《Linux应用程序开发实验报告》

题目: 进程间通信

班级: 17计算机4班

学号: 10417426

姓名: 沈雨晨

东南大学成贤学院计算机系

2020年4月23日

**一、实验内容**

1、有关管道的问题

（1）有关pipe()函数的问题：如果将waitpid(pc,NULL,0);注释掉，程序执行会发生什么结果，为什么？

注释掉父进程早于子进程结束，子进程退出成为孤儿进程另起shell行进行操作。因为waitpid(pc,NULL,0);就是用来阻塞父进程收集子进程退出信息后才做退出操作的

（2）改写标准流管道程序standard\_pipe.c，将popen函数中的r改成w，再做相应调整使得该程序可以正确运行？



改过正确运行了。cmd敲mkdir adc命令出现创建文件的正常效果。

（3）管道、标准流管道、有名管道各自的特征是什么？它们之间的区别何在？

管道( pipe )：管道是一种半双工的通信方式，数据只能单向流动，而且只能在具有亲缘关系的进程间使用。进程的亲缘关系通常是指父子进程关系。

标准流管道：通过popen（）函数实现

过程：创建管道

fork创建子进程

父子进程中关闭不需要的文件描述符

执行exec函数族调用

执行函数中所指定的命令

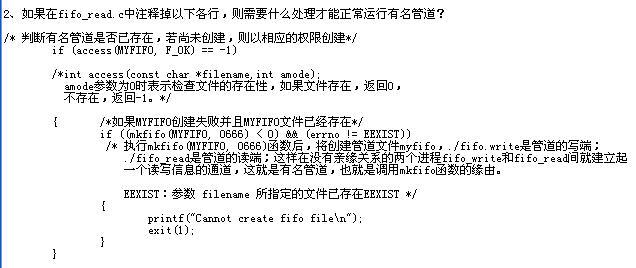
（整合操作）

效果和局限：popen()大大减少代码的编写量，只能使用标准I/O函数

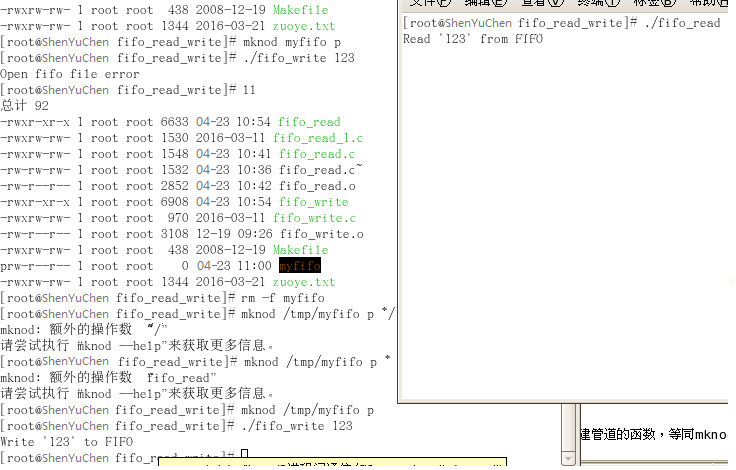
有名管道 (named pipe) ： 有名管道也是半双工的通信方式，但是它允许无亲缘关系进程间的通信。

有名管道是先进先出的，可以实现互不相关的两个进程实现彼此通信，本质是文件的读写操作。

扩展有名管道本质理解：有名代表有个文件叫myfifo，它是特殊的管道文件。其中fifo\_write进程可以往里面写字符，而另外一个进程fifo\_read进程可以从myfifo读字符；通过这个文件myfifo实现了两个不同进程对同一个文件的读写操作，以该文件为接口，间接实现了两个进程间的通信；这就是有名管道的本质。最简化说法：不同进程对同一个文件(该文件是管道文件)的读写操作就是有名管道。



答：在shell命令行下敲入对等命令：mknod /tmp/myfifo p 然后运行就会显示正确



3、有关信号的问题：

（1）如果在int ret=alarm(5);上面再加一条语句休眠10秒，会出现什么结果？

时间变久了

如果注释掉 pause();会打印底下的语句I have been waken up吗？为什么?

会，因为pause（）函数作用就是会把程序中断，提示定时时间到不会执行下面语句，去掉就有打印输出的内容

（2）信号是什么？信号集函数族处理信号的步骤是什么？

信号是指UNIX中所使用的进程通信的一种最古老的方法，是软件层次上对中断机制的一种模拟，是一种异步通信方式。信号是一种进程间通信机制，它给应用程序提供一种异步的软件中断，使应用程序有机会接受其他程序活终端发送的命令(即信号)。应用程序收到信号后，有三种处理方式:忽略，默认，或捕捉。进程收到一个信号后，会检查对该信号的处理机制。

