

1. Including the initial parent process, how many processes are created by the program shown in Figure 1?

加上父进程，总共会创建16个进程。

2. Explain the circumstances under which the line of code marked printf (“LINE J”) in Figure 2 will be reached.

执行execlp后将不会返回到原来的代码，因此如果要printf (“LINE J”)被执行，除非execlp执行失败。

3. Using the program in Figure 3, identify the values of pid at lines A, B, C, and D. (Assume that the actual pids of the parent and child are 2600 and 2603, respectively.)

A:0 B:2603 C:2603 D:2600

4. Using the program shown in Figure 4, explain what the output will be at

X: CHILD: 0 CHILD: -1 CHILD: -4 CHILD: -9 CHILD: -16

Y: PARENT:0 PARENT:1 PARENT:2 PARENT:3 PARENT:4

5. For the program in Figure 5, will LINE X be executed, and explain why.

如果execl执行成功，则X不会被执行，因为execl打开的程序将覆盖原程序的代码段。

6. Explain why “terminated state” is necessary for processes.

如果没有“终止状态”，进程从创建出来后，将永远循环在进程调度队列中，而进程数量是有限的，如果没有终止状态，不释放资源，那么计算机将很快死机。同时，进程结束和释放资源之间还需要时间，进程在终止状态可能还有消息需要传递，并等待其父进程决定对其后续的操作。

7. Explain what a zombie process is and when a zombie process will be eliminated (i.e., its PCB entry is removed from kernel).

僵尸进程是已经处于终止状态但其父进程尚未调用wait()的进程。当一个进程终止时，操作系统将释放其所占用的资源，但是，他在进程表中的条目仍然存在，直到它的父进程调用wait()。当僵尸进程的父进程调用了wait后，僵尸进程的SIGCHLD信号将被接收，wait将释放该僵尸进程的PCB，彻底清除该进程

8.Explain what data will be stored in user-space and kernel-space memory for a process.

用户空间的数据包括代码段（存放程序代码），数据段（存放全局变量等），堆栈空间（存放局部变量等）。内核空间的数据则包含进程状态，程序计数器，CPU寄存器，CPU调度信息，内存管理信息，记账信息，I/O状态信息（打开文件列表等）。

9.Explain the key differences between exec() system call and normal function call.

普通函数调用不会更改原程序的代码段，执行完成后将返回调用它后的位置。而exec（）则会对原来的代码段进行更改，包括局部变量、全局变量、堆栈空间、程序计数器等等，执行完成后不会返回调用位置。