

1. 16 个进程
2. 父进程通过 fork 调用创建子进程，如果该进程为子进程，执行 exec ()，
如果执行成功，此时子进程被 exec 中参数指定的程序覆盖，执行新的程序而不返回，printf("Line J")无法执行。
如果执行失败，printf("Line J")可以执行。
3. A:0 B:2603 C:2603 D:2600
4. X:0,-1,-4,-9,-16 Y:0,1,2,3,4
5. 同 2.，如果 exec 执行成功，则进程被覆盖，LINE X 不会被执行。若 exec 执行失败，则 LINE X 会被执行。
6. 终止状态的存在是实现一种子进程向父进程反馈信息的一种机制，子进程执行完代码退出后，父进程应该读取子进程的退出状态。进程的终止状态可以显示进程如何结束，它不仅可以通过发送带有终止状态信息的信号来唤醒父进程，还可以通知父进程子进程的终止信息。如果子进程异常终止，则父进程可以通过反馈信息做出应对。
7. 僵尸进程是指一个进程在调用 exit 命令结束时，清理所有的用户空间和大部分已经分配的内核空间，仅仅在进程列表中保留一个位置，记载该进程的退出状态等信息供其他进程收集。
僵尸进程会以终止状态保持在进程表中，并且会一直等待父进程读取其退出状态。即使子进程已经退出，父进程还在运行，但没有读取子进程状态，子进程变为僵尸进程。如果父进程安装 SIGCHLD 信号处理函数调用 wait 或 waitpid 函数等待子进程结束或显示忽略信号，那么子进程就会结束僵尸进程状态，其 PCB 会被回收。如果父进程提前终止，则子进程的父进程变为 init 进程，init 进程将周期性调用 wait 消除僵尸进程。
8. User space: Global variables, Local variables, Dynamically-allocated memory, code + constants
Kernel space: PCB, kernel code with system calls
- 9.

函数库调用	系统调用
在所有的 ANSI C 编译器版本中，C 库函数是相同的	各个操作系统的系统调用是不同的
它调用函数库中的一段程序（或函数）	它调用系统内核的服务
与用户程序相联系	是操作系统的一个入口点
在用户地址空间执行	在内核地址空间执行
属于过程调用，调用开销较小	需要在用户空间和内核上下文环境间切换，开销较大

10. bc
11. a.6 个进程 b.两个线程
12. LINE C : CHILD: value = 5
LINE P : PARENT: value = 0
13. 匿名管道只用于相关进程，命名管道可以是非父子进程间通信
匿名管道是单向的，命名管道是双向的
匿名管道随通信结束而结束，命名管道会一直存在直至被显示释放
匿名管道一般只为两个进程间使用，命名管道可在多个进程间使用。