Question/Answer Sheet COMPSCI 340 & SOFTENG 370

Surname: .... Forenames:..

ID: ....

THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2010
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	*/13*	*Question 7*	*/10*
Question 2	*/12*	*Question 8*	*/7*
Question 3	*/8*	*Question 9*	*/12*
Question 4	*/8*	*Question 10*	*/6*
Question 5	*/15*	*Question 11*	*/4*
Question 6	*/5*	*Total*	

ID: ....

1. Concurrency [13 marks]

(a) Briefly describe the readers-writers problem.

There is some shared memory which some processes want to read from and some want to write to. Multiple readers may share the resource but only one writer at a time is allowed access to the resource. If a writer is accessing the resource no other process (reader or writer) may have concurrent access.

We also want to ensure that how we solve this ensures that neither readers nor writers starve...

....

....

(4 marks)

(b) One solution to the readers-writers problem is to give readers preference. What is a problem with this solution? When would this solution be acceptable?

Writers can starve. ....

This would be acceptable if there were few requests from readers or at least regular intervals when no readers arrived. This way the writing processes would not be delayed for long...

. ...

....

(3 marks)

(c) What is distributed shared memory (DSM)?

Shared memory running on separate machines. The sharing is done by copying data over the network between the machines but the programs are written as if they were sharing memory on the same machine. ....

....

....

(2 marks)

(d) Describe a problem of concurrent access to the same area of distributed shared memory.

Two processes on different machine may want to write to the shared memory simultaneously. If this is shared at the page level, they will write to their local copies and one of the writes may be completely overridden by another when sent to the originating memory, breaking the illusion of truly shared memory. ....

....

....

(2 marks)

ID: ....

(e) The concurrent access problem of distributed shared memory is simplified if we implement a readers-writers solution over the shared memory. Explain how this helps.

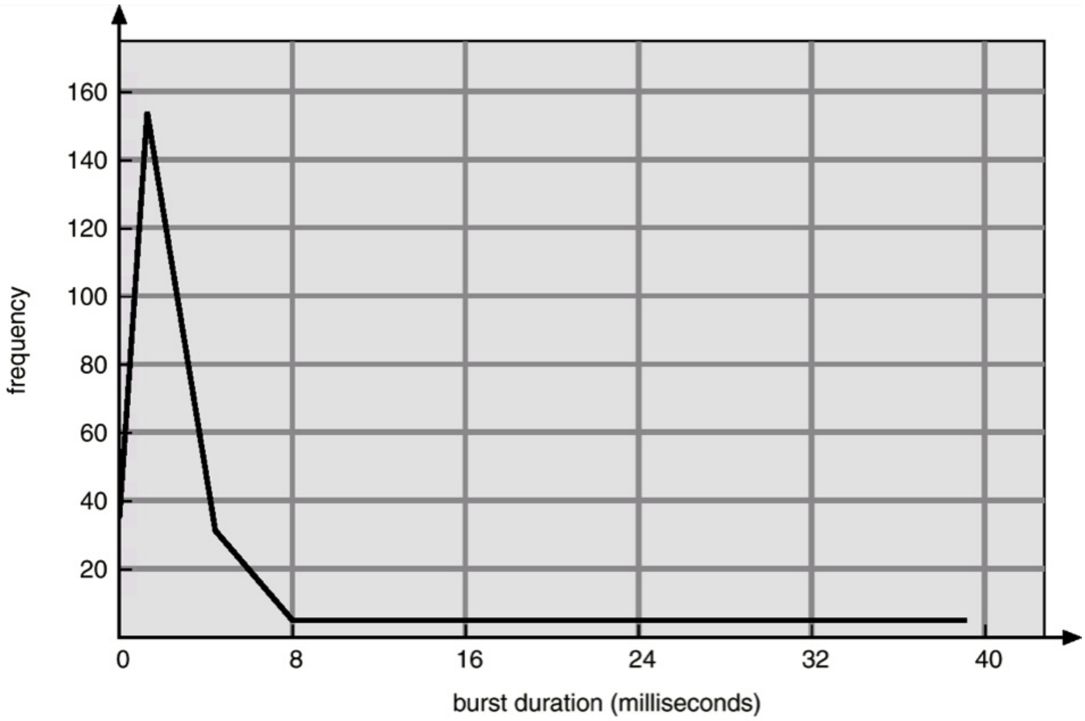
By only allowing one process at a time to have write access to the shared memory we prevent any concurrency problems. Also if we have no reads whilst a process is writing to the shared memory then no readers see inconsistent data. ....
....
....

