西安交通大学

操作系统专题实验

任务与要求

计算机科学与技术学院

2024-08-20

# 1 openEuler 系统环境实验

此实验为必做实验。

## 1.1 实验目的

（1）熟悉基于鲲鹏架构弹性云服务器ECS上openEuler操作系统基本系统环境；

（2）理解进程的创建、进程地址空间的概念、父子进程资源的关系；

（3）观察进程调度，了解进程调度的过程，了解孤儿进程和僵尸进程的区别；

（4）理解线程与进程的关系，对等线程间的关系。

（5）理解并运用信号量及其PV操作、自旋锁实现线程间的同步互斥。

## 1.2 实验思想

openEuler基于Linux内核，支持多种架构，支持多种虚拟化技术，支持多种容器技术，支持多种云计算技术。openEuler采用Linux内核，通过了解openEuler系统下的shell命令和系统命令后，所学习的知识可以轻松迁移到其他Linux发行版本中，如Ubuntu。

通过配置基于鲲鹏架构弹性云服务器ECS上openEuler操作系统基本系统环境，运行shell命令查看系统信息以达到了解和使用openEuler操作系统的目的。

通过进程相关编程实验，编写和运行简单的进程、线程相关程序，理解进程与线程概念、进程空间与物理内存空间、进程调度、进程间变量管理、进程调用其他程序、如何实现正确的并发（同步互斥）等方面在操作系统中的实际操作。具体来讲，通过输出子进程和父进程的PID，观察进程调度，了解进程调度的过程，了解孤儿进程和僵尸进程的区别。观察并发进程中的全局变量改变，输出父子进程共享变量地址以了解物理地址与虚地址概念，从而理解进程地址空间的概念，了解关于地址绑定的基础知识。创建两个线程运行后体会线程与进程的关系、对等线程间的资源共享以及同步与互斥的知识。

## 1.3 实验内容

（1）在华为云上搭建openEuler操作系统环境，通过运行shell命令查看系统信息以了解并熟悉openEuler操作系统。

（2）熟悉操作命令、编辑、编译、运行程序。完成给定程序（图1.1所示的程序）的运行验证，多运行程序几次观察结果；去除wait后再观察结果并进行理论分析。

（2）在所给程序中添加一个全局变量，在父进程和子进程中分别对这个变量做不同操作并输出操作结果，对结果进行观察并解释，同时输出两种变量的地址观察并分析。

（3）修改程序，在子进程中调用system函数和在子进程中调用exec族函数以执行自己写的一段程序，在此程序中输出进程PID，进行比较分析。

（4）线程实验:在进程中给一变量赋初值并创建两个线程，在两个线程中分别对此变量循环五千次以上做不同的操作并输出结果 ；多运行几遍程序观察运行结果，如果发现每次运行结果不同，请解释原因并修改程序解决，考虑如何控制互斥和同步；在线程中调用system函数/exec族函数以执行自己写的一段程序，输出进程PID和线程TID, 进行比较分析。

