THE UNIVERSITY OF AUCKLAND

EXAMINATION FOR BSc ETC 2003

COMPUTER SCIENCE

Operating Systems

(Time allowed: **TWO** hours)

NOTE:

Attempt ALL questions.

Marks for each question are shown and total **100**.

1.

a) Sometimes a user, administrator or the operating system itself has to stop a process. Give an example showing why the user might want to stop a process.
Give an example showing why the administrator might want to stop a process.
Give an example showing why the operating system might stop a process.

[3 marks]

b) Many operating systems, including Unix, provide a way to temporarily suspend a process from the user's keyboard. Describe how this could be done. Mention everything from the initial keyboard interrupt until the process has been suspended. (N.B. This question is not about the Java suspend thread call.)

[4 marks]

c) What are the main differences and similarities between a suspended process and a process waiting for a resource?

[3 marks]

2.

a) System call interfaces have to be carefully checked to ensure that there are no protection violations. Explain what could go wrong if the buffer to be written in the following write call was not checked for read access permission.

```
write(file, bufferAddress, lengthOfBuffer)
```

[3 marks]

b) Buffer overrun problems are a major security hole in many programs and operating systems. Describe how an input buffer overrun can be used to run another program. Draw a diagram if that helps your explanation.

[5 marks]

c) Describe a simple solution to buffer overrun problems.

[2 marks]

3.

Here is a class which can be used for a Java solution to the producer/consumer problem.

```
public class LimitedBuffer {

   private Object[] array;  // where the results are temporarily stored
   private int inPosition;  // where the next result goes
   private int outPosition; // where the next result comes from
   private int number;    // number of elements in the buffer

   public LimitedBuffer(int size) {
       array = new Object[size];
       inPosition = 0;
       outPosition = size - 1;
       number = 0;
   }

   public synchronized void insert(Object data) {// called by producers
       while (number == array.length) {
        try {
            wait();
        }
        catch (InterruptedException e) {}
       }
       array[inPosition] = data;
       inPosition = (inPosition + 1) % array.length;
       number++;
       notifyAll();
   }

   public synchronized Object retrieve() {     // called by consumers
       while (number == 0) {
        try {
            wait();
        }
        catch (InterruptedException e) {}
       }
       outPosition = (outPosition + 1) % array.length;
       Object data = array[outPosition];
       number--;
       notifyAll();
       return data;
   }

}
```

a) This code uses Java synchronization. What does the synchronized keyword do?

[2 marks]

b) What do the calls to wait() in both the insert and retrieve methods do? Why are they there?

[6 marks]

c) Why are the instance variables of this class all declared private? What could go wrong if they were public?

[2 marks]

d) Suppose that there are multiple producers waiting in the insert method. Explain in detail the steps that occur when one consumer calls retrieve.

[5 marks]

4.

a) The wound-wait and wait-die schemes are solutions to what problem?

[2 marks]

b) Of these two schemes which one produces fewer process restarts and why?

[5 marks]

5.

There are two copies of a resource R in a system. There are two processes P and Q with the following maximum resource requirements for completion:

P needs 2 Rs
Q needs 2 Rs

a) Using the Banker's algorithm demonstrate why it is not safe to allocate one R to P after allocating one R to Q as in the following table:

command	state (allocation of Rs)	
Q requests R	P0, Q1	safe
P requests R	P1, Q1	not safe

[2 marks]

b) Given the system in part a) with one R allocated to P and one R allocated to Q, list a series of requests and releases of Rs that prove it is possible for both processes to complete without deadlock.

[3 marks]

6.

Both access control lists (ACLs) and capabilities can be used to provide protection in operating systems. Which of these types better satisfies the following requirements? If both are equally as good write both. Justify your answers.

a) Information about all access rights on a particular object is easy to collect.

[2 marks]

b) Information about all access rights in a particular domain is easy to collect.

[2 marks]

c) Removing access rights from a particular user is easy.

[2 marks]

d) Provides convenient access to resources over a large network.

[2 marks]

e) Allows very specific allocation of access rights.

[2 marks]

7.

The standard Unix file access bits are neither capabilities nor access control lists.

a) What disadvantages does the file access bit system have compared to capabilities and access control lists? Give an example.

