Surname: ....**Forenames:**..

STUDENT ID:...

THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2011
Campus: City

COMPUTER SCIENCE & SOFTWARE ENGINEERING

Operating Systems

(Time allowed: TWO hours)

NOTE:

Attempt ALL questions.

Answer the questions in the spaces provided.

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

For markers only:

Question 1	/12	*Question 5*	*/12*
Question 2	/14	*Question 6*	*/12*
Question 3	/10	*Question 7*	*/14*
Question 4	/14	*Question 8*	*/12*
		Total	

STUDENT ID:...

1. Concurrency & Deadlock [12 marks]

(a) Use the Dining Philosophers' Problem to demonstrate deadlock? Include a pseudo code solution to the Dining Philosophers' Problem which will likely cause deadlock.

We have a circle of philosophers with a fork shared between each adjacent pair of philosophers. The philosophers have to pick both forks (to the left and to the right of them) and each of them can pick up one fork. In this way every philosopher is also waiting for a fork and they are all deadlocked.

```
Each philosopher
      Thinks
      Picks up right fork
      Picks up left fork
      Eats
```

....

....

....

....

(5 marks)

(b) Briefly explain how deadlock can be detected.

We look for cycles in the wait-for graph. ....

....

....

(2 marks)

(c) Describe one way of dealing with deadlock when it is detected.

Kill or restart one or more of the processes involved in the cycle.................................

....

....

(2 marks)

(d) Give one reason why it is more difficult to detect deadlock in a distributed system than in a single machine system.

The information gathered from one node may be out of date by the time it is used by the deadlock detection mechanism...

....

....

....

....

(3 marks)

STUDENT ID: ....

2. File Systems [14 marks]

(a) Explain the difference between soft links and hard links to a file.

A soft link is a special type of file (usually a text file) which contains a pointer to the original file (usually just its full or relative pathname). The file system uses the soft link to find the real file.
A hard link is another name for the file. The actual file information is stored somewhere else, such as the Master File Table or the inode table, and all hard links refer to that information. The file information must hold a count of how many hard links there are to the file.

(6 marks)

(b) Give an example when the use of a hard link is preferable to a soft link?

When we want to avoid the problem of a broken link when the original file gets moved.

(2 marks)

(c) Give an example when the use of a soft link is preferable to a hard link?

When the link is to a file on a different device or volume. Or the link is to a directory.

(2 marks)

STUDENT ID:..

(d) NTFS stores the information about the location of file data on a disk using extents. Describe what "extents" are and explain how a particular cluster or block in a file can be located on disk from the information in the list of extents.

An extent consists of a starting cluster number and a number of contiguous clusters. To find a particular cluster we traverse the list of extents, adding the size of the extents as we go. When the total first exceeds the particular cluster number we are searching for we have found the extent holding the cluster. The previous cluster total is subtracted from the cluster number we are looking for and this gives the offset into the cluster. We then add the offset to the starting cluster number of the extent to find the actual disk cluster number...
....
....
....

(4 marks)

STUDENT ID:...

3. Distributed File Systems [10 marks]

(a) What is location transparency and why is it a desirable property for a distributed file system?

Location transparency means that the location of a resource or file is not related to its name. It is desirable because users and programs should be able to locate files without having to know what server they are on. It also makes it possible to migrate the resource to another location without breaking programs which refer to it. ....

(4 marks)

(b) Briefly summarise two different ways the location of a file can be discovered from the file name if a location transparent file system is distributed over a number of machines on a network.

The information could be kept locally on each machine, as in NFS. When parsing the pathname some directories are flagged as existing remotely and the particular server is found by consulting maps on the client.

The other main approach is to use a location database running on the network which can be queried for any file or resource which is not local. ....

(6 marks)

STUDENT ID: ....

4. Memory Management [14 marks]

(a) Using virtual memory slows down the effective access time, EAT, of memory for our processes. In lectures we saw two different ways in which access to a particular address in memory can be slowed down compared to a non-virtual memory system. Describe the two different ways and relate both of them to the page table information.

1. The page table information needs to be accessed for every memory access. This either leads to two (or more) memory reads for every memory access or else requires a cache (the TLB) to hold the page table information for recently accessed pages.
2. The page may not be resident in RAM but on disk. In order to access data on the page it has to be brought into a frame of real memory. The information about whether the page is in RAM or on disk is stored in the page table entry.

(6 marks)

(b) What is locality of reference and how does it help to reduce the effective access time in both situations?

