TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

EXAMINATIONS — 2010
END-YEAR

NWEN 301

OPERATING SYSTEMS DESIGN

Time allowed: THREE HOURS

Instructions: The examination contains 5 questions. You must answer ALL questions

Each question is worth 20 marks.

The exam consists of 100 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 [20 marks]

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

i. [2 Mark] A Thread,
There are two types, kernel and user. A user thread is supported (by libraries) at user level. Kernel is not aware of the user threads – and as such creation and scheduling is done outside the kernel. A kernel thread is managed by the kernel directly. Threads share most state, but have different stacks and PC.
ii. [2 Marks] A Light Weight Process.
An abstraction used to map n user threads on to m kernel threads.

(b) [2 Marks] Outline the lifecycle of a process.
L3, Slide 22

(c) Process Control Blocks (PCB)

i. [2 Marks] Why does an OS maintain a PCB structure?
To record state when context switching, and to record other 'useful' items such as open files, memory, scheduling information etc.
ii. [2 Marks] Outline a typical PCB.
L3, Slide 16.

(d) Synchronisation

i. [1 Marks] Explain the need for synchronisation.
To ensure data integrity/consistency. Serialises parallel operations.
ii. Consider the SWAP hardware primitive:
    a. [2 marks] explain how it is used as a synchronisation primitive, and
    L4, Slide 18
    b. [2 marks] for each of the conditions; mutual exclusion, progress and bounded wait, state whether your solution in 1.(d)ii.a. meets the condition or does not meet the condition. If it does not meet the condition, briefly explain how this could be achieved.
    Does not meet bounded wait. Would need a queue added to ensure this.
iii. [5 Marks] Explain Peterson's 2-process solution to synchronisation.
L4, Slide 13 & 14.

Question 2 Scheduling **[20 marks]**

(a) [3 Marks] What is the major difference between *multi-programmed* and *timesharing* systems? Be sure to note the impact this difference has on scheduling.

A multi-programmed system runs multiple programs concurrently to attempt to keep both the processor and I/O devices busy. Timesharing is an extension of multiprogramming in which the system switches between processes rapidly enough to allow user interaction. In a batch multi-programming environment it is sufficient to allocate the cpu to a process until it is released by an I/O request. In a timesharing system some form or round-robin quanta is required. 1 mark, MP, TS and the need for RR.

(b) Consider a system with one CPU and four jobs. Each job has an arrival time, burst time and priority as given in the following table. Priorities are ranked on a scale from 0 (lowest) to 127 (highest). REDO THESE NUMBERS

Job #	Arrival (t)	Burst (t)	Priority
1	0	6	60
2	3	8	70
3	10	7	80
4	9	3	122

i. [2 Marks] Draw a Gantt chart and then find the average waiting time when using a NON-preemptive SJF scheduling algorithm.
They execute in order of arrival: 1,2,4,3. The average waiting time is 3.75.

ii. [2 Marks] Draw a Gantt chart and then find the average turnaround time for a pre-emptive priority scheduling algorithm.
1 executes in periods 0-2 and then from 21-23, turnaround = 24;
2 executes in periods 3-8 and then from 19-20, turnaround = 18;
3 executes from 9-11, turnaround = 3;
4 arrives at 10 and executes in 12-18, turnaround = 9.
Average turnaround = 13.5.

iii. [6 Marks] Find the average waiting times for the RR scheduling algorithm, with quanta 4,6 and 8.
4.75, 5.5, 3.75s.

iv. [4 Marks] Explain your results for the RR scheduling algorithm as the quantum is changed in 2(b)iii.
LX, SX. Waiting time is not a linear relationship vs quanta size.

(c) [3 Marks] Explain the difference between a Dispatcher and a Scheduler.

Dispatcher determines which jobs from the batch are selected and loaded into the machine, the scheduler determines which process gets to run.

Question 3 Resource Allocation **[20 marks]**

(a) [2 Marks] List the 4 conditions necessary for deadlock to occur.
Mutual exclusion, hold and wait, no pre-emption and circular-wait.

(b) [2 Marks] For each of the conditions in 3(a) explain if elimination of the condition is sensible.
Mutual exclusion – no, hold and wait – usually, no pre-emption – sometimes, circular wait – yes.

(c) Consider a system with three processes (P1, P2, P3) and three resource types (R1 = 2, R2 = 2, R3 = 3), having the following state:

Process	Allocated			Requesting		
	R1	R2	R3	R1	R2	R3
P1	1	1	0	0	1	1
P2	1	0	1	0	1	0
P3	0	0	2	1	0	0

i. [3 Marks] Draw the resource allocation graph for this resource state.
Boxes and circles…

ii. [4 Marks] Is this system deadlocked? Be sure to explain your reasoning.
This is a little tricky. If P1 goes first and is allocated the spare instance of R2 then the system will indeed be deadlocked. However, the order in which the processes run is not defined, hence if the request from P2 is granted for R2, then P2 can complete, freeing 1 instance of R1 and 1 instance of R2 and 1 instance of R3. Now either P1 or P3 can complete.