(2 marks)

ID: ....

2. Scheduling [12 marks]

The following chart comes from the textbook and shows the frequencies of CPU burst times.



(a) What is a CPU burst?

The amount of time a process or thread uses the CPU before waiting or finishing. ......
....
....

(1 mark)

(b) In referring to the chart the textbook states "This distribution can be important in the selection of an appropriate CPU scheduling algorithm." Explain how this is so?

We want the scheduling algorithm to allow processes to run without unnecessary interruption from the scheduler and without causing unnecessary waiting for processes currently not running...

The CPU burst distribution shows us how long processes would run before waiting for some resource. If we give processes time slices which are slightly longer than most CPU bursts we can minimise context switches and still ensure the progression of all processes. ....
....
....

(3 marks)

CONTINUED

(c) In periodic real-time scheduling we commonly use the triple (c, p, d) to describe a process. Explain what the three letters mean. Which of the three is most closely related to CPU burst time?

C – computation time, how long the process will run in this period. ....
P – the period, how long before the process needs to repeat. ....
D – the deadline, when after the start of the period the process must complete, this is commonly the end of the period. ....
C is most closely related to CPU burst time, it is how long the process will run before stopping. ....
....

(4 marks)

(d) Given the following real-time processes calculate a cyclic schedule using Least Slack Time. If the slack times are the same, do NOT unnecessarily pre-empt the running process. If the slack times are the same for a number of non-running processes, choose the alphabetically lowest. e.g. If at time 9 both Process B and Process C have the same amount of slack time and neither process was running at time 8 then choose B. Show the schedule as a Gantt chart.

Process A(1, 6, 6) Process B(1, 3, 3) Process C(2, 4, 4)

B	C	C	A	B	C	C	B	C	C	A	B
0	1	2	3	4	5	6	7	8	9	10	11

(4 marks)

3. Deadlock [8 marks]

Havender's four conditions for deadlock are:

I. There is a circular list of processes each wanting a resource owned by another process in the list.
II. Resources cannot be shared.
III. Only the owner can release a resource.
IV. A process can hold a resource while requesting another.

For the following conditions describe a method or technique to prevent deadlock by breaking that condition.

(a) Condition I.

You can break the circular list by enforcing an ordering on resource (or lock) allocation.. ....
Another approach is to enforce an ordering on processes and only allow processes to wait if they are waiting on an earlier process (or vice versa). ....
....
....

(3 marks)

(b) Condition II.

Make resources sharable. In practice this may be impossible but some resources can be shared if they are made virtual, e.g. spooling to a printer effectively allows the printer to be used by multiple processes simultaneously...
....

(2 marks)

(c) Condition IV.

Allocate all resources before a process begins.
Or allocate resources in groups. The resources in a group are enough for a process to proceed, if it needs to use resources from another group it must return the current group first. . ...
....
....
....

(3 marks)

ID: ....

4. Threads [8 marks]

(a) Threads can be implemented as system-level threads or as user-level threads. What is the main difference in implementation between a system-level thread and a user-level thread?

The operating system knows about system-level threads and hence can schedule them independently etc. With user-level threads the OS only schedules a process and the user-level thread library is responsible for scheduling the threads................................

. ...

....

....

(2 marks)

(b) Give two advantages of system-level threads?

Blocking system calls don't block all threads in a process...

On a multiprocessor different threads from the same process can be run on different cores or processors. ....

Each thread can be scheduled separately. ....

....

....

(2 marks)

(c) Some web browsers (such as Google Chrome) create a new process for each tab showing a web page. Other browsers create a new thread for each web page tab. Explain what advantage there is in creating processes rather than creating threads for a new web page. Refer to at least one difference between threads and processes.

Security and protection. Because each web page is running in a separate process there is little shared memory between the web pages. If there is a security problem in one web page it is isolated from the other web pages and from the main browser process. This is true for both malicious and accidental problems. Separate processes share much less memory than separate threads in the same process......................................

....

....

....

....

(4 marks)

ID: ....

5. Assignment 2 [15 marks]

(a) Some people did assignment 2 by creating the entire file system in main memory and dumping the whole thing to the `device_file` whenever any changes were made.
Explain why this would not be a realistic solution for a real file system.

The file system is limited by the size of primary memory..
Writes are very expensive because the whole "file system" has to go to the disk. ....
There are ways around this by delay writing but it is still expensive. ....
....
....