[3 marks]

b) What advantage does the file access bit system have compared to capabilities and access control lists?

[1 mark]

8.

a) Buffers are used in a variety of ways in operating systems. Describe two ways in which buffers are used when dealing with devices. Explain how the buffers help in each case.

[6 marks]

b) Device drivers use different methods to access devices and transmit data between the operating system and devices; interrupt-driven I/O and programmed I/O (PIO) are two such methods. Describe a situation where it may be appropriate to use PIO rather than an interrupt-driven device driver.

[3 marks]

9.

a) Files can be allocated on disk either contiguously or non-contiguously. Give two reasons why you would use non-contiguous allocation.

[2 marks]

b) The Inode in a Unix file system contains the information on disk of where the file data is stored. It also contains the count of the number of hardlinks to the file. Give a reason why it needs to store the count.

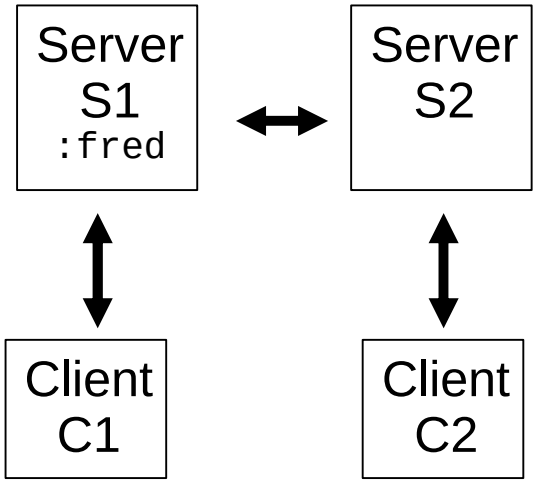
[2 marks]

c) Give two examples of when a versioning file system might be useful.

[2 marks]

10.

Suppose that I have client machines C1 and C2, and server machines S1 and S2. Suppose that there is a distributed file system across these three machines that supports location transparency. The connections between clients and servers are shown below. The file "fred" resides on server S1.



a) What does location transparency mean in this context?

[2 marks]

b) Suppose C2 wishes to open file "fred" for reading. The file system on S2 needs to find file "fred"somehow. How might it do so?

[2 marks]

c) Suppose that C2 caches the file locally. Describe a problem that can arise?

[2 marks]

11.

(The following question is based on assignment 3, Step 1.)

My operating system does not use virtual memory. The first free frame is always allocated first. The first free frame is the free frame with the lowest frame number.

Suppose I start with a physical memory containing 8 frames and that the memory is initially empty.

I then perform the following sequence of operations

```
allocateMemory("Process1", 3); // allocate 3 pages to process1
allocateMemory("Process2", 2); // allocate 2 pages to process2
free("Process1"); // free all pages in process1. Assume that
                  // this disposes of the page table.
```

Physical memory looks like this:

Frame Number	Label
0	free
1	free
2	free
3	Process2
4	Process2
5	free
6	free
7	free

Now I perform the following request

```
allocateMemory("Process3", 4); // allocate 4 pages to process3
```

Draw a diagram that shows the frame table **and** any page tables after performing the request. The frame table should look similar to the one given above. You do not need to provide a page table for Process1 that has been freed. Only show valid page table entries.

[8 marks]

12.

a) What main issue does virtual memory help you solve?

[1 mark]

b) I have 3 frames of physical memory and the following reference string of (virtual) page addresses.

0, 2, 1

These references simulate a running process that generates these addresses. Suppose I apply this reference string on an initially empty virtual memory of 3 frames then I will end up with physical memory that looks like:

Physical Memory:

Frame Number	Page Number
0	0
1	2
2	1

Suppose I applied the following reference string instead to the same virtual memory (of three frames).

0, 2, 1, 3, 1, 0, 3

Give the corresponding physical memory table after the reference string is applied. Assume that the memory is initially empty. Use a least recently used (LRU) method for choosing victims for page replacement. Show your working for part marks.

[5 marks]

c) If I executed the same reference string as question b) on a bigger physical memory of 4 frames (again starting from an empty memory), would I have more page faults? Why or why not?

[2 marks]

d) What does thrashing mean in the context of virtual memory? When does it occur?

[2 Marks]