（5）自旋锁实验：在进程中给一变量赋初值并成功创建两个线程；在两个线程中分别对此变量循环五千次以上做不同的操作（自行设计）并输出结果；使用自旋锁实现互斥和同步。

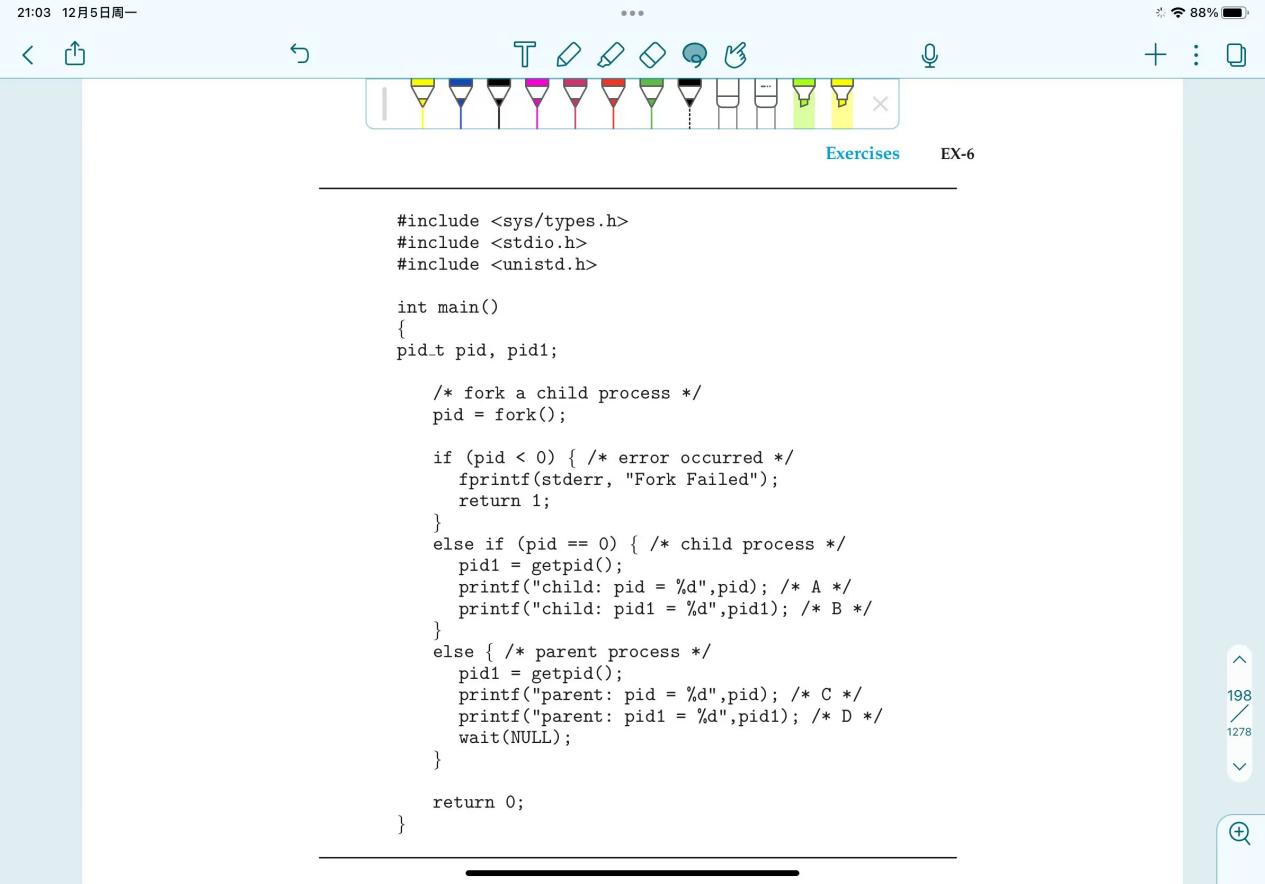


图1.1 给定程序

## 1.4 实验步骤

（1）登录华为云,搭建openEuler操作系统环境。

（2）编辑、编译、运行给定的程序，多次运行观察实验结果。

（2）去除wait()后观察结果并进行理论分析, 比较程序在有无wait()函数时的运行结果，分析wait()函数的作用。

（3）添加一个全局变量，在父进程和子进程中对这个变量做不同操作，输出操作结果，同时输出两种变量的地址并进行分析。

（4）在子进程中调用system函数执行自己写的一段程序，在此程序中输出进程PID和其父进程PID进行比较分析。

（5）在子进程中调用exec函数执行自己写的一段程序，在此程序中输出进程PID和其父进程PID进行比较分析。

（6）修改给定程序，给一变量赋初值并创建两个线程，在两个线程中分别对此变量循环5000次做不同的操作并输出结果。

（7）多运行几遍程序观察运行结果。

（8）完成上述过程的同步和互斥操作。

（9）在两个线程中调用system函数执行自己写的一段程序，在此程序中输出线程TID及进程PID,进行分析。

（10）在两个线程中调用exec函数执行自己写的一段程序，在此程序中输出线程TID及进程PID,进行分析。

（11）参考实验指导书，编写模拟自旋锁程序代码，补充主函数代码，用自旋锁实现线程间的同步。编译并运行程序，分析运行结果

