

ECE 350 W24 Midterm Exam Solutions

(1.1) To handle the SIGKILL:

- Set the state of the process to terminated
- Mark its allocated memory as free
- Close its open files
- Notify the parent process (SIGCHILD) it has terminated
- If the parent process has already waited for the child, remove the PCB
- If the process was executing, run the scheduler to choose another process to run

Note that the answer has to be specific about deallocation of the resources; it's not enough to just say clean up resources.

Any 5 of the 6 above is sufficient for full marks.

(1.2) Yes, we can store registers and the program counter in another place than the PCB. The important thing is that we store them in a place where we know where they are and can retrieve them again later. One possibility is to push them onto the stack for the executing thread, but the OS has complete control over how to manage this data.

(1.3) It's possible to have mandatory locks, but is very expensive in runtime costs. Valgrind's Helgrind tool, for example, can detect when there is a race condition and report it on the console. If instead of just printing it to the console, it terminated the program with something like a Segmentation Fault, that would be mandatory enforcement of the locks. Therefore the OS could easily do the same thing by a similar mechanism (track locks held, observe all memory accesses) and terminate the process if an invalid access is detected. However, the major tradeoff is that it would slow down your program a lot, and is subject to false positives (meaning a valid program could be terminated as a result).

Note: your argument doesn't have to reference Helgrind, but if you identify that it requires observing memory accesses and tracking locks held, that's fine.

(2.1)

1. When a memory access occurs, trying to find the physical address requires consulting the page table. If an access is invalid, we will not find a page for that given address.
2. If an invalid reference is detected, the OS sends a segmentation fault signal to the process.
3. The logical address will be split into the page and offset. The TLB is checked to find the frame number, and if it is not found, then the page table must be checked to find the frame number. However it is found, the frame number is combined with the offset to produce the physical address.
4. This is a speed vs. cost tradeoff; a larger TLB has a higher hit rate but costs more money.
5. No, a hierarchical page table would require an additional memory access for the 2nd level of the page table whenever the value is not found in the TLB. For the size of memory we have, this is not worth it.

(2.2)

First-In-First-Out (FIFO)						Second-Chance (Clock)					
F0	F1	F2	F3	Fault?	Total Faults	F0	F1	F2	F3	Fault?	Total Faults
8	–	–	–	Y	1	8	–	–	–	Y	1
8	7	–	–	Y	2	8	7	–	–	Y	2
8	7	3	–	Y	3	8	7	3	–	Y	3
8	7	3	1	Y	4	8	7	3	1	Y	4
5	7	3	1	Y	5	5	7	3	1	Y	5
5	7	3	1	N	5	5	7	3	1	N	5
5	9	3	1	Y	6	5	9	3	1	Y	6
5	9	3	1	N	6	5	9	3	1	N	6
5	9	7	1	Y	7	5	9	3	7	Y	7
5	9	7	6	Y	8	5	9	6	7	Y	8

(3.1)

[A] [A] [B] [A] [B] [A] [B] [-] [-] [C] [D] [C] [D] [C] [E] [C] [E] [C] [E] [F]

Your sequence could be different if you chose D to go first in place of C at $t = 10$, but

(3.2) Was this question inspired by the recent news articles about how difficult it has been to get tickets for Taylor Swift concerts? Yes, it was.

Part 1. If demand so vastly exceeds supply, there's no realistic possibility that every process gets enough CPU time. As long as the selection process is sufficiently fair, and FCFS is fair by some definition, it's okay if some processes don't get a turn. It's like there are not enough seats in a venue – not everyone will get a seat, but if the process for getting a seat is fair, then it doesn't violate the principle of fairness, even if some people are unhappy at the end of the day.

(Okay, we all know sales of concert tickets are not fair – the lawsuits make this clear – but they are supposed to be.)

Part 2. Your strategies don't have to be exactly this, but my answers are:

- Literally raise the price – if it's a supercomputer or some other system that sells CPU time, just, increase the price and that reduces demand
- Reduce the number of users on the system
- Limit the number of concurrent processes or threads each user is allowed