(4 marks)

(b) Show the output produced by the following program running on a correct solution to Assignment 2:

```
#include <stdlib.h>
#include "system.h"

int main(void) {
     char listing[256];
     format("exam");
     create("/dir/fileA");
     create("/dir/fileB");
     create("/dir/fileC");
     a2write("/dir/fileA", "aaa", 4);
     a2write("/dir/fileB", "bbbb", 5);
     list(listing, "/dir");
     puts(listing);
     return EXIT_SUCCESS;
}
```

```
/dir:
fileA:   4
fileB:   5
fileC:   0
```

....
....
....
....

(4 marks)

(c) What is the volume name of the disk created by the code in part (b)?

exam ....
....

(1 mark)

ID: ....

There was no open call to access files in Assignment 2. Files were written to and read from by including the filename in all write or read calls.

(d) What disadvantage does this technique have compared to using an open call? How could this disadvantage be minimised?

Every access has to parse the filename again; the filename could be different each time and it has to be parsed and compared with the directories in the file system. This could be minimised by caching (probably using a hashtable) the filenames and file information of recently accessed files so that new accesses don't have to repeat this process. ....
....
....

(4 marks)

(e) What advantage could this technique have over a normal open call which is only called once for many references to the file?

There could be better security as this technique could check the process' access rights as part of its pathname parsing for every read or write. This way if a process loses access rights the change would be reflected immediately in the behaviour of the file system and the running process – solving the TOCTTOU problem.
Of course if the caching optimisation mentioned in (d) is employed this advantage may not exist. ....
....
....

(2 marks)

6. Distributed File Systems [5 marks]

(a) Many distributed file systems cache file data locally. What is the major advantage of this file data caching and why does it work well?

Efficiency and speed. Once a file is cached locally there is no longer any need to send messages over the network for file access. ....
Accessing cached data from a remote file is just as fast as accessing data from a local file...
It works well because file accesses usually exhibit good locality of reference. ....
....
....
....
....

(3 marks)

(b) What is the major disadvantage with caching compared to remote service?

Consistency is harder to maintain between multiple caches of the same file data.
Remote service does not suffer from this...
....
....

(2 marks)

7. Devices [10 marks]

(a) A device controller includes the electronics and built-in software which controls a device. Over time many device driver functions have moved from kernel level code into device controllers. What advantages and disadvantages are there in moving functions into device controllers?

Advantages:
Faster.
OS independent, so the same functions work on different operating systems.
Hopefully less likely to have problems because of rigorous testing..............................
Disadvantages:..
Once something is in the device controller it is much harder to fix bugs or upgrade the function...
Less flexible, less control is given to the operating system. E.g. If the OS wants to implement a certain algorithm with data going to a device it may find that it either can't or that the controller insists on using its own algorithms which may conflict with the effort of the OS..
....
....
....

(6 marks)

(b) What characteristics of solid state disk devices mean that they should be scheduled differently from traditional disk devices?

Random reads are just as fast as sequential reads so there is no advantage in carefully scheduling reads. ....
Writes are done in large groups to minimise the overwrite problem. Otherwise a read/erase/write is required on flash memory. ....
....
....
....
....

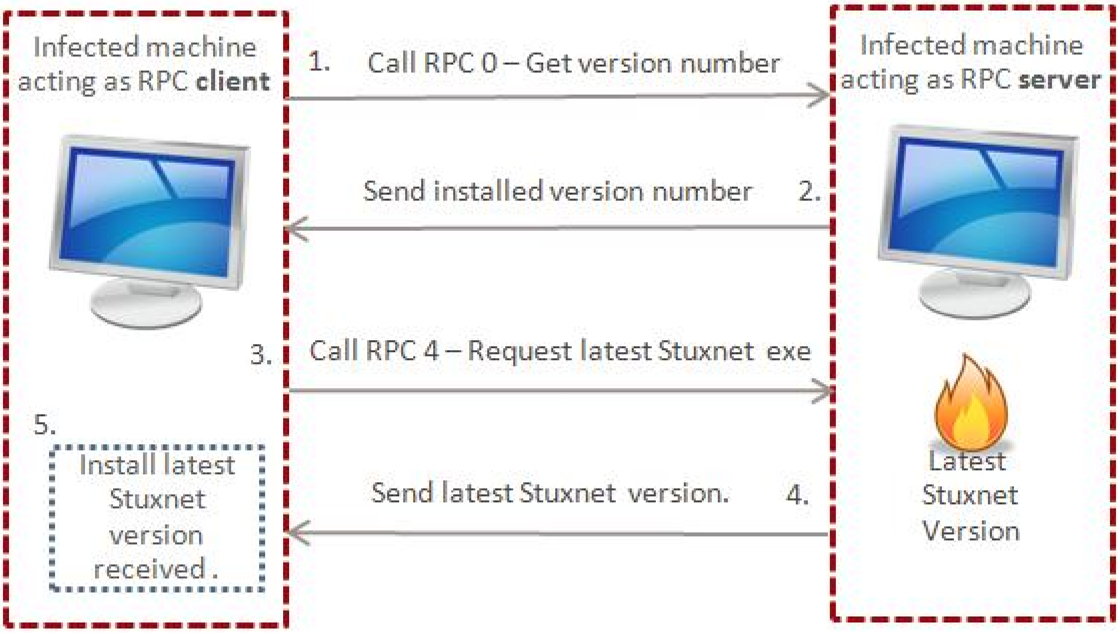
(4 marks)

