

1

进程在Linux系统中的用途

进程是对运行中的程序的一种抽象。操作系统以进程为单位进行程序执行的描述与资源的分配和调度。

理论上单核CPU硬件上并不能同时运行两段程序，但是借助进程的概念和CPU轮询，可以让计算机达到同属处理多个任务的目的。

代码段A和代码段B的区别

这两段代码最大的区别在于是否在创建进程时判断本进程是不是父进程。

代码A没有进行判断，所以无论是子进程还是父进程，运行到此处时都会创建一个新的进程，每执行一次for循环，总的进程数就会变成原来的2倍。因此，最后一共有 $2^5 = 32$ 个进程。

代码B中，只有原始的父进程会创建进程，子进程会跳出循环，因此创建了5个子进程，共6个进程。

2

什么是僵尸进程

进程调用exec()后，会留下一套基本空的数据结构，只是在进程表中占一个位置，并且只能由父进程进行回收。如果父进程不对其进行回收，这个进程就成了僵尸进程。

调用wait()捕获僵尸态进程

```
1 | pid_t wait(int *status);
```

wait函数会等待任何一个子进程变为僵尸态。如果有一个子进程变为僵尸态，函数会返回子进程的pid，status中会存储进程的结束信息。如果没有子进程变为僵尸态，该进程会挂起。

如果不需要结束信息可以传入NULL

可以利用宏对返回信息进行解析

3 信号在Linux系统中的作用

信号是Linux中的一种通讯机制，信号是发送给进程的特殊的异步消息。进程在接收到信号时立即处理，并不需要执行完当前函数甚至是代码行。

Linux中有多种信号，可以实现进程中断、进程中止、系统定时、异常处理等

信号什么时候处于未决状态

信号未决是指信号已经被产生但是还没有被传递

信号未决状态的作用

未决状态信号的产生主要是因为进程对此信号的阻塞。例如为进程产生一个选择为阻塞的信号，而且对该信号的动作是系统默认动作或捕捉该信号，则为该进程将此信号保持为未决状态，直到该进程对此信号解除了阻塞或者对此信号的动作改为忽略。

未决状态实现了信号的阻塞，如果没有未决状态，信号没有被及时处理可能会造成信号的丢失。

5 设计一种通过信号量来实现共享内存读写操作同步的方式

可以利用一组含有一个信号量的信号量集。

进程A创建进程B，父子进程中同时获取信号量，先执行获取代码的进程实际上是进行创建信号量，创建后对共享内存初始化；后创建的因为创建失败，执行代码获取这组信号量。

信号量的初始值为1。

读和写之前都要对信号量进行P操作，读或者写之后对信号量进行V操作。