步骤：

a.定义信号集合

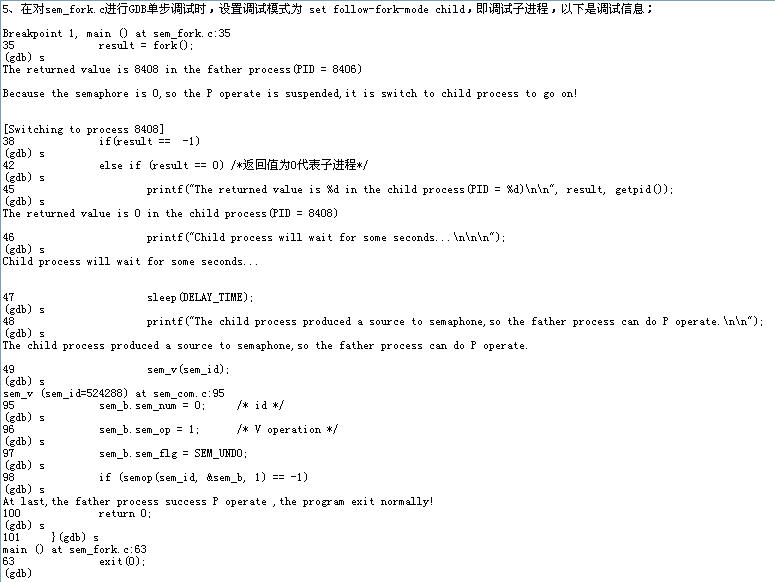
b.设置信号屏蔽位

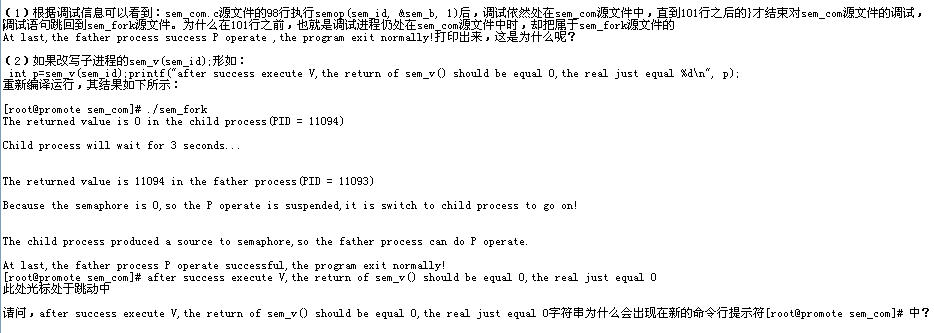
c.定义信号处理函数

d.测试信号



因为调试的时候是要父进程顺序执行，waitpid等不到子进程的信号，从而进入if语句内子进程强行被父进程关掉（杀死），然后调试的时候就会出错





1. 因为调试的子进程，做了V操作，有了信号量为1，父进程不再挂起，抢占资源执行P操作
2. 因为子进程执行V操作后，父进程抢占到了进程率先执行，结束语句在父进程内所以会退出，而子进程尚未执行完，变成了孤儿进程把打印语句执行完了。

（3）如果调试方式换成系统默认调试，即调试父进程而子进程不受影响继续执行。

根据逻辑以及（2）sem\_v(sem\_id)的打印语句：printf("after success execute..."); 子进程首先执行完成，而信号量的值也由0变成1；在接下来调试父进程的过程中，调试环节会跳到sem\_com源程序的sem\_p函数部分，在给sem\_b结构体赋初值后，再键入s单步执行下面的语句if (semop(sem\_id, &sem\_b, 1) == -1)，此时GDB调试就处于暂停状态，再没有任何响应了。

根据逻辑，子进程先执行过后，信号量必定变成1，目前有资源可消耗，则父进程的P操作一定可以执行，但是在本次GDB调试中，父进程在执行P操作的semop()函数时，实验现象却显示被阻塞，请问父进程为什么不能执行P操作，也就是不能执行semop()函数？？？

因为子进程执行过后未释放信号量退出，而在p操作函数中，sembuf结构体中的sem\_flg设置为SEM\_UNDO,会让系统自动释放子进程中未释放的信号量。信号量变为初值0，此时父进程再进行操作会被阻塞，不能执行P操作们，也就是不能执行semop（）函数。

6、共享内存是最为高效的进程间通信方式，请写出课后共享内存实验是如何使用信号量实现进程间通信的效果？搞懂生产者消费者程序执行的流程，并用自己的话描述两个源程序执行的关键步骤。

共享内存就是映射一段能被其他进程所访问的内存

关键步骤：

通过shmget（）函数创建共享内存和信号量

通过shmat（）函数将共享内存映射到地址空间中

生产者将资源写入共享内存

消费者从共享资源获得资源

7、在消息队列中，消息发送端进程和消息接收端进程之间为什么不需要额外实现进程间同步的机制？

因为消息队列本身提供进程间同步的机制。

**二、实验心得**

通过此次实验，我们了解了进程间通信和相应的函数的使用方法，掌握了各种管道，信号，信号量，消息队列，内存共享等相关知识，为以后的Linux学习打下了基础。