TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI

VICTORIA
UNIVERSITY OF WELLINGTON

EXAMINATIONS — 2008
END-YEAR

COMP305

OPERATING SYSTEMS

Time allowed: THREE HOURS

Instructions: The examination contains 5 questions, you must answer all questions

Each question is worth 30 marks.

The exam consists of 150 marks in total.

Paper foreign to English language dictionaries are allowed.

Electronic dictionaries and programmable calculators are not allowed.

Question 1 Processes, Threading and Synchronisation **[30 marks]**

(a) Provide a short explanation for each of the following terms:

i. [2 Marks] A Process,

ii. [2 Marks] A Kernel Thread, and

iii. [2 Marks] A Context Switch.

(b) [5 Marks] Outline a typical sequence of steps an OS goes through during a **FORK** system call.
L4 page 5.

(b) Provide:

i. [2 Marks] An advantage that user level threads have over processes, and
Lightweight (little state), easier IPC (well ITC) and does not need calls to kernel to CS.

ii. [2 Marks] A major disadvantage of user level threads.
Blocking

(c) Briefly outline:

i. [2 Marks] a programming scenario that suits the use of kernel threads, and
Web, file server etc is a good one for this.

ii. [3 Marks] how kernel threads would be used in your scenario.
Use a thread per client request typically from a thread pool.

(d) Synchronisation

i. [2 Marks] What is a critical section (CS)?
A CS is a section of code in which improper concurrency control may result in corruption of shared data (or structure).

ii. [3 Marks] State and briefly explain the three necessary and sufficient conditions for a solution to the CS problem.
Mutual exclusion, progress and bounded wait.

iii. [3 Marks] Consider a two-process concurrency scheme that uses strict alternation. For each of the criteria outlined in question (d) ii. indicate if that criteria is satisfied and if it is not satisfied explain why not.
Progress not met: threads that are not in entry and exit sections have control over who enters next.

Question 2 Scheduling **[30 marks]**

(a) [2 Marks] What is the optimal CPU scheduling algorithm, and why is it not normally possible to implement.
SJF, we don't have a perfect oracle

(b) [4 Marks] Explain, with the aid of an annotated, where the scheduling opportunities lie during the lifecycle of a process.
On entry to ready queue, when doing IO and Disk.

(c) Consider the following table of processes and their arrival times:

Process	Arrival Time	CPU Burst
1	0	4
2	3	4
3	4	3
4	9	2

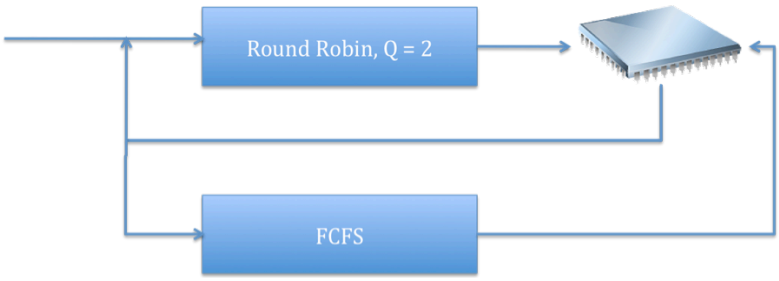
Table 1: Process arrival and burst times.

Draw a Gantt chart for the:

i. [4 Marks] FCFS scheduler. What is the average waiting time.
1 = 0, 2 = 1, 3 = 4, 4 = 2. Total wait = 7, av wait = 7/4

ii. [2 Marks] non pre-emptive SRTF scheduler. What is the average waiting time.
1 = 0, 2 = 4, 3 = 0, 4 = 2. Total wait = 6, av wait = 6/4

(d) Imagine we have a multi-level feedback queue with 2 queues. The highest priority queue is a RR scheduler with a quantum of 2. The second priority queue runs as a FCFS queue.



Processes start in the RR queue and are demoted to the FCFS queue if they exceed their quantum. Processes in the RR queue are always prioritised over the FCFS queue. Use the processes from Table 1.

i. [8 Marks] Show the state of the queues at time intervals of 2 from time 0 to the finish.

ii. [2 Marks] What is the average waiting time?

(e) Disk Scheduling

i. [2 Marks] What physical limitation of Hard Drives is disk scheduling intended to minimise?
Seek times

ii. [4 Marks] For the following cylinder sequence 87, 6, 158, 49, 487, 576, 588, 267, 12, what is the total head movement for the **scan** scheduling algorithm with start position 300 and seeking initially in increasing cylinder number.

iii. [2 Marks] What is the problem with the scan algorithm? Suggest a remedy.
Long potential wait time (2 x disk traverse). Remedy = C-Scan

Question 3 Memory Management **[30 marks]**

(a) [3 Marks] Explain the difference between compile time, load time and execution time binding, specifically talk about the memory addresses assigned to the program.
Compile time, program must reside in same address space, must be recompiled if address changes
Load time, once loaded cant be moved, slightly more flexible
Execution, program can move during execution

(b) [2 Marks] Summarise the difference between logical and physical addresses.
Logical is program view 0...n (n max program memory)
Physical is system view 0..end of memory

(c) [6 Marks] What is a Translation Lookaside Buffer (TLB)? Outline how the TLB works using a diagram describing what happens in the event of a 'hit' and a 'miss'.
High speed hardware lookup cache

(d) [5 Marks] Show, with the aid of a diagram, logical to physical address binding using a Hashed page table.

