

OS 第二次作业

范翔宇 PB18000006

1. 答：循环四次，每次循环父进程和其子进程都会产生各自的子进程，呈指数增长，故有 $2^4=16$ 个进程。
2. 答：子进程创建成功且 `execlp()` 函数调用失败。
3. 答：
 - A: 子进程，`fork()` 函数返回 0，`pid=0`，输出 0；
 - B: 子进程，`getpid()` 函数返回当前进程 `pid`，`pid1=2603`，输出 2603；
 - C: 父进程，`fork()` 函数返回子进程 `pid`，`pid=2603`，输出 2603；
 - D: 父进程，`getpid()` 函数返回当前进程 `pid`，`pid1=2600`，输出 2600。
4. 答：子进程中的改变并不会影响父进程
故 LINE X: 0 -1 -4 -9 -16
LINE Y: 0 1 2 3 4
5. 答：若 `execl()` 函数调用成功，则会更改正在执行的代码，并且永远不会返回到原始代码，故 LINE X 不会执行；若 `execl()` 函数调用失败，则 LINE X 执行。
6. 答：
 - (1) 回收进程所占用的资源；
 - (2) 保持 PCB 所占资源最小化，使子进程成为僵尸进程便于父进程访问；
 - (3) 如通过 `exit()` 函数进入该状态，会向父进程发送 SIGCHLD 信号，使后续能被彻底清除。
7. 答：僵尸进程是当子进程比父进程先结束，而父进程又没有回收子进程，释放子进程占用的资源，此时子进程将成为一个僵尸进程。可以用 `signal` 函数为 SIGCHLD 安装 handler，因为子进程结束后，父进程会收到该信号，可以在 handler 中调用 `wait` 回收。如果他的父进程没安装 SIGCHLD 信号处理函数调用 `wait` 或 `waitpid()` 等待子进程结束，又没有显式忽略该信号，那么它就一直保持僵尸状态，如果这时父进程结束了，那么 `init` 进程会自动接手这个子进程，为它收尸，它还是能被清除的。