ID: ....

8. RPC [7 marks]

The RPC server for the Stuxnet worm offers 10 services, listed below.

0. Version( ): returns the version number of Stuxnet
1. Execute(*x*): execute *x*, where *x* is a binary executable
2. Load(*m, a*): load module m into memory at address *a*, so that it can be called by an executable
3. Lsass(*x, o*): inject binary code *x* into lsass.exe (the Local Security Authority Subsystem Service of a Windows OS), at offset *o*
4. Build( ): builds the latest version of Stuxnet, returning it
5. CreateProcess(*c)*: create a shell process, with command line input *c*
6. Read(*f*): read a file, return contents
7. Drop(*f*): put file into drop area, for P2P filesharing
8. Delete(*f*): delete file
9. Write(*d, f*): write data *d* to file *f*


Infected machine acting as RPC **client**
1. Call RPC 0 – Get version number
Infected machine acting as RPC **server**
Send installed version number 2.
3. Call RPC 4 – Request latest Stuxnet exe
5.
Install latest Stuxnet version received .
Send latest Stuxnet version. 4.
Latest Stuxnet Version

A sample client-server interaction in Stuxnet is shown above.

ID: ....

(a) The server in your RPC system of Assignment 3 offered only one service. Explain how you could extend your RPC system, so that its server offers a service similar to the CreateProcess(c) service of the Stuxnet RPC. In your answer, you should also explain how a client in your system could request this new service.

My client process could be modified to write a tuple of the form [“createprocess”, DRb.uri, c] to the tuplespace of my system, using ts.write(). Using ts.take(), my server process could retrieve any tuple whose first element is “createprocess”, then execute (in a subshell) the string c in its third element. ....

....

....

....

....

....

(4 marks)

(b) Name, and briefly describe, one feature that is typically provided by an RPC which is ***not*** provided by the Stuxnet RPC (as described above).

Possible answers: The Stuxnet RPC does not appear to have any provision for time-stamping messages. The Stuxnet RPC does not appear to have any provision for marshalling and unmarshalling parameters in different machine representations (i.e. little-endian). The Stuxnet RPC probably doesn’t have an RPC protocol version number in each service request, although it does have a provision for updating the server code. ....

....

....

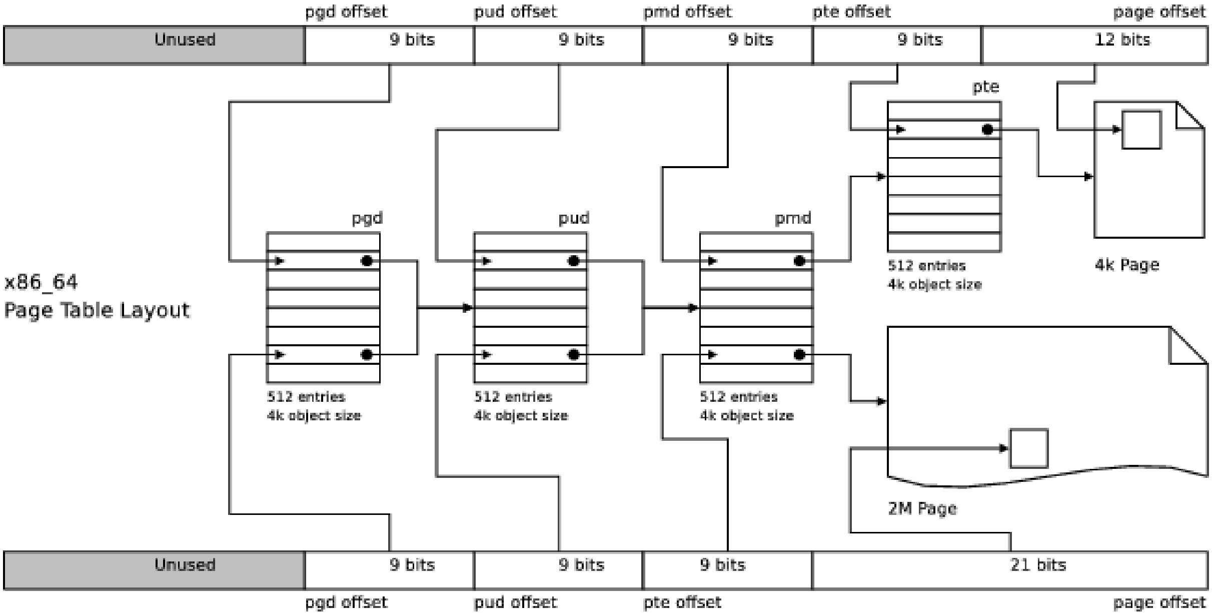
....