(e) [4 Marks]What is segmentation and what benefit does it provide?
Alternative to paging model, 2 D view of mem, abstraction over physical
Fits user view of a program, nice abstraction above memory

(f) [5 Marks] Describe the process of logical to physical address translation when using segmentation.
Address by segment name, and offset
Uses seg table to translate
Segs stored non-contig in phys memory although individual segments stored as contig

(g) [5 Marks] How is segmentation different from paging in terms of addressing and translation between logical and physical addresses
Variable size logical chunks
Address
Seg each add - > segname and offset
Paging only a single address HW turns this into page no and offset

Question 4 File Systems **[30 marks]**

(a) [3 Marks] Describe the role that a Virtual File System plays in a system.
VFS allows the same system call interface (the API) to be used for different types of file systems.

(b) [5 Marks] Outline, using a diagram, a typical sequence of steps an OS goes through during a **Open** file system call.
L16 P6.

(c) [4 Marks] What is meant by a **linked** file allocation scheme?
Essentially a linked list on the disk, each sector indicates the next sector in the file. Non contig.

(d) [2 Marks] What is the primary drawback of linked file allocation schemes for hard drives?
Random access cannot be supported efficiently.

(e) [5 Marks] Describe the structure of a unix inode.
Some time stamps, owner, etc. 12 direct pointers, 1 single, 1 double and 1 triple.

(f) [4 Marks] Given a block size of 1K, how many disk reads are needed to seek to a position 15K into a file. Note, there is no caching.
This is more than 12K, so it will be a single indirect. 1x inode, 1x single, 1x data block – so 3.

(g) [3 Marks] Draw a graph structured directory and explain how we can traverse this structure without suffering from cycles.
Use symbolic links and ignore during traversal.

(h) [4 Marks] Summarise the principle of a log structured file system.
Writes to the disk, are committed to a log. Writes then progress asynchronously and once written to disk removed from the log.

Question 6 Security **[30 marks]**

(a) [4 Marks] Define the principle of **least privilege**, give an example of a possible threat and show how the principle of least privilege can be applied.
Programs, users and systems should be given just enough privilege to perform their tasks, thus minimising damage done by compromised component
Example buffer overflow, overflow may cause a program to fail but should not allow execution of code from the process stack which would enable an unauthorized user to gain extra privileges.

(b) [4 Marks] Explain what Access Rights are and describe how they relate to Domains.
Access right is ability to execute a given operation on an object -> <object, access-set>
Domain contains a set of access rights relating to different objects

(c) [6 Marks] Describe an access control list and a capability list and provide one drawback of each.
Access Control List
List of access rights for each object corresponding to each domain.
Drawback – time to search, could get big with lots of domains. Determining rights for a domain is difficult.

Capability
List of objects and operations allowed on those objects for each domain.
Drawback – hard to revoke, hard to determine rights for a particular object

(d) [16 Marks] Consider the following access control matrix. The matrix describes the access rights for a simple system containing three domains (Administrators, Programmers, and Guests), three files and an executable program.

	File A	*File B*	*File C*	*Program P*
D1 (Admin)	*W*	*RW **	*R**	*Ex*
D2 (Programmer)	*RW*	*R*		*Ex*
D3 (Guest)	*W*			

i. [2 Marks] Which entities can *Read* File A?
Only a member of D2, i.e a programmer because that is the only entry with an R in the A column

ii. [2 Marks] How could a Guest gain the appropriate *Read* privilege for File C?
If given permission by Admin, i.e the transfer of the R right

iii. [4 marks] Specify the access control lists to implement this access control matrix.
File A - (D1, W)(D2, RW)(D3,W)
File B
File C
Program P

iv. [4 Marks] If the Read right for File B has limited copy semantics, what would the matrix look like if the right is transferred from an Admin (D1) to a Guest (D3)
Delete the right from D1 and move to D3

v. [4 Marks] Imagine a new file (File D) was added to the system, allowing only the admin to have initial Write privileges. Is it possible for the admin to hand over the Write privilege to a Programmer such that the Programmer is now the only domain to have write priviilge on File D? Justify your answer.
Using Transfer copy semantics

* * * * * * * * * * * * * * * * * * * *