(d) [3 Marks] Explain the difference between deadlock detection, avoidance and prevention.
Deadlock detection is identifying the problem and then resolving it. Avoidance is making sure allocations are only made such that deadlock cannot arise. Deadlock prevention is making sure it cannot happen in the first place – i.e. removing hold-and-wait.

(e) Consider a system with the following resource state:

Process	Allocated				Maximum			
	R1	R2	R3	R4	R1	R2	R3	R4
P1	0	0	1	1	0	0	1	2
P2	1	3	3	4	2	3	4	6
P3	1	0	0	0	1	7	5	0
P4	0	0	1	4	0	6	5	6
P5	0	6	3	2	0	6	5	2

Available			
R1	R2	R3	R4
1	5	3	1

For each of the following sequences of requests, indicate if the request can be granted. You will need to show your working, the intermediate steps and explain how you have arrived at your answer. **NOTE: questions i, ii, and iii, are independent and each start with the same state, as described in the table.**

i. [2 Marks] P1 requests (0,0,0,1), then P3 requests (0,1,3,0).
ii. [2 Marks] P4 requests (0,3,1,1), then P2 requests (1,0,1,0).
iii. [2 Marks] P5 requests (0,0,1,0), then P2 requests (1,0,1,1).

Question 4 File Systems **[20 marks]**

(a) [4 Marks] Outline the FAT file allocation scheme. Remark on how it resolves problems inherent in the plain linked allocation scheme.
FAT = file allocation table. A table of pointers showing the block links for each file. Essentially all it does is abstract out the linking information into a table small enough to cache in memory. This means that we can now efficiently perform random accesses into a file. Plain linked allocation requires us to read in the entire block before we can access the linking info.

(b) Assume we have a file system implemented with 32 bit file addresses and 8Kb blocks. It uses inodes with 12 direct blocks, 1 single indirect, 1 double indirect and 1 triple indirect block. Assume that any files are already open, but no references have yet been performed.

i. [4 Marks] Draw the inode structure and indicate the addressing ranges for each of the block pointers.
Some time stamps, owner, etc. 12 direct pointers, 1 single, 1 double and 1 triple.

ii. [3 Marks] How many disk accesses are required to read file location 53,248? Be sure to explain your reasoning.
File is open, so inode is cached (FCB). The direct blocks can reference 96K, so this is a direct reference. Therefore we only need one access (to read the data block).

iii. [3 Marks] How many disk accesses are required to read file location 28,311,552? Be sure to explain your reasoning.
File is open, so inode is cached. Direct blocks only go up to 96K. SID go from 96K to 16MB+96K. As this is 27MB we therefore need a DID block. So we have 1 read for the indirects, one read for the directs and 1 read for the data = 3.

(c) Given the following queue of disk accesses [120, 12, 56, 89, 134, 11, 3 ,4 ,16], a disk of 150 cylinders, a liner seek time of 0.8ms/cylinder, and a starting position of 75:

i. [3 Marks] Calculate the total time spent seeking for a FCFS schedule.
|75-120| + |120 - 12| + |12-56| + |56 - 89| +|89-134| + |134-11| + |11-3| + |3-4|+|4-16|
= 419 seeks and 33.52ms

ii. [3 Marks] Calculate the total time spent seeking for a SSTF schedule.
= 203 seeks and 16.24ms

Question 5 Memory **[20 marks]**

(a) [5 Marks] Explain what is meant by the terms external and internal fragmentation. Give examples of where external and internal fragmentation arise.
External fragmentation is when the memory is available, unallocated, but is too small to be useful. Internal fragmentation is when the memory has been allocated to a process, but is in excess of what was asked for/required. External fragmentation exists when you have variable sizes of memory being allocatable such as pure segmentation, internal fragmentation exists in schemes such as paging.

(b) [2 Marks] Explain why worst-fit minimises external fragmentation, whereas best-fit increases external fragmentation.
Worst fit chooses the memory fragment that leaves the largest remainder. The theory being that this larger left over will also be allocable, whereas best fit leaves lots of very small, unallocable fragements.

(c) [2 Marks] Why is it a good idea to map between logical and physical memory locations?
Because, this means that the program can potentially be placed in non-contiguous memory and therefore increases the ability of the OS to manage the memory resource.

(d) [5 Marks] Consider a system with 128 byte pages and 2^{16} addresses. How much memory would we need to set aside in the process to implement a plain hierarchical page-table?
We can fit 64 addresses in each page. To address 65536 locations we need 65536/128 pages = 512 pages. This is 8 pages worth of pagetable, so 1 parent and 8 children. Even though we only need a little bit of the parent (6 of 64 addresses) we still have to count it. Hence 9 pages of 128 bytes or 1152 bytes.

(e) [6 Marks] Provide a diagram showing how a TLB would work with segmentation.
Essentially identical to L14, S13&14. No real difference between segmentation and paging answers here.

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