In this case locality of reference means that most memory accesses occur close to a recent memory access. This is true for both code and data.
It reduces the EAT in the case of the TLB by the fact that a small number of pages are being used by a process over a short period of time and the page table information for these pages can all be held in the TLB, reducing the chances of a cache miss.
It reduces the EAT for the real vs disk problem by the fact that the small number of pages that are currently being used have been moved into real memory and this reduces the number of page faults.

(4 marks)

STUDENT ID:..

(c) The following diagrams represent a page access reference string on a machine with four frames of real memory. Complete the diagrams by filling in the frames according to the specified page replacement algorithms. Show the contents of every frame at each step of the reference string. The first columns have been completed for you.

FIFO – first in first out

page request	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**4**	**3**
frame 0	1	1	1	1	5	5	5	5	5
frame 1		2	2	2	2	1	1	1	1
frame 2			3	3	3	3	2	2	2
frame 3				4	4	4	4	4	3

(2 marks)

LRU – least recently used

page request	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**4**	**3**
frame 0	1	1	1	1	5	5	5	5	3
frame 1		2	2	2	2	1	1	1	1
frame 2			3	3	3	3	2	2	2
frame 3				4	4	4	4	4	4

(2 marks)

STUDENT ID:...

5. Assignment 2 [12 marks]

(a) The file synchronization mechanism used in assignment 2 tried to work without relying on the modification times associated with the files. Give two reasons why file modification times were avoided.

File modification times are unreliable. Different computers have different times.
Also file modification times can be changed easily by programs or the user. ....
....
....
....
....

(4 marks)

(b) Here are two sync files from a correctly running solution to assignment 2 just before the merge operation. What should happen in each directory when merging the two directories associated with these two files? What should happen to the modification times of the files?

Sync file from directory 1:

```
{
  "file1.txt": [
    [
      "09f7e02f1290be211da707a266f153b3",
      "2011-10-06 17:09:35 +1300"
    ],
    [
      "deleted",
      "2011-10-06 17:12:46 +1300"
    ]
  ],
  "file2.txt": [
    [
      "03342af7073c03024a1423900602e60e",
      "2011-10-06 17:10:37 +1300"
    ]
  ]
}
```

STUDENT ID:...

Sync file from directory 2:


```
{
  "file1.txt": [
    [
      "09f7e02f1290be211da707a266f153b3",
      "2011-10-06 17:09:35 +1300"
    ]
  ],
  "file2.txt": [
    [
      "03342af7073c03024a1423900602e60e",
      "2011-10-06 17:10:37 +1300"
    ],
    [
      "012ed860b18633488383dd8652748211",
      "2011-10-06 17:11:17 +1300"
    ]
  ]
}
```


file1.txt should be deleted from directory 2 and file2.txt should be copied from directory2 to directory1, overwriting the current version of the file. The modification time of file2.txt in directory1 should be the same as it appears in directory2. ....

(4 marks)

(c) A versioning file system would provide extra information to the synchronization process. What would this extra information be and how could it be used to make the process better? Remember that the sync command in the assignment was only called at certain times; it did not run continuously in the background.

The versioning file system would provide information about all changes performed on the file (at least when the file was saved). This could be used by the synchronization process to examine all different versions of the files including those modified between syncs. So the version in one directory might be an older version of a version in the other directory which would not have been visible otherwise. ....

(4 marks)

STUDENT ID:...

6. Assignment 3 [12 marks]

(a) Here are the summarised contents of the /proc/pid/maps file for A3Program1 (where pid represents the process id of the A3Program1 process):

```
00400000-00402000 r-xp 00000000 08:01 2497839  /home/robert/.../A3Program1
00601000-00602000 r--p 00001000 08:01 2497839  /home/robert/.../A3Program1
00602000-00603000 rw-p 00002000 08:01 2497839  /home/robert/.../A3Program1
010d8000-010f9000 rw-p 00000000 00:00 0        [heap]
7f5b91ddf000-7f5b91f69000 r-xp 00000000 08:01 2625263 /lib/.../libc-2.13.so
7f5b91f69000-7f5b92168000 ---p 0018a000 08:01 2625263 /lib/.../libc-2.13.so
7f5b92168000-7f5b9216c000 r--p 00189000 08:01 2625263 /lib/.../libc-2.13.so
7f5b9216c000-7f5b9216d000 rw-p 0018d000 08:01 2625263 /lib/.../libc-2.13.so
7f5b9216d000-7f5b92173000 rw-p 00000000 00:00 0
7f5b92173000-7f5b92194000 r-xp 00000000 08:01 2625250 /lib/.../ld-2.13.so
7f5b9237f000-7f5b92382000 rw-p 00000000 00:00 0
7f5b9238f000-7f5b92393000 rw-p 00000000 00:00 0
7f5b92393000-7f5b92394000 r--p 00020000 08:01 2625250 /lib/.../ld-2.13.so
7f5b92394000-7f5b92396000 rw-p 00021000 08:01 2625250 /lib/.../ld-2.13.so
7fffe308e000-7fffe30b0000 rw-p 00000000 00:00 0       [stack]
```