## 1.5实验报告内容要求

报告中运行结果与分析部分，要给出每一步实验的过程和输出结果，并对结果进行分析，分析的内容包括fork()不同返回值的含义、进程地址空间的概念（父子进程资源的关系、为什么同一个变量会有不同的值）、wait()的作用、孤儿进程、并发进程的调度、system与exec函数的不同、线程与进程的不同、如何实现线程间的同步与互斥等。

# 2 进程通信与内存管理

此实验中，进程的软中断通信和管道通信为必做实验，内存的分配与回收和页面的置换为二选一实验。

## 2.1进程的软中断通信

2.1.1实验目的

编程实现进程的创建和软中断通信，通过观察、分析实验现象，深入理解进程及进程在调度执行和内存空间等方面的特点，掌握在POSIX 规范中系统调用的功能和使用。

2.1.2实验内容

（1）使用man命令查看fork 、kill 、signal、sleep、exit系统调用的帮助手册。

（2）根据流程图（如图2.1所示）编制实现软中断通信的程序：使用系统调用fork()创建两个子进程，再用系统调用signal()让父进程捕捉键盘上发出的中断信号（即5s内按下delete键或quit键），当父进程接收到这两个软中断的某一个后，父进程用系统调用kill()向两个子进程分别发出整数值为16和17软中断信号，子进程获得对应软中断信号，然后分别输出下列信息后终止：

Child process 1 is killed by parent !!

Child process 2 is killed by parent !!

父进程调用wait()函数等待两个子进程终止后，输出以下信息，结束进程执行：

Parent process is killed!!

