

系统编程LAB12

17379009 张曼薇

Q1

一个进程是正在运行的一个程序，有自己的独占的地址空间和系统资源。进程的管理使CPU能在不同的进程之间切换，创造“同时运行多个程序”的假象，并且能管理相互之间有逻辑关系的进程。

代码段A会执行共5次 `fork()` 命令，每次进程数*2，最后共有32个进程。代码段B每次循环中创建一个子进程，创建的子进程pid为0，因此立刻退出循环，因此一共在原进程上创建5次子进程。

Q2

僵尸进程是进程退出后，父进程回收前该进程留下的一种数据结构，不占用空间和资源，只记录进程的退出状态等有用的信息，以供父进程或者init进程回收并检查退出状态。

`wait()` 函数的原型是 `pid_t wait(int *status)`，父进程调用 `wait()` 函数时自身会被挂起，并等待自己的子进程中某个进程运行结束变成僵尸进程，当某个子进程僵死，`wait()` 就会返回这个子进程的 `pid`，并使父进程恢复运行。

Q3

信号在进程之间传递信息，用来通知进程某个异步事件的发生并根据相应的执行函数进行处理，在软件层次上是对中断机制的模拟。

Q4

信号被发出但还未被进程接收时，为未决状态。若信号的发送对一个进程来说过于密集，则可能会在处理一个信号的同时收到下一个信号，后一个信号在没有信号队列的情况下会被丢弃。在有信号队列的情况下，这个后来的信号会被内核设定为阻塞+未决状态，等待进程处理完先到的信号。

Q5

可以取一个二进制信号量，只能为0或者1，初始时为1。

这个信号量有两个操作：等待和发送，某个进程想要访问共享内存时，必须先执行等待操作。等待操作中若信号量的值>0，则减一，并授权该进程访问内存；若信号量的值=0，则说明有其他进程在访问内存，因此挂起该进程。

当使用该内存的进程离开，须执行发送操作。发送操作中，若有挂起的进程在等待，则授权进程访问内存，若没有进程在等待，就将信号量+1。