

作业012

1.

进程是正在执行的一个程序或命令，每个进程都是一个运行的实体，都有自己的地址空间，并占用一定的系统资源。程序是人使用计算机语言编写的可以实现特定目标或解决特定问题的代码集合。当程序被执行时，执行人的权限和属性，以及程序的代码都会被加载入内存，操作系统给这个进程分配一个 ID，称为 PID（进程 ID）。也就是说，在操作系统中，所有可以执行的程序与命令都会产生进程。只是有些程序和命令非常简单，如 ls 命令、touch 命令等，它们在执行完后就会结束，相应的进程也就会终结，所以我们很难捕捉到这些进程。某些进程会产生一些新的进程，我们把这些进程称作子进程，而把这个进程本身称作父进程。子进程是依赖父进程而产生的，如果父进程不存在，那么子进程也不存在了。进程管理主要的作用是查看服务器是否健康、查看进程、以及杀死进程。

2.

代码段A是父进程fork出子进程，每个子进程执行循环继续fork；而代码段B是父进程fork出子进程，之后子进程退出循环，不再fork。

fork()函数执行一次，却有两个返回值。代码段B中增加了通过fork()返回值来判断当前进程是不是子进程，如果fork()的返回值是0，则是子进程。如果是子进程则退出循环，不再继续fork。

3.

如果一个进程已经终止，但是它的父进程尚未调用wait或waitpid对它进行清理，这时的进程状态称为僵尸进程。

进程通过调用wait()捕获僵尸态的子进程的过程：

如果子进程已经终止，并且是一个僵尸进程，则wait立即返回并取得该子进程的状态；否则wait使其调用者阻塞，直到一个子进程终止。如调用者阻塞并且有多个子进程，则在其某一子进程终止时，wait就立即返回。

4.

信号是Unix、类Unix以及其他POSIX兼容的操作系统中进程间通讯的一种有限制的方式。它是一种异步的通知机制，用来提醒进程一个事件已经发生。当一个信号发送给一个进程，操作系统中断了进程正常的控制流程，此时，任何非原子操作都将被中断。如果进程定义了信号的处理函数，那么它将被执行，否则就执行默认的处理函数。

5.

在信号产生到递送之间的时间间隔内，称信号是未决的。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，或者将对此信号的动作更改为忽略，才执行递送的动作。

信号存在未决状态的作用：

内核在递送一个原来被阻塞的信号给进程时（而不是在产生该信号时），才决定对它的处理方式。于是进程在信号递送给它之前仍可改变对该信号的动作。

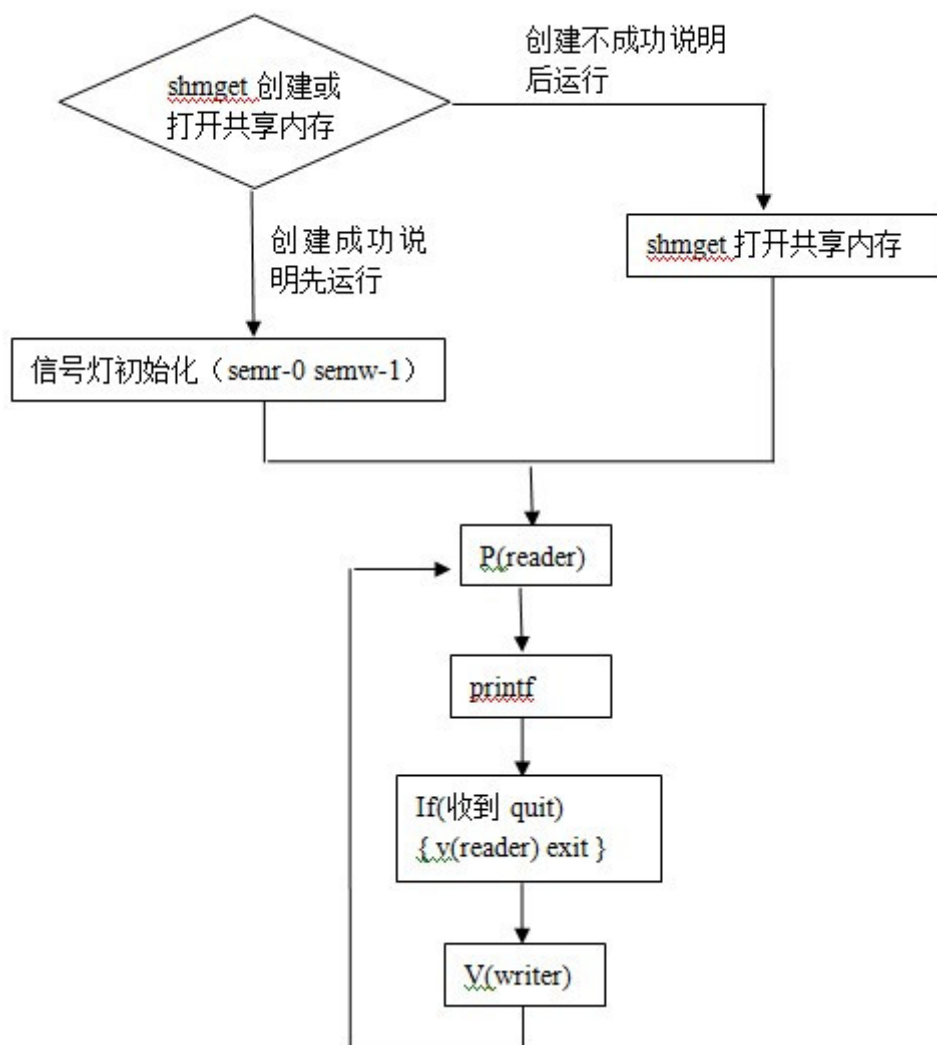
6.

利用POSIX无名信号灯实现共享内存的同步。定义以下结构体：

```
typedef struct {  
    sem_t semr;  
    sem_t semw;  
    char buf[MAXN];  
} SHM;
```

读写流程如下图所示。

reader 流程图



writer 流程图

