

1.简述“进程”这个概念在Linux系统中有什么用途。

划分了最小的程序执行和资源共享的单位

2.阅读代码段

```
//代码段A
int i;
for(i=0;i<5;i++)
{
    pid = fork();
}
//代码段B
int i;
for(i=0;i<5;i++)
{
    if((pid=fork())==0)
        break;
}
```

A执行结果：出现 $2^5=32$ 个除了进程id外完全相同的进程。

B执行结果：所有的子进程不会继续fork，因此只有10个除了pid外完全相同的进程。

3.僵尸进程与wait()捕获僵尸进程的过程

僵尸进程是运行完毕或异常终止后未被清理的进程，通常是未被父进程清理。这样的进程会被init进程收养。

使用wait捕获僵尸进程：父进程调用wait(int *status)函数，是为了等待子进程终止，子进程终止后，wait()函数返回子进程的退出状态(status即为这个退出状态)，由此就获得了子进程（函数退出时已经是僵尸进程）的状态。

当父进程使用这个函数时，若没有它的子进程僵死，则父进程会阻塞，直到wait()分析到出现了一个它的僵尸子进程，wait()函数才会返回，父进程才能继续进行。由此，wait()函数就捕获了调用者的僵尸子进程。

4.信号的作用

它是进程通讯机制，是发送给进程的特殊的异步消息。可以被系统立即处理，不必等待当前函数或代码执行完毕。并不由进程发出。本质是软件层上对中断机制的模拟。

5.信号的未决状态和作用

发送的信号被阻塞，无法到达目标进程，这个状态是未决状态。

信号存在未决状态，是为了解决后发送的信号被正在处理当前信号的内核忽略最后丢失的问题。若某一信号在进程的掩码集中被设位1，意味着它被屏蔽，即进入阻塞状态，同时内核修改未决集中该信号的位为1，在该信号解除屏蔽之前，内核不会再发送这个信号给进程。

6.请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

- 1.创建/获取信号量。
- 2.使用上面创建/获取的信号量键值，在读/写的进程中都由它来创建/获取共享空间。
- 3.分别在读/写进程中绑定这个共享空间。
- 4.在写进程中，先调用P操作，以防未完成写操作时读进程开始读、其他进程进入共享空间。在写操作完成后再调用V操作释放。
- 5.在读/写进程的最后，释放共享空间和信号量。