CS 111 Midterm Exam

Cody S Hubbard

TOTAL POINTS

84 / 100

QUESTION 1

1 Pages and page frames 8 / 10

- **- 0 pts** Correct
- **- 10 pts** No answer.
- ✓ **- 2 pts** **Did not explain that pages get placed into page frame.**
- **- 1 pts** Said page is mapped to page frame
- **- 1 pts** No mention of location of page & page frame in system
- **- 4 pts** Incorrect Reasoning
- **- 3 pts** Not accurate enough
- **- 1 pts** No mention of virtual address
- **- 5 pts** Page frame contains exactly one page.
- **- 2 pts** Did not mention that page and page frame are the same size.

QUESTION 2

2 ABIs 9 / 10

- **- 0 pts** Correct
- **- 10 pts** No answer
- ✓ **- 1 pts** **Did not mention operating system**
- **- 4 pts** Said applications don't need to worry about hardware differences
- **- 2 pts** Off-topic
- **- 3 pts** Incorrect reasoning
- **- 2 pts** Did not mention software distribution
- **- 3 pts** Did not mention hardware and OS
- **- 2 pts** Left out hardware

QUESTION 3

3 Information hiding 3 / 10

- **- 0 pts** Correct
- **- 10 pts** No answer.
- **- 6 pts** Primary benefit is to avoid bugs arising from improper dependencies among modules.
- **- 3 pts** Major element of benefit is that it allows changes in module implementation.
- ✓ **- 7 pts** **Really about hiding details of OS modules' implementation, not about concealing processes' address spaces from each other.**
- **- 2 pts** Nothing to do with open vs. closed source.
- **- 9 pts** Primarily an issue involving abstraction.

QUESTION 4

4 Context switches 4 / 10

- **- 0 pts** Correct
- **- 2 pts** Not mentioning general registers
- ✓ **- 2 pts** **Not mentioning PC**
- ✓ **- 2 pts** **Not mentioning Stack ptr**
- ✓ **- 1 pts** **Not mentioning PSW**
- **- 2 pts** No discussion of memory mapping data
- **- 2 pts** No need to explicitly save data, since it's already sitting in memory.
- **- 3 pts** Generally nothing goes to disk on a context switch.
- **- 1 pts** What about memory needs to be saved?
- **- 2 pts** Much of this stuff need not be saved, since it's already in memory. OS just needs to be sure it can be found again when process is switched back in.
- **- 2 pts** The PCB is an OS data structure that exists as long as the process is around, so it need not be saved on a context switch.
- **- 1 pts** File size has nothing to do with a context switch.
- **- 1 pts** I have no idea what the flag you're talking about is.
- ✓ **- 1 pts** **"state of the process" is vague.**
- **- 2 pts** Caches aren't saved.
- **- 2 pts** No need to update a file descriptor during a context switch.

QUESTION 5

5 Trap tables 10 / 10

✓ - 0 pts Correct

- 10 pts No answer
- 3 pts Answer incomplete, should mention trap table is used to specify what code to run when trap occurs.
- 8 pts Wrong answer.
- 3 pts Answer incomplete.
- 2 pts User process has no thing to do with trap?

QUESTION 6

6 Race conditions 10 / 10

✓ - 0 pts Correct

- 10 pts No answer
- 5 pts Answer incomplete.
- 8 pts Answer incorrect.
- 2 pts Missing some details.

QUESTION 7

7 Blocking and threads 10 / 10

✓ - 0 pts Correct

- 10 pts No answer or Wrong anwser
- 5 pts Missing: User-mode threads block other threads of the same process.
- 5 pts Missing: Kernel-mode threads do not block other threads of the same process, as other threads can be scheduled to run on the same or another core.

QUESTION 8

8 STCF 10 / 10

✓ - 0 pts Correct

- 10 pts No answer or Wrong answer
- 5 pts Missing: interrupt the running one OR switch to the newly-added shorter ones.
- 8 pts Missing mention of new processes that might have shorter time to completion.

QUESTION 9

9 Fork and exec 10 / 10

✓ - 0 pts Correct

- 10 pts No answer
- 4 pts Not mentioning code replacement.
- 3 pts Not mentioning stack replacement.
- 3 pts Not mentioning heap replacement.
- 9 pts The question was about what happens after the exec, not the fork.
- 8 pts What resources are replaced by the exec?
- 7 pts Stack and code are changed by exec.
- 5 pts Fork/exec work with processes, not threads.
- 2 pts Even any data written after fork gets replaced by exec.
- 2 pts The old stack is totally overwritten.
- 10 pts Totally wrong. Nothing to do with multithreading and multicore.
- 6 pts So, what resources are replaced?

QUESTION 10

10 Fragmentation for memory management schemes 10 / 10

✓ - 0 pts Correct

- 10 pts No answer
- 5 pts Not identifying internal fragmentation for pages.
- 5 pts Paged segments suffer 1/2 page fragmentation.
- 5 pts Fixed segments suffer 1/2 internal segment fragmentation.
- 3 pts On average 50%
- 2 pts The 1.5% was a particular example. It will be 1/2 page, on average.
- 2 pts Internal fragmentation has little to do with how long the system runs, unlike external.
- 2 pts Paging and fixed size partitions never experience external fragmentation.
- 2 pts The paging form of fragmentation you describe is internal fragmentation.
- 2 pts Paging doesn't use binary buddy. It allocates in fixed size pages.
- 4 pts Fixed segments are likely to waste more memory on internal fragmentation than paging, not less.
- 3 pts No external fragmentation with paging.

- **5 pts** No answer on segmented system.
- **2 pts** Calling internal fragmentation "external".
- **1 pts** This form of fragmentation is called "internal."
- **4 pts** Paged segment fragmentation only occurs in the last page.