

课后习题

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。

进程是描述程序执行过程和资源共享的基本单位

主要目的：控制和协调程序的执行

进程组由一个或多个相关联的进程组成，目的是为了进行作业控制

每个进程都属于某个进程组

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。

A 会创造 31 个进程

B 会创造 5 个进程

在 B 中，fork 函数返回 0（在子进程中）则 break，一次只会创建一个进程

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。

子进程已结束，但父进程未调用 wait() 函数等待，就意味着：

子进程已终止，但没有被正确清除，成为僵尸进程

即使子进程在父进程调用 wait() 函数前已死亡（成为僵尸），其退出状态也可以被抽取出来，然后被清除

未清除的子进程自动被 init 进程收养

wait()

pid_t wait(int* status); pid_t waitpid(pid_t pid, int* status, int options);

阻塞主调进程，直到一个子进程结束

子进程的异步清除：

SIGCHLD 信号：子进程终止时向父进程自动发送，编写此信号处理例程，异步清除子进程

4. 请简述信号在 Linux 系统中的作用。

信号是发送给进程的特殊异步消息

进程接收到信号会立即处理

Linux 系统中有多种型号，各具有不同意义；系统以数字标识不同的信号，程序一般以名称引用之

系统信号

缺省处理流程：终止进程，生成内核转储文件

有 31 个

进程间发送的信号

终止进程信号 + 用户自定义信号

信号处理

进程接收到信号后，根据信号配置进行处理

缺省配置：在程序没有处理时确定信号该如何处理

程序处理信号的方式：按照信号处理例程所提供的函数指针调用指定函数

int sigaction(int signum, const struct sigaction* act, struct sigaction* oldact);

signum 为信号编号，act 和 oldact 分别为指向信号结构体 struct sigaction 的指针，前者为新配置，后者为需要保存的老配置

信号结构体 struct sigaction

最重要的成员为 `sa_handler`，其取值为 `SIG_DFL`（缺省配置）、`SIG_IGN`（忽略该信号）或指向信号处理函数的指针

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。

信号在产生和递达之间处于未决状态（Pending）

信号存在未决状态的作用：

1 需要在信号程序和其它程序共享全局变量时，如果全局变量的类型不是 `sig_atomic_t` 类型，当一部分程序恰好读、写到变量的一半发生信号，而信号程序里会改变该信号，那么就会产生混乱。为了避免这种混乱，提供程序的可靠性，在操作这类变量前阻塞信号，操作完成后恢复信号的传递

2 用来处理必须保证连续操作的完整性方面

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）

使用两个信号量 `semr`，`semw` 标记可读/写状态