

Lab 1: C System Programming Review

CS536

Duc Viet Le

September 12, 2016

Problem 1.

1. What happens if the return value of `execlp()` is not checked and you give an invalid command, say, `lssss`, as input to the shell?

Ans. if `execlp()` is not checked, if an invalid command is passed to the shell, the child process will not be terminated, and the code will create a new child process to handle next user input.

2. What happens if the parent process does not perform `waitpid()` and immediately returns to the beginning of the `while-loop`?

Ans. Even when a child exits, the Linux system will still keeps some information of child processes. The parent is required to clean up child processes (i.e using `waitpid()`).

If the parent process does not perform `waitpid()`, its child processes are still there and do nothing (i.e becoming zombies). They waste memory and may cause the kernel to run out of processes.

3. If the concurrent server were a file server that receives client requests from processes on the same host/OS (or over a network), why is performing `waitpid()` as a blocking call from within the parent process not a valid approach? Describe an asynchronous method for performing `waitpid()` so child processes are prevented from becoming zombies and their exit status can be checked.

Ans.

4. Ignoring the functional simplifications of the shell code, point out at least two programming bugs that should be fixed to yield more reliable server code.

Ans.

- Did not check if `k < 0`, when the code fails to create child process (i.e fail to `fork()`)
-