

Name:

email:

ID:

Midterm Exam
CS402
26 Oct 1998

You have 1 hr. 20 min. for this exam. The exam has 6 pages. There are 100 possible points. Show all your work for partial credit.

Definitions

Each question includes the number of points. Answer all questions in this section.

1. Give a term that describes the following:

a) The four requirements for deadlock (4 pts)

Answer: Mutual exclusion, non-preemption, hold and wait, circular wait

b) A persistent, named collection of data (1 pt)

Answer: File

c) The set of addresses that a process may access (1 pt)

Answer: Address space

d) Two process scheduling disciplines (2 pts)

Answer: Pick 2 of FIFO, shortest job first, round robin, priority. There are others.

e) Two page replacement algorithms (2 pts)

Answer: Any 2 of FIFO, second chance, clock, wsclock, LRU, NRU, random, optimal

f) The OS data structure used to map between virtual pages and physical frames (1 pt)

Answer: Page table. TLB is not acceptable - it's hardware not a data structure.

g) Synchronization mechanism that is an abstraction of a counting number (non-negative integer) (1 pt)

Answer: Semaphore. I'll accept Event Counter, too, but not happily.

h) Portion of a processes code that manipulates shared variables (1 pt)

Answer: critical section

i) The OS data structure that represents a running process (1 pt)

Answer: Process control block

j) Moving an entire process's memory allocation out of memory to secondary storage (1 pt)

Answer: Swapping

Name:

email:

ID:

Short Answer

Each question is worth 10 points. Do all questions in this section.

2. Jimmy says that he has an algorithm that will allow him to perfectly pick the results of the coming week's professional football games if he knows how many points each team will score¹. This assertion is similar to assertions made by some of the scheduling algorithms we have discussed in class. Name 2 such algorithms (1 point for each) and explain how Jimmy's statement is similar to statements made about those algorithms. (4 points)

Answer: The scheduling discipline is similar to SJF or the optimal paging strategy in that it behaves perfectly given knowledge of the future.

Using your knowledge of how those scheduling algorithms are implemented in real systems, explain how Jimmy could implement his algorithm or an approximation of it. (4 points)

Answer: Jimmy could keep track of the reported scores in the paper and keep a weighted average of those scores as an estimate of the inputs to his algorithm.

3. In a single-CPU multiprogramming system, what is the difference between a process that memory-maps a file, makes some changes to the file in memory, and exits and a process that reads a file into its memory, makes some changes and writes those changes back to the file (4 points)?

Answer: The in the memory mapped file, the changes are immediately visible to other processes accessing the file, in the second case, the changes are not visible until the write is completed.

Describe a situation where the memory-mapped semantics are more convenient. Describe a situation where the open/read/write/close semantics are preferable. (3 points each)

Answer: The memory mapped semantics are preferable for applications that require tight synchronization of changing data, and where other processes may wish to see intermediate steps. For example a system that tracks users or that keeps track of multiple bank accounts. The open/read/write/close semantics are preferable where several mostly independent processes access the data, and partial updates are not acceptable. An example a program maintained by many people. Partial updates may result in code that doesn't compile, which is (usually) worse than code where whole feature updates are lost.

¹ In case you know nothing about football, all you need to know for this question is that the team that scores the most points wins, and the list of which teams play each other is publicly available. Jimmy's algorithm should not be hard to determine.

Name:

email:

ID:

4. The following jobs are queued to be executed on a timesharing system (they will enter the system at the same time):

Job	Run Time	Priority
A	10	1
B	3	2
C	5	4
D	7	3

The response time of a job is the total time between when the job enters the system and when it completes. What is the average response time of a job if the jobs are run using:

Run-to-completion, Shortest Job First (SJF)? (3 pts)

Answer: Total time is $3 + (3+5) + (3+5+7) + (3+5+7+10) = 51$ avg response is $51/4 = 12.75$

Run-to-completion, Priority (a job with priority 1 runs first)? (3 pts)

