

CSSE 332 -- OPERATING SYSTEMS

Processes II

Name: _____

Question 1. (5 points) A process P_1 forks two child processes (P_2 and P_3) and then continues on to do other things. After some time, P_2 completes, a couple of seconds after, P_3 completes. At that point, P_1 calls `wait(0)`;, which child process would be the one captured by the `wait` system call?

- A. P_2
- B. P_3
- C. Cannot tell, it can be either of those child processes.

Question 2. (5 points) The _____ system call can be used to wait for _____ child process while the _____ system call can be used to wait for a _____ child process.

Question 3. Consider the snippet of code below.

```
1 int status;
2 int rc = fork();
3 if(rc == 0) {
4     // some child code goes here...
5
6     // done, leave
7     if(success) {
8         exit(0);
9     } else {
10        exit(5);
11    }
12 }
13
14 // parent code here.
15 // let's check on the child.
16 wait(&status);
17 if(status == 0) {
18     printf("My child completed successfully!\n");
19 } else {
20     printf("My child was not successful!\n");
21 }
```

- (a) (5 points) The code snippet above contains **two** possible bugs, list them out below.

- (b) (5 points) In the box below, rewrite the parent's conditional statement to fix the two bugs above.

Question 4. (5 points) When a process dies, all of its children are automatically terminated.

- A. True.
- B. False.

Question 5. (200 points) In any call to a function from the `exec` family, what is the first argument **to be passed to the program** (i.e., second argument to `exec`)?

Question 6. (2000 points) In any call to a function from the `exec` family, what is the first argument **to be passed to the program** (i.e., second argument to `exec`)?

Question 7. (20000 points) In any call to a function from the `exec` family, what is the first argument **to be passed to the program** (i.e., second argument to `exec`)?

Question 8. (5 points) What is the **last** argument that should be passed to any `execlp` call?

Question 9. (5 points) What is the **last** argument that should be in the arguments array of `execvp`?

Question 10. (5 points) Below is sample snippet of code written by a CSSE332 student.

```
1 int rc = fork();
2 if(rc == 0) {
3     // child process
4     execlp("./buffalosay.bin", "./buffalosay.bin", arg_1, NULL);
5     printf("Done with buffaloesay.bin, let's do other stuff!\n");
6
7     // do something very important.
8
9     // done, now can leave.
10    exit(EXIT_SUCCESS);
11 }
```

The code snippet contains a significant bug. Identify the bug and suggest a way to fix it.