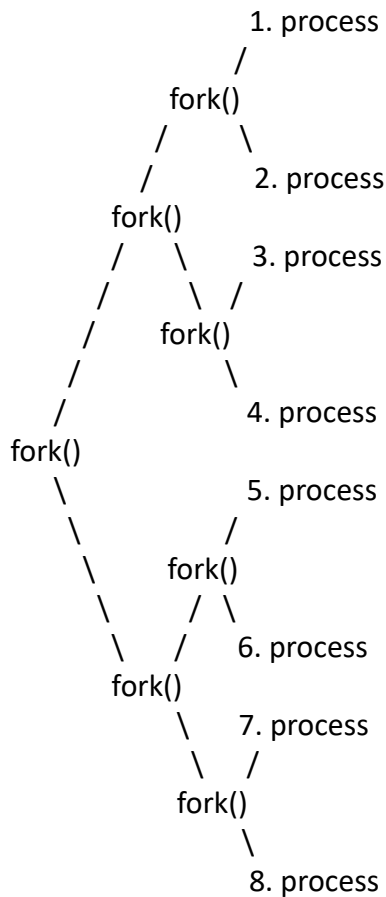


1. Source code and image of output submitted.
2. a. The code is not correct because the `val` variable that is defined in the if statement changes the pointer and then goes out of scope and returns the pointer as a different value.

b.

```
int f1(int val) {  
    int num = 0;  
    int *ptr = &num;  
  
    if (val == 0) {  
        int zero_val = 5;  
        ptr = &zero_val;  
    }  
    return (*ptr + 1);  
}
```

3. a. The fork function can take an existing process and create a new one. This is considered the child process. The function returns twice. The parent will return the process ID of the new child so it can keep track of all children and the child will return 0. A child can only have one parent. The child is identical to the parent, but has its own space in memory. It's important to understand that the order of when a child and parent process is run is determined by the kernel and the variation could cause errors in code. The parent and child also share a file table entry for every open descriptor. If both are run at the same time, you may receive the return from the child first or you could even get a mix of output from the parent and the child simultaneously.
- b. 8 new processes (see diagram on next page)



4. a. The system function allows you to write c code in the same syntax that you use on the command line. The function will work differently depending on what operating system you are on, so you must be careful. System calls fork, exec, and waitpid. The fork creates a new process and then exec runs the argument you specified for system. System automatically does all the error handling required with fork and exec. An error can occur if the process that called children already has children and terminates, because that will also terminate the child created by systems and that is why waitpid is used.

b. Advantages:

- The syntax can be easier to read/write
- Allows applications to work on multiple platforms without the code needing to be changed.

Disadvantages:

- Can cause security risks
- May not work between different systems

- System calls are slower

Own guidelines:

- If anyone else is working on the same server, it's important to consider the security risk, because they could potentially exploit the call by changing system level logic.
- Good option to consider if there are issues where in one platform a certain function won't work in a language, but then you have to consider if the system call will still work between systems.