

## 1.

请简述“进程”这个概念在Linux系统中有什么用途。

进程是一个二进制程序的执行过程。在Linux操作系统中，想命令行输入一条命令，按下回车键，便会有一个进程被启动，每个命令都会对应一个进程，若使用管道符连接两个或多个命令，系统就会创建多个进程。

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。

进程是一个具有一定独立功能的程序关于某个数据集合的一次运行活动。它是操作系统动态执行的基本单元，在传统的操作系统中，进程既是基本的分配单元，也是基本的执行单元。

每个进程在内核中有一个进程控制块（PCB）来维护进程相关的信息，Linux内核的进程控制块是task\_struct结构体。

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。

代码A产生了32个进程，运行代码B产生了6个进程。

fork()函数执行后，系统会创建一个与原进程近乎相同的进程，之后父子进程都继续往下执行。代码A中，循环了5次，每次都会调用fork()函数都会复制一个子进程，创建的子进程在循环里，然后继续调用fork()进行复制，最终创造出来32个进程。代码B中每次创建出的子进程都被break掉，跳出循环，只有父进程还在循环中，也就是子进程不会继续创造子进程，最终只有6个进程。

## 3.

如果一个进程已经终止，但是他的父进程尚未调用wait或waitpid对它进行清理，这时进程状态称为僵尸进程。任何进程在刚终止时都是僵尸进程，正常情况下，僵尸进程都立刻被父进程清理了。

调用 wait() 捕获僵尸态的子进程的过程。

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的信息，wait()函数就会返回并使进程恢复运行。

## 4.

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.

---

未决状态：发送的信号被阻塞，无法到达进程，内核就会将该信号的状态设置为未决

执行信号的处理动作称为信号递达，信号从产生到递达之间的状态称为信号未决。进程可以选择阻塞某个信号。

被阻塞的信号产生时将保持在未决状态，直到进程解除对此信号的阻塞，才执行递达的动作。阻塞和忽略是不同的，只要信号被阻塞就不会递达，而忽略是在递达之后可选的一种处理动作信号不会丢失，如果信号被阻塞，只会保持信号未决，但是信号不丢失，只要取消阻塞 信号依然会到来。

## 6.

---

有名信号量既可用于线程间的同步，又可用于进程间的同步。

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