

系统编程第12周作业

1.每个概念被创造都有其意义，请简述“进程”这个概念在Linux系统中有什么用途

答：对应用程序来说，进程就像一个大容器：在应用程序被运行后，就相当于将应用程序装进容器里了，你可以往容器里加其他东西(如:应用程序在运行时所需的变量数据、需要引用的DLL文件等)，当应用程序被运行两次时，容器里的东西并不会被倒掉，系统会找一个新的进程容器来容纳它。一个进程可以包含若干线程(Thread)，线程可以帮助应用程序同时做几件事(比如一个线程向磁盘写入文件，另一个则接收用户的按键操作并及时做出反应，互相不干扰)。

2.阅读以上代码段，回答代码段A和代码段B的执行结果有什么区别？并解释为什么会有这样的区别。

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

代码段A执行后会产生 2^5 个进程，而代码段B执行后，只有6个进程。

原因：每次调用fork()函数，系统都会复制源程序。那么代码段A在第一次循环后，现在就有了两个进程，此时两个进程都会继续运行for循环，在循环一次后，进程数在翻倍，最后的总数就是 2^5 个进程。但是代码段B执行循环后，子进程不会继续for循环，会直接break，所以最后只有5个子进程一个父进程。

3.用自己的话阐述什么是僵尸进程，并描述进程通过调用 wait() 捕获僵尸态的子进程的过程。

僵尸进程：当进程调用了exit()函数后，该进程不会马上消失，而是成为僵尸进程。僵尸进程几乎放弃进程推出前占用的所有内存，既没有可执行代码也不能被调度，只是在进程列表中保留一个位置，记载进程的退出状态等信息供父进程收集。

调用wait()函数的进程会被挂起，进入阻塞状态，直到子进程变为僵尸态，wait()函数捕获到该子进程的退出信息时才会转为运行态，回收子进程的资源并返回；若没有变为僵尸态的子进程，wait()函数会让进程一直阻塞。若当前进程有多个子进程，只要捕获到一个变为僵尸态的子进程信息，wait()函数就会返回并使进程恢复执行。

4.请简述信号在Linux系统中的作用。

信号在Linux系统中本质上是软件层次上对中断机制的一种模拟，用于提醒进程某件事情即将发生。对于软件编程和进程管理等都具有重要的作用。

5.请简述信号什么时候处于未决状态，并简述信号存在未决状态的作用

未决状态：发送的信号被阻塞，无法到达进程，内核就会将信号的状态设置为未决状态。

对进程来说，若信号的发送过于密集，即在处理信号的同时再次收到信号，那么进程会将后到的信号丢弃。对于信号的发送方来说，应该发送的信号已经发送，自然不会再次发送；但对于作为信号接收方的进程来说，未对信号做出应有处理，这显然是不符合预期的。信号存在未决状态的作用就是用于处理这一情况。

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，文字阐述即可，不需要代码实现。（提示：在写进程操作未完成时，需要防止其他进程从共享内存中读取数据）

两个进程,一个写，一个读，写进程不断向创建的共享内存写数据，读进程通过getchar()共享内存的最新数据，读写共享内存时通过信号量同步。其中一个文件read.c创建共享内存，并读取其中的信息，另一个文件write.c向共享内存中写入数据。为了方便操作和数据结构的统一，为这两个文件定义了相同的数据结构，定义在文件data.c中。结构shared_use_st中的written作为一个可读或可写的标志，非0：表示可读，0：表示可写，text则是内存中的文件。

程序read创建共享内存，然后将它连接到自己的地址空间。在共享内存的开始处使用了一个结构struct_use_st。该结构中有一个标志written，当共享内存中有其他进程向它写入数据时，共享内存中的written被设置为0，程序等待。当它不为0时，表示没有进程对共享内存写入数据，程序就从共享内存中读取数据并输出，然后重置设置共享内存中的written为0，即让其可被write进程写入数据。程序write取得共享内存并连接到自己的地址空间中。检查共享内存中的written，是否为0，若不是，表示共享内存中的数据还没有被完，则等待其他进程读取完成，并提示用户等待。若共享内存的written为0，表示没有其他进程对共享内存进行读取，则提示用户输入文本，并再次设置共享内存中的written为1，表示写完成，其他进程可对共享内存进行读操作。