Answer: Total time is $10 + (10+3) + (10+3+7) + (10+3+7+5) = 68$ avg response is $68/4 = 17$

Preemptive multitasking, Round Robin, a time quantum \ll the run time, context switch time negligible? (3 pts)

Answer: Each job gets an equal share of the CPU until they complete, so while 4 jobs are running, each runs at $1/4$ the rate. While 3 are running at $1/3$, etc. You can go through and calculate the number of quanta if you want, but it is a 3 point problem.

total time is 12 (B completes at $1/4$ speed) + $(12+6)$ (C finishes its 2 remaining units at $1/3$ speed) + $(12+6+4)$ (D completes its last 2 units at $1/2$ speed) + $(12+6+4+3)$ (A completes its last 3 alone) = 77 avg response time = $77/4 = 19.25$

Which scheduling discipline has the lowest average response time? (1 pt)

Answer: SJF. You don't have to do any calculations to know that.

5. Consider a physical memory allocation system that uses base and limit registers for memory protection. The CPU has only one base and one limit register. If a process attempts to access a memory location outside the range specified by the base and limit registers, the CPU causes a trap into the operating system to handle the memory error. When the OS is called in this way, it has access to the CPU state at the time of the attempted access, and may restart the instruction.

Describe a method that will allow a process to access 2 or more non-contiguous areas of memory. I am only concerned with the issues of memory protection, not modifying the code or data of the process to take advantage of the ability to access the non-contiguous regions. Your description should include the algorithm followed by the OS when it receives a trap and any data structures that are used in the operation of your system. **You are not expected to write code for this question.** (10 pts)

Answer: The OS keeps a table (array) in the PCB of base/limit pairs that describe the various memory areas a process may access and a PC value that is the last faulting instruction and one more base/limit page that is the first base/limit pair tried for this instruction. On a memory fault, if this is the first time this instruction has faulted, it records the instruction and the current base limit pair in the PCB, then tries the next base/limit pair in the list. If the same instruction faults with all the possible base/limit pairs the process is terminated (otherwise it succeeded in accessing the data).

Long answer

Each question gives its value. Do all questions in this section.

6. Consider the following computer memory system. It consists of a CPU with a data cache (that stores data by virtual address), an MMU with a translation lookaside buffer, a page table, and a core map managed by the operating system. The core map reflects the contents of physical memory. The MMU maintains the used and dirty bits of the page table directly. The OS uses the clock page replacement algorithm. On the core map the clock hand is shown with an arrow (→). The data cache and the TLB are managed by the hardware using Least Recently Used (LRU). Only one process is currently using the machine. Memory addresses are expressed as virtual page:offset. Show the state of the various elements of the system after the following memory accesses. You only have to show what addresses reside in the data cache, not the contents of those addresses. (20 pts)

Data Cache	TLB	
Address	Page	Frame

Core Map		
Frame	Page	Process
0	3	P1
1	2	P1
2	1	P1
3	0	P1

Page Table				
Page	Frame	Use	Dirty	Valid
0	3	x		x
1	2		x	x
2	1			x
3	0	x		x
4				
5				
6				

- a) write 0:2

Data Cache	TLB	
Address	Page	Frame

Core Map		
Frame	Page	Process
0		P1
1		P1
2		P1
3		P1

Page Table				
Page	Frame	Use	Dirty	Valid
0				
1				
2				
3				
4				
5				
6				

Name:

email:

ID:

b) read 2:3

Data Cache	TLB	
Address	Page	Frame

Core Map		
Frame	Page	Process
0		P1
1		P1
2		P1
3		P1

Page Table				
Page	Frame	Use	Dirty	Valid
0				
1				
2				
3				
4				
5				
6				

c) write 4:6

Data Cache	TLB	
Address	Page	Frame

Core Map		
Frame	Page	Process
0		P1
1		P1
2		P1
3		P1

Page Table				
Page	Frame	Use	Dirty	Valid
0				
1				
2				
3				
4				
5				
6				

Answer:

Name:

email:

ID:

a) write 0:2

Data Cache	TLB	
Address	Page	Frame
0:2	0	3

→

Core Map		
Frame	Page	Process
0	3	P1
1	2	P1
2	1	P1
3	0	P1

Page Table

Page	Frame	Use	Dirty	Valid
0	3	x	x	x
1	2		x	x
2	1			x
3	0	x		x
4				
5				
6				

b) read 2:3

Data Cache	TLB	
Address	Page	Frame
0:2	0	3
2:3	2	1

→

Core Map		
Frame	Page	Process
0	3	P1
1	2	P1
2	1	P1
3	0	P1

Page Table

Page	Frame	Use	Dirty	Valid
0	3	x	x	x
1	2		x	x
2	1	x		x
3	0	x		x
4				
5				
6				

Name:

email:

ID:

c) write 4:6

Data Cache	TLB	
Address	Page	Frame
4:6	4	2
2:3	2	1

→

Core Map		
Frame	Page	Process
0	3	P1
1	2	P1
2	4	P1
3	0	P1

Page Table

Page	Frame	Use	Dirty	Valid
0	2	x	x	x
1				
2	1			x
3	0			x
4	2	x	x	x
5				
6				

7. You have been hired by NASA to study Operating Systems issues in their latest Mars probe. Consider the following situations:

- a) The probe has several interconnected fuel tanks. The tanks are connected in a ring and fuel can be pumped between adjacent tanks by any of the processes running on the probe. To avoid race conditions, processes use the following algorithm for moving fuel: lock the tank from which fuel is being moved; lock the tank to which fuel is being moved; move the fuel between tanks (increment the contents of one and decrement the contents of the other); release the locks in the reverse order. What is the most serious problem this solution has? (1 pts) Give an example illustrating the problem (2 pts). Suggest a way to fix it (4 pts).

Answer: The solution can deadlock. Consider 2 processes, p1 & p2, p1 is moving fuel from tank 1 \rightarrow 2 and p2 from 2 \rightarrow 1. p1 acquires lock1; context switch; p2 acquires lock 2; context switch p1 waits for lock 2; p2 waits for lock 1 - deadlock. You can fix it by numbering the tanks and acquiring them in order.

- b) This problem can be reduced to another problem that was discussed in class. Name the problem (2 pts), and show how the two problems are equivalent (4 pts).

Answer: This is the dining philosophers problem. The processes are philosophers and the forks are tanks.

- c) Because we can reduce the fuel problem to the class problem (the one discussed in class), we can use the known solution for that problem. Other than avoiding the problem you identified above, name an advantage that using the known solution has. (4 pts)

Answer: There are known solutions to the dining philosophers problem that get maximum concurrency - that is as many fuel transfers as possible will occur using the dining philis solution, as opposed to the numbering that may not.

- d) This probe is a self-contained system being sent far from its controllers into a dangerous and unpredictable environment with the intent of reporting interesting data back to its owners. Discuss 3 ways in which the **operating system** of this probe must differ from the operating system of a general purpose earthbound computer. (10 pts)

Answer:

- It's not connected to a reliable, effectively infinite amount of power. Power becomes a resource to be managed, like CPU time or disk space.
- It is not intended for interactive use, so optimizations for interactive use, like aging non-interactive jobs are unlikely to be helpful.
- Various control tasks on the probe may have hard deadlines. Scheduling mechanisms tuned for real-time deadlines are more appropriate.
- Either crises or interesting events may need to be reacted to quickly. A scheduling mechanism that allows certain tasks to have absolute priority is important.
- It's difficult to reboot a remote probe. The system must recognize that it's deadlocked and reboot itself. In general it needs to run with much less user intervention than a desktop computer.
- Probably not much use for sophisticated screen drivers or graphical interfaces.
- System components are likely to be fixed - new processes will be rare, as will new memory allocations. Schedulers and allocators can be simplified.