CMSC216: Bonus Review 3A

Chris Kauffman

Last Updated:
Sun Dec 7 09:34:10 PM EST 2025

Bonus Review Rules

- 3 Questions will be shown with about 5min per question, 15min total, time limit enforced on Gradescope Quiz
- Individual student bonus dots will be calculated as

  ```
  BonusDots = floor(log2(TotalCorrectSectionAnswers))
                - YourIncorrectAnswers
  ```
- Cooperation is allowed and encouraged within your discussion section: the more correct answers in the section, the more bonus points for all
- Staff will try to facilitate discussion but will not comment on correct/incorrect answers during the quiz
- Scores will posted after all sections have taken the done the bonus review, likely the following day
- Student in the Discussion Section with the highest `TotalCorrectSectionAnswers` will get +2 BonusDots
- Bonus Review is Open Resource just like the exam: https://www.cs.umd.edu/~profk/216/exam-rules.pdf

Staging

- Open up the Gradescope Bonus Review Quiz for the day
- Once started, the quiz closes after 15min
- Get your resources set for the quiz

Okay...

Question 1

Which of the following is NOT information that is tracked by the operating system about a running process?

- (A) Open files and positions within them for read/write operations
- (B) Valid virtual memory address ranges
- (C) Physical locations corresponding to virtual memory address ranges
- (D) Direct child processes created by the process
- (E) Parent process that created this process
- (F) Heap memory blocks that are available and in-use

Question 2

Threads and processes have much overlap with one another in modern computing systems but have some distinct difference from one another. Select ALL of the TRUE STATEMENTS below:

- (A) Within a process tree, changes made to values in memory by one child process will be “seen” by all other child processes
- (B) Within a process, changes made to values in memory by one thread will be “seen” by other threads in the process
- (C) All parent / child processes share registers so must coordinate their use manually as the OS does not save/restore registers when switching between processes
- (D) Within a process, all threads share registers so must coordinate their use manually as the OS does not save/restore registers when switching between threads
- (E) The standard system calls we studied which create processes and threads all take a parameter for the function the new process/thread will execute when scheduled to run

Question 3

Nearby is part of an implementation of `job_update_status()` from P4's Shellac. Select the option that best describes this implementation.

```c
int job_update_status(job_t *job){
  int ret, status;
  int options = WNOHANG;
  ret = waitpid(job->pid, &status, options);
  if(!job->is_background){
    while(ret==0){
      ret = waitpid(job->pid, &status, options);
    }
  }

  if(ret == -1){
    perror("ERROR: -1 from watpid()");
    exit(1);
  }
  if(ret != job->pid){
    return 0;
  }

  // remaining code to handle normal and
  // abnormal exits is correct but omitted
  // ...
  return 1;
}
```

- (A) The code is functionally CORRECT but inefficient as it exhibits polling behavior
- (B) The code is functionally CORRECT but is not accounting for some system call failures so will fail test cases
- (C) The code is functionally INCORRECT because it creates concurrency problems between processes and will fail test cases
- (D) The code is functionally INCORRECT because the chosen parameters will cause system calls to fail and leading to test case failures