(3 marks)

ID: ....

9. Virtual Memory [12 marks]

The current version of the Linux x86 memory manager maintains a 4-level page table, with the structure shown in the figure below.



(a) Can you determine the size of the virtual address space of a Linux x86 process, by referring to this figure? If not, explain why. If so, show your work, by first expressing your answer as a power of two, then simplifying to terabytes (TB).

According to this figure, the virtual address space is 2^(9+9+9+9+12) = 2^(48). 2^(40) = 1 TB. 2^8 = 256. So the virtual address space is 256 TB. (Actually, user processes are limited to 128 TB, because half of the address space is reserved for the system.). ....

....

....

....

....

(3 marks)

CONTINUED

(b) Can you determine the size of the physical address space of a Linux x86 system, by referring to this figure? If not, explain why. If so, show your work, by first expressing your answer as a power of two, then simplifying to terabytes (TB).

This figure does not show how much physical memory is installed in the system. Furthermore, the figure does not show the width of the PTE field that contains a main-memory pointer, so we cannot determine the maximum amount of physical memory that could be addressed by these page tables...

....

....

....

....

(3 marks)

(c) Prior to 2006, Linux used only 3-level page tables with a 9:9:9:12 format. Briefly discuss one advantage, and one disadvantage, of the 4-level format in comparison with the old 3-level format.

The older page tables could address only 2^(39) bytes – less than 1 TB. Although few PCs have more than 1 TB of physical memory, programmers occasionally find it convenient to memory-map very large datasets (e.g. a DVD, or a matrix for a large scientific computation), so the 4-level page table is advantageous for such large- memory programs. The disadvantage is that the worst-case access time (when there are page-table faults) is increased slightly, from 3 faults to 4. Another disadvantage is that there is one more page table to set up when a process is forked: this will take a little more time.. ....

....

....

(3 marks)

(d) If a TLB lookup takes 0.5 ns, a memory read takes 10 ns, and the TLB hit rate is 90%, what is the effective access time (EAT)? Assume there is no TLB miss on the PTE read. Show your work.

EAT = 0.5 ns + 10 ns + (100% - 90%)(10 ns) = 10.5 ns + 1 ns = 11.5 ns

(The formula in the lecture slides is EAT = 2(10 ns) – (90%)(10 ns) + 0.5 ns = 11.5 ns.)...

....

....

(3 marks)

ID: ....

10. Access Control [6 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	–	–	–	Switch	Switch
D2	Read	Owner, Write	–	–	Switch
D3	Owner, Write	Read	–	–	–

(a) If process P1 is in domain D1, can it write to file F2? Explain.

Yes. A process in D1 can switch to D2, then write to F2.

(3 marks)

(b) If the access control matrix is stored in object A, what privileges should domains D1, D2, and D3 have for A? Explain briefly.

D1 is the most appropriate owner for object A. The reference monitor doesn't need to read or write files, but must be able to read the access-control matrix and other kernel objects. The other domains should have neither read nor write privilege for A.

(3 marks)

ID: ....

11. Security [4 marks]

(a) A Kerberos system has four security principals: an authentication server KAS, a ticket-granting server TGS, a user Alice (A), and a service provider Bob (B). There are two types of tickets in this system. Describe them briefly, indicating which principal creates the ticket, and which other principal(s) use the ticket.

The ticket-granting ticket TGT is issued by the KAS to Alice. Alice uses the ticket to authenticate with the TGS. The TGS issues a service ticket to Alice, whenever she authenticates her identity and requests a service for which she is authorised; Alice uses the service ticket to authenticate herself to Bob, as an authorised user of his service..
....
....
....
....
....

(4 marks)