

CMSC 216 Quiz 5 Worksheet

The next quiz for the course will be on Wed, Nov 14. The following list provides additional information about the quiz:

- The quiz will be a written quiz (no computer).
- Closed book, closed notes quiz.
- Answers must be neat and legible.
- Quiz instructions can be found at <http://www.cs.umd.edu/~nelson/classes/utilities/examRules.html>
- Make sure you know your section number and your TA's name.

Solutions to the exercises below will not be provided, but you are welcome to discuss your solutions with the TA or instructor during office hours.

Exercises

1. What is the kernel?
2. What is context switching?
3. What is a system call?
4. What is `fork()`?
5. Which process has pid #1?
6. Which system call allows a process to reap another process?
7. What happens to a process ID when `exec()` is called successfully?
8. What happens to the process's address space when `exec()` is called successfully?
9. What happens to the (parent) process's address space when `fork()` is called successfully?
10. What happens to environment variables when `exec()` is called? When `fork()` is called? When the parent modifies environment variables after `fork()`?
11. What is a signal?
12. Provide two examples of signals.
13. What is the difference between standard I/O and Unix I/O?
14. In the example **`wait4ReapNonDeterministic.c`** (part of the class lecture examples) several processes are created in a loop. What will happen if the statement `"exit(idx)"` is removed?
15. Write a program that creates a child process that displays even values between 1 and a 100. The parent will wait for the child to finish (using **`wait`**) and print the message "Processing Done" once the child is done. The parent will display the child pid after the child has been created.
16. Write a program that creates two processes. The first process will execute the `"cal"` command and the second the `"date"` command. Use **`execvp`** in your implementation.