系统编程 12 周

18373389 金子晖

1.简述“进程”这个概念在 Linux 系统中有什么用途

进程是计算机中的程序关于某数据集合上的一次运行活动，是系统进行资源分配和调度的基本单位，是操作系统结构的基础。在早期面向进程设计的计算机结构中，进程是程序的基本执行实体；在当代面向线程设计的计算机结构中，进程是线程的容器。程序是指令、数据及其组织形式的描述，进程是程序的实体。

进程是操作系统中最基本、重要的概念。是多道程序系统出现后，为了刻画系统内部出现的动态情况，描述系统内部各道程序的活动规律引进的一个概念,所有多道程序设计操作系统都建立在进程的基础上。

2.代码段 A 和代码段 B 的执行结果有什么区别 解释为什么

运行代码段 A 产生 32 个进程，运行代码 B 产生 6 个进程。

当一个进程调用 fork()函数后，系统先给新的进程分配资源，例如存储数据和代码的空间。然后把原来的进程的所有值都复制到新的新进程中，只有少数值与原来的进程的值不同，相当于克隆了一个自己。

代码段 A 最开始的父进程和每个新产生的子进程都在循环里，每次循环每个进程都会再分出来 1 个进程，所以最后有 32 个进程，代码段 B 每次 fork()的子进程都会跳出循环，只有父进程留在循环里，所以最后有 6 个进程。

3.什么是僵尸进程 进程通过调用 wait()捕获僵尸态的子进程的过程

一个进程结束了,但是他的父进程没有等待(调用 wait / waitpid)他, 那么他将变成一个僵尸进程。但是如果该进程的父进程已经先结束了,那么该进程就不会变成僵尸进程,因为每个进程结束的时候,系统都会扫描当前系统中所运行的所有进程,看有没有哪个进程是刚刚结束的这个进程的子进程,如果是的话,就由 Init 来接管他,成为他的父进程。

一个进程在调用 exit 命令结束自己的生命的时候，其实它并没有真正的被销毁，而是留下一个称为僵尸进程（Zombie）的数据结构（系统调用 exit，它的作用是使进程退出，但也仅仅限于将一个正常的进程变成一个僵尸进程，并不能将其完全销毁）

进程一旦调用了 wait，就立即阻塞自己，由 wait 自动分析是否当前进程的某个子进程已经退出，如果让它找到了这样一个已经变成僵尸的子进程，wait 就会收集这个子进程的信息，并把它彻底销毁后返回；如果没有找到这样一个子进程，wait 就会一直阻塞在这里，直到有一个出现为止,当然 如果在调用 wait（）时子进程已经结束，则 wait（）就会立即返回子进程结束状态值。参数 status 用来保存被收集进程退出时的一些状态，它是一个指向 int 类型的指针。但如果我们对这个子进程是如何死掉的毫不在意，只想把这个僵尸进程消灭掉，我们就可以设定这个参数为 NULL，pid = wait(NULL); 如果成功，wait 会返回被收集的子进程的进程 ID，如果调用进程没有子进程，调用就会失败，此时 wait 返回-1，同时 errno 被置为 ECHILD。

4.信号在 Linux 系统中的作用

软中断信号（signal，又简称为信号）用来通知进程发生了异步事件。在软件层次上是对中断机制的一种模拟，在原理上，一个进程收到一个信号与处理器收到一个中断请求可以说是一样的。信号是进程间通信机制中唯一的异步通信机制，一个进程不必通过任何操作来等待信号的到达，事实上，进程也不知道信号到底什么时候到达。进程之间可以互相通过系

统调用 kill 发送软中断信号。内核也可以因为内部事件而给进程发送信号，通知进程发生了某个事件。信号机制除了基本通知功能外，还可以传递附加信息。

5.信号什么时候处于未决状态 信号存在未决状态的作用

信号产生和传递之间的时间间隔内，称此信号是未决的；简单的说就是：一个已经产生的信号，但是还没有传递给任何进程，此时该信号的状态就称为未决状态。

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。

信号屏蔽机制专门用于解决常规信号不可靠这一问题。在进程的 PCB 中存在两个信号集，一个称为信号掩码，另一个称为未决信号集。这两个信号集的实质都是位图，其中的每一位对应一个信号；若 mask 中某个信号对应的位被设置为 1，信号会被屏蔽，进入阻塞状态，此时内核会修改 pending 中该信号对应的位为 1，使该信号处于未决态，之后除非该信号被解除屏蔽，否则内核不会再向进程发送这个信号。

6.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

两个进程对同一个共享内存读写，一个进程写，另一个进程读，利用两个有名信号量 semr，semw 进行同步。semr 信号量控制能否读，初始化为 0。semw 信号量控制能否写，初始化为 1。