Given this information which of the following statements must be incorrect? In each case explain why.

```
End of code segment:                  0x402e26
Address in the code segment:          0x400980
Address of a string literal:          0x400f03
End of initialized data segment:      0x602094
Address of initialized variable:      0x603090
End of uninitialized data segment:    0x6020b0
Address of uninitialized variable:    0x6020a8
End of the heap:                      0x10f9000
Address on the stack:                 0x7f5b92394000
Command arguments address (argv):     0x7fffe30ae2a8
Environment variables address (envp): 0x7fffe30ae2b8
The last environment variable address: 0x7fffe30b0010
```

The end of code segment - the address is outside the executable section of memory
The address of initialized variable – the address is outside allocated memory
Address on the stack – the address is inside ld-2.13.so
The last environment variable address – this is off the bottom of the stack..................
....
....
....
....
....
....
....

(8 marks)

STUDENT ID:...

(b) This is the output of a program which prints the value of the end of the heap, then calls malloc(1) and prints the value of the new end of the heap and reports how many bytes the heap moved by.

```
End of heap: 0x170c000

Number of bytes malloced before the heap moved: 1

Number of bytes the heap moved: 135168 = 0x21000

End of heap: 0x172d000
```

Explain why allocating one byte with malloc has caused the heap to move by over 130,000 bytes.

Malloc requests large amounts of memory from the operating system and then allocates that memory to the process as it is called upon. In this case there was no memory left in the malloc buffer when the call was made, and so 135168 bytes were allocated by the operating system.. ....

(4 marks)

STUDENT ID:...

7. Protection and Security [14 marks]

Consider the following access control matrix for a system with two file objects (F1, F2) and three domain objects (D1, D2, D3).

D/O	F1	F2	D1	D2	D3
D1	Read	–	Control	–	Switch
D2	–	Owner	–	–	–
D3	–	–	–	Switch	–

(a) Can a process initially running in domain D1 write to file F2? If yes, explain how, if no explain why not.

Yes. The process can switch to D3 and then to D2 where it is the owner of F2 and so it could allocate itself the write permission.

(3 marks)

(b) Explain how a process running in domain D1 can stop any process in D1 from reading file F1.

It can remove the read permission on F1 because it has the control permission over D1. This means it can remove any permission in the domain.

(2 marks)

STUDENT ID:..

(c) With reference to an access matrix explain what a capability list is. What does each element of a capability list look like?

A capability list is a row of the access matrix, i.e. it is a list of all the objects and the corresponding privileges which a particular domain is allowed to use.
Each element of a capability list is <object, privileges>. ....
....
....
....
....
....
....
....

(4 marks)

(d) What is difficult about revoking (or removing) access privileges from a capability based system?

Because capabilities may be spread throughout a system and associated with domains it is difficult to keep track of them, especially if a capability can be passed from one domain to another.
In the case where a capability is stored in the user space of a process, such as with encrypted capabilities, it can't be removed by the protection system. ....
....
....
....
....
....

(3 marks)

(e) Describe one way in which capabilities could be revoked.

One of:
Keeping track of all capabilities and being able to remove them.
Using indirection, so that capabilities point to a pointer to the real resource and changing that pointer.
Time outs. Only allocating capabilities with time limits. ....
....
....
....
....

(2 marks)

STUDENT ID:...

8. Devices [12 marks]

(a) Briefly describe how a device driver can be written in user mode. Mention any services the operating system needs to provide in order to make this work.

The privileges required to access the device, either from IO ports or from memory addresses must be allocated to the user mode driver. In the case of memory-mapped devices the devices registers are mapped into read/writable memory allocated to the driver. The operating system needs to provide the functions to allow these things to happen. For some devices it needs to be able to reserve DMA and lock areas of real memory.

(4 marks)

(b) Unix separates devices into two main categories, block devices and character devices. Explain the main differences between these two types of devices and give an example of each type.

The main difference is that block devices use the block-buffer cache for communication with the device. IO is carried out in block sized chunks via these buffers. Whereas with character devices IO is usually unbuffered and is written to or read from the device immediately.

Block - disks, tapes, memory regions, etc

Character - keyboard, terminal (tty), mouse, anything USB, etc

(6 marks)

STUDENT ID: ....

(c) Where is the information stored which specifies a device is either a character device or a block device?

In the device's inode. It is visible when you list the device file from a directory as either a "c" or a "b". ....
....
....

(2 marks)