:	2½ Hours	**Total Marks:** 70 (70% of the final grade)
Lecturer:	Dr. N. S. Nikolov	

Instructions to Candidates:

Please write **ALL** answers in the answer booklet.
State clearly any assumptions you make.

Choose to answer either Q5 or Q6. All other questions should be answered.

QUESTIONS

Q1. Operating System Services: **(10 marks)**

a. **(8 marks)** Explain how each of the following five services provided by an operating system provides convenience to the users: *program execution*, *I/O operations*, *file-system manipulation*, *communications*, *error detection*. Explain also in which cases it would be impossible for user-level programs to provide these services.

b. **(2 marks)** Consider whether the operating system should include applications such as Web browsers and mail programs. Argue both that it should and that it should not, and support your answer.

Q2. Processes and Threads: **(10 marks)**

a. **(5 marks)** 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 diagram of the state transition.

b. **(5 marks)** 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?

Q3. System Calls: (5 marks)

What will be printed out by the following single-threaded C program? Explain why.

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

int a = 10;
int b = 5;

int main()
{
  pid_t pid;

  pid = fork();

  if (pid == 0) {
    a = a+b;
  }
  else if (pid > 0) {
    wait(NULL);
    a = a+b;
    printf("The value of a is %d", a);
  }
  exit(0);
}
```

Q4. CPU Scheduling: (15 marks)

Suppose that the following processes arrive for execution at the times indicated. Each process will run the listed amount of burst time

Process	**Arrival Time**	**Burst Time**
P_1	0.0	7.0
P_2	0.5	4.0
P_3	1.0	2.0
P_4	2.0	1.0

a. **(5 marks)** What is the average turnaround time for these processes with the First-Come First-Served (FCFS) scheduling algorithm?

b. **(5 marks)** What is the average turnaround time for these processes with the nonpreemptive version of the Shortest-Job First (SJF) scheduling algorithm?

c. **(5 marks)** What is the average turnaround time for these processes with the preemptive version of the Shortest-Job First (SJF) scheduling algorithm?

Q5. Process Synchronization (Option 1): (12 marks)

Consider two concurrently running processes P_0 and P_1 which share a variable.

a. **(2 marks)** Explain what is the *critical-section problem* for P_0 and P_1?

b. **(10 marks)** A solution to the critical-section problem should satisfy the *bounded-waiting* requirement. In particular, there should be a bound, or limit, on the number of times P_0 is allowed to enter its critical section after P_1 has made a request to enter its critical section and before that request is granted. Show that this is true for the Peterson algorithm.

The Peterson algorithm for processes P_0 and P_1 shown below:

P_0	P_1

`while (true) {`
`  flag[0] = true;`
`  turn = 1;`
`  while ( flag[1] && turn == 1);`
`  CRITICAL SECTION`
`  flag[0] = false;`
`  REMAINDER SECTION`
`}`
	`while (true) {`
	`  flag[1] = true;`
	`  turn = 0;`
	`  while ( flag[0] && turn == 0);`
	`  CRITICAL SECTION`
	`  flag[1] = false;`
	`  REMAINDER SECTION`
	`}`

Q6. Process Synchronization (Option 2): (12 marks)

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(S):

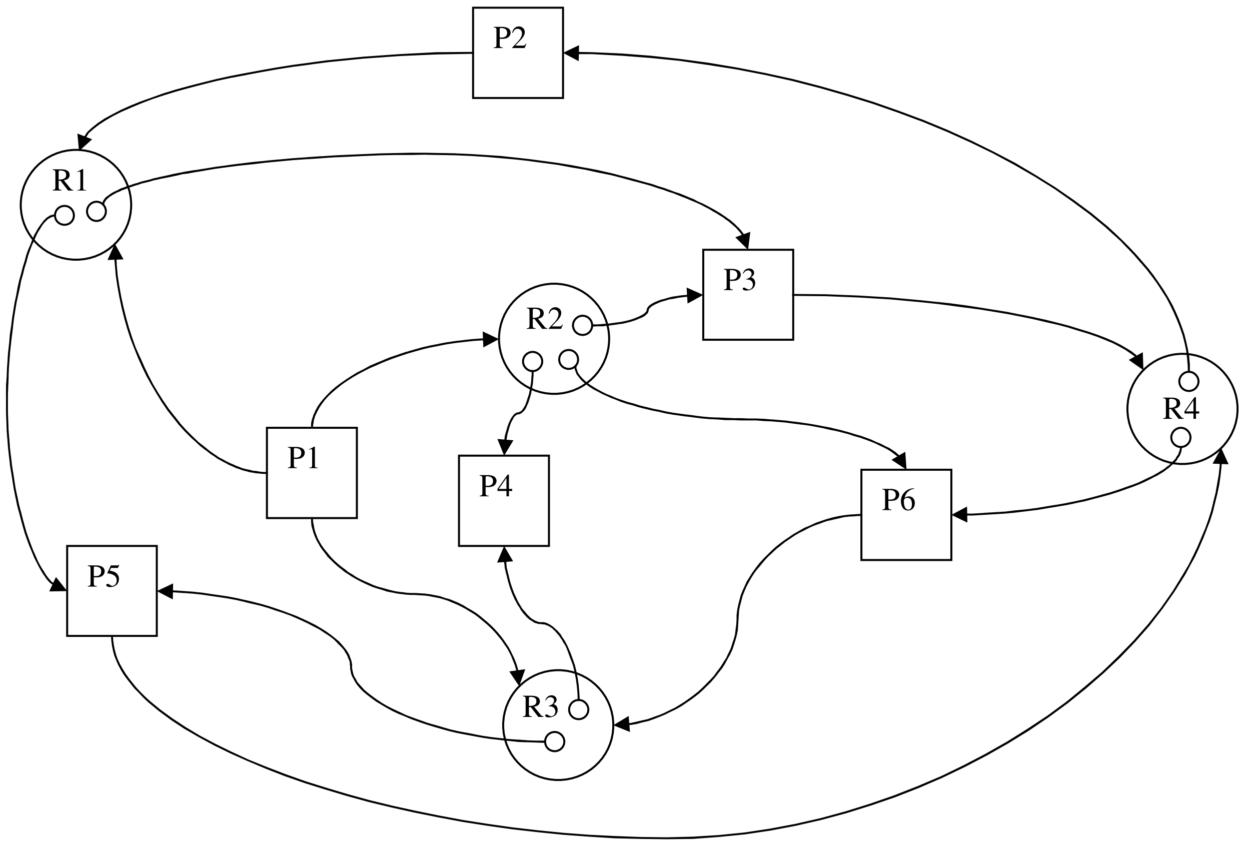
```
wait(S)
{
  while(S <= 0);
  S--;
}
```

a. **(2 marks)** Write the corresponding version of the signal(S) operation.

b. **(10 marks)** Show that, if the wait(S) and signal(S) semaphore operations are not executed atomically, then mutual exclusion may be violated.

Q7. Deadlock Detection: **(8 marks)**

In the following resource allocation graph, the squares represent processes, the larger circles are classes of identical resources, and the small circles are the resources. Reduce the graph by process. Draw each step. Is there a potential danger for deadlock in the system represented by the graph?



Q8. Virtual Memory: **(10 marks)**

A system supports virtual memory with demand paging. 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. **(3 marks)** Compute the number of page faults if the First-In First-Out (FIFO) page replacement algorithm is used.
b. **(3 marks)** Compute the number of page faults if the Least-Recently-Used (LRU) page replacement algorithm is used.
c. **(4 marks)** Compute the minimum number of page faults.

END OF EXAM