Ollscoil na hÉireann, Gaillimh GX 1505
National University of Ireland, Galway

Semester II Examinations, 2003/2004

Exam Code(s)	2IF1
Exam(s)	Second Year Information Technology
Module Code(s)	CT213
Module(s)	Computer Systems and Organization
Paper No.	1
Repeat Paper	
External Examiner(s)	Professor Paddy Nixon
Internal Examiner(s)	Professor G. Lyons
	Mr. P. Bigioi

Instructions: Answer any 4 questions.
Use a separate answer book for each section.
All questions will be marked equally.

Duration	3 hrs
No. of Answer Books	2
No. of Pages	4
Department(s)	Information Technology

Section A (Questions 1 to 3)

Question 1

a) What is the structure of a synchronous finite state machine (FSM)? Describe its principle of operation.

6 MARKS

b) What is the difference between Mealy and Moore finite state machines? Draw the structure for each of them.

9 MARKS

c) Describe the basic computer organization and describe the operation and role of each of the subsystems (buses, central processing unit, memory and I/O subsystems). What is the main difference between Von Neumann and Harvard architectures?

10 MARKS

Question 2

a) Describe briefly the ISA (Instruction Set Architecture) of a processor and processor micro-architecture. Perform a RISC vs. CISC comparative analysis.

6 MARKS

b) What is an instruction? Describe possible instruction formats. Describe at least four addressing modes typically used in processor.

9 MARKS

c) Describe (draw) the typical organization of a processor and describe its components (registers, execution unit and control unit). What is pipelining? What are the advantages and associated problems with pipelining?

10 MARKS

Question 3

a) Describe briefly the main types of memory and the memory hierarchy.

6 MARKS

b) What is the cache? Describe its principle of operation and possible types of cache.

9 MARKS

c) In the context of cache memory, describe possible approaches to block placement, block identification, block replacement and write strategies.

10 MARKS

Section B (Questions 4 to 6)

Question 4

a) What are the possible states of a process? Draw the transition diagram and explain in detail the states and transitions.

6 MARKS

b) Describe the process scheduler organization and operation. Describe the cooperative and pre-emptive schedulers.

9 MARKS

c) Describe the process scheduling algorithms. Give a practical example for time slice (round robin) algorithm for a number of four processes and a time slice size of 500us. The switching time is 50 us. The processes have the following service times: $\tau(p1) = 2.2ms$, $\tau(p2) = 5ms$, $\tau(p3) = 1ms$, $\tau(p4) = 3.2ms$. Compute the average turn around time and average wait time.

10 MARKS

Question 5

a) Describe the typical operating system organization and describe each of the subsystems.

6 MARKS

b) Consider four processes (A, B, C and D) that are stored on disk and are about to be loaded in the main memory, in the following order: C, D, B and A. Process A has five pages, process B has three pages, process C has five pages and process D has five pages. Assume that the main memory has a size of 16 physical frames (numbered 0 through 15, see Figure 1), all the frames are available and that the operating system will swap out the smallest process to make more room in the main memory.

- Show how the main memory will look after each process is loaded into the main memory.
- Show each process page table after the process A has been loaded into the memory.

9 MARKS

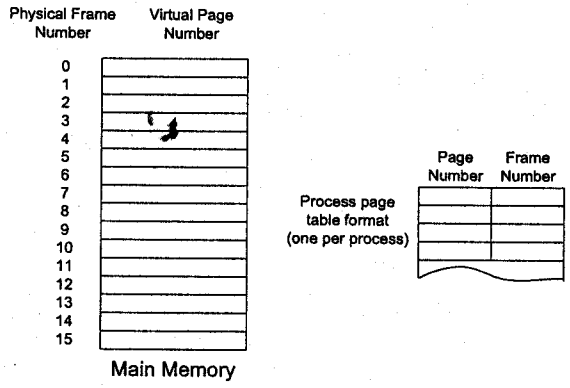

Figure 1: Initial Memory State and Process Page Table Format

c) Describe the I/O subsystem organization. Describe how direct I/O with polling is typically implemented. What is the main disadvantage of this approach? Why interrupt driven I/O is any better?

10 MARKS

Question 6

a) What is a critical section? Give examples of typical critical sections and describe briefly the methods to deal with critical sections.

6 MARKS

b) What are messages? What are the operating systems primitives to manipulate Messages? Describe the following design issues related to messages: synchronization, addressing and queuing discipline.

9 MARKS

c) Consider the code in Figure 2. What is the output of this program, under normal execution conditions (no errors)? Explain the output.

10 MARKS

```
#include <stdio.h>

main()
{
  int pid;
  int var = 1;

  pid = fork();

  if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork failed!\n");
        exit(-1);
  }
  else if (pid == 0) {
        /* child process */
        printf("Child: variable value is %d\n",var);
        var = 2;
        printf("I am the child, pid=%d\n", getpid());
        printf("Child: variable value is %d\n",var);
  }
  else {
        /* parent process */
        printf("I am the parent, pid=%d\n", getpid() );
        printf("Parent: variable value is %d\n",var);
        exit(0);
  }
}
```

Figure 2: Linux Program