Massey University
ALBANY CAMPUS

EXAMINATION FOR 159.335
OPERATING SYSTEMS AND CONCURRENT PROGRAMMING
Semester Two - 2002

Time Allowed: THREE (3) Hours

INSTRUCTIONS

Attempt **ALL SEVEN (7)** questions.

This final examination contributes 70% to the final assessment.
Calculators are permitted

1. (a) Using a simple diagram, briefly explain the concept of multiprogramming.

[3 marks]

(b) An operating system could be described as 'A very large interrupt handler', explain what this means.

[2 marks]

(c) Using a simple example, briefly describe a cache.

[3 marks]

(d) Give 3 operations that are permitted in supervisor mode but not user mode.

[2 marks]

2. (a) Briefly describe the steps involved in installing an operating system.

[2 marks]

(b) What could the following code print when run under a UNIX OS?

```
for(i=0;i<5;i++) {
  n=fork();
  printf("%d%",n);
}
```

[3 marks]

(d) Explain why the output for part (b) may not be the same each time the program is run.

[2 marks]

(e) Draw a diagram to illustrate the life cycle of processes in an operating system which supports only non-preemptive scheduling.

[2 marks]

(d) When a process is in the ready state, can it perform calculations?

[1 mark]

3. (a) The following processes are to be scheduled

Process	***Arrival Time(ms)***	***Burst Time(ms)***
P_1	5	15
P_2	10	5
P_3	20	15
P_4	30	5

Draw scheduling diagrams and calculate the average waiting time and response time for these processes when using the following algorithms.

(i) FCFS
(ii) SJF
(iii) SRTF
(iv) RR with q=5
(v) RR with q=10

Comment on your results

[6 marks]

(b) A good value to choose for the Round Robin time quantum is one where most bursts complete in 1 time quantum, why is this?

[2 marks]

(c) What is the 'load' of a system, how is it calculated?

[2 marks]

4. (a) Give entry and exit code that could be used to protect a critical section using the testandset instruction.

[2 marks]

(b) The following is an attempt at the implementation of the wait operation on a semaphore. Explain why this is not a good solution.

```
while (TestandSet(lock))
S = S - 1;
if (S < 0) {
     lock = false;
     sleep(S);
} else
     lock = false;
```

[3 marks]

(c) A semaphore that is to be used as a mutex to protect a critical section is initialised to 2 instead of 1, what will happen to the program?.

[2 marks]

(d) A process has a number of threads sharing a mutex semaphore (and other shared variables) and running the following code

```
….
signal (mutex);
… Critical section
wait (mutex);
…
```

It has also one thread running the following code also sharing the same semaphore:

```
….
wait (mutex);
… Critical section
signal (mutex);
…
```

(i) What problems may occur with this process?
(ii) Why is not always possible to reproduce errors caused by this situation?
(iii) Suggest a high-level synchronization construct that could help avoid mistaken usage of semaphores.

[3 marks]

5. (a) The following solution to the dining philosopher problem has possible starvation. Assuming that the time to eat is limited to a certain amount, change the code so that starvation is avoided.

Shared data

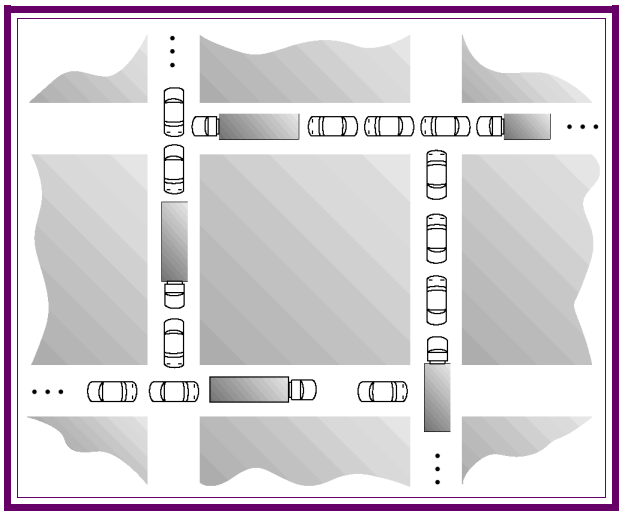
```
int p[N];                  /* status of the philosophers */
semaphore s[N]=0;          /* semaphore for each philosopher*/
semaphore mutex=1;         /* semaphore for mutual exclusion*/
```

Code

```
#define LEFT(n) (n+N-1)%N    /* Macros to give left */
#define RIGHT(n) (n+1)%N     /*and right around the table*/

void test(int no) {          /* can philosopher 'no' eat */
  if ((p[no] == HUNGRY) &&
      (p[LEFT(no)] != EATING) &&
      (p[RIGHT(no)] != EATING) )  {
    p[no]=EATING;
    signal(s[no]);          /* if so then eat */
  }
}

void take_forks(int no) {    /* get both forks */
  wait(mutex);         /* only one at a time here please */
  p[no]=HUNGRY;              /* I'm Hungry */
  test(no);                  /* can I eat? */
  signal(mutex);
  wait(s[no]);              /* wait until I can */
}

void put_forks(int no) {     /* put the forks down */
  wait(mutex);              /* only one at a time here */
  p[no]=THINKING;            /* let me think */
  test(LEFT(no));            /* see if my neighbours can now eat */
  test(RIGHT(no));
  signal(mutex);
}

void philosopher(int no) {
  while(1) {
    ...think....
    take_forks(no);         /* get the forks */
    ....eat (for a limited time).....
    put_forks(no);          /* put forks down */
  }
  return NULL;
}
```

[5 marks]

(b) Given the following diagram of a traffic jam, show that the 4 conditions for the deadlock to occur are true. Explain what considerations/analogies can be made (for example a car is like a process).



[5 marks]

6. (a) Briefly describe the following allocation algorithms:

(i) First fit
(ii) Best fit
(iii) Worst fit

[3 marks]

(b) Briefly explain what **trashing** is and what problems it may causes in a system.

[2 marks]

(c) Explain why segmentation and paging are often combined into one scheme.

[2 marks]

(d) Consider a demand-paging system with the following time utilization:

CPU	10%
Paging disk	97%
Other I/O	1%

Three consultants suggested the following three different solutions:

(i) Get a faster CPU
(ii) Install a bigger disk dedicated to paging
(iii) Install more memory

Comment about the consequences of adopting each of the proposed solution and indicate which one would be the best choice.

[3 marks]

7. (a) If a choice was possible, what file system strategy (contiguous, linked or indexed) should be used in the following cases:

(i) Files are small and always accessed sequentially.
(ii) Files are large and accessed randomly.

[2 marks]

(b) An operating system uses a combined indexing file system that has a block size of 4KB and block numbers of 32 bits. The FS may use double indirect blocks at most. The file's inode contains 14 direct blocks and 2 indirect blocks (one for each level).

(i) What is the maximum possible size for a file in this FS?

[2 marks]

(iii) How many blocks in total would a 200MB file use?

[2 marks]

(c) You are asked to specify a server to be used in the Small & Busy Ltd. Company's main branch. Critical data for the company will be stored in this machine. Although the data acquired during the day is eventually backed-up overnight, the IT manager is worried about the possibility that a disk crashes and that a day's work may be lost. He is also concerned about the amount of time to fix the server if such a disastrous event occurs. Propose a solution based on RAID that provides a reliable and fast solution for the server's storage. Explain how your proposal would work in a disaster/recovery situation (use drawings if necessary).

[4 marks]

+ + + + + + + + + +