**注： delete 会向进程发送 SIGINT 信号，quit 会向进程发送 SIGQUIT 信号。 ctrl+c 为 delete，ctrl+\为 quit 。**

**参考资料https://blog.csdn.net/mylizh/article/details/38385739**

（3）多次运行所写程序，比较5s内按下Ctrl+\或Ctrl+ c发送中断，或5s内不进行任何操作发送中断，分别会出现什么结果？分析原因。

（4）将本实验中通信产生的中断通过14 号信号值进行闹钟中断，体会不同中断的执行样式，从而对软中断机制有一个更好的理解。

2.1.3实验报告内容要求

报告中运行结果与分析部分，请回答下列问题。

（1）你最初认为运行结果会怎么样？写出你猜测的结果。

（2）实际的结果什么样？有什么特点？在接收不同中断前后有什么差别？请将5秒内中断和5秒后中断的运行结果截图，并对产生该现象的原因进行分析。

（3）改为闹钟中断后，程序运行的结果是什么样子？与之前有什么不同？

（4）kill 命令在程序中使用了几次？每次的作用是什么？执行后的现象是什么？

（5）使用kill 命令可以在进程的外部杀死进程。进程怎样能主动退出？这两种退出方式哪种更好一些？

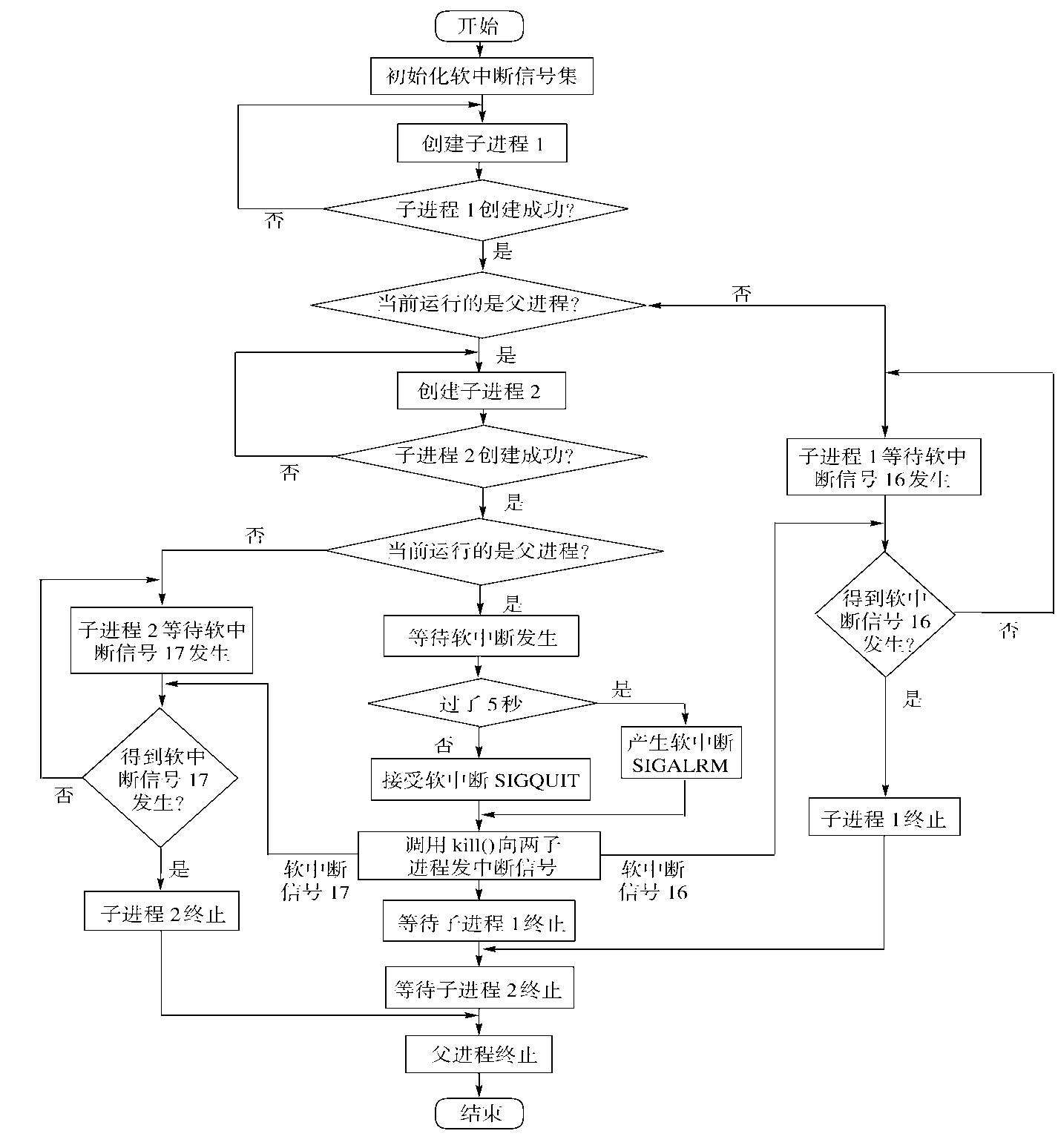


图2.1软中断通信程序流程图

## 2.2进程的管道通信

2.2.1实验目的

编程实现进程的管道通信，通过观察、分析实验现象，深入理解进程管道通信的特点，掌握管道通信的同步和互斥机制。

2.2.2实验内容

（1）学习man 命令的用法，通过它查看管道创建、同步互斥系统调用的在线帮助，并阅读参考资料。

（2）根据流程图（如图2.2所示）和所给管道通信程序，按照注释里的要求把代码补充完整，运行程序，体会互斥锁的作用，比较有锁和无锁程序的运行结果，分析管道通信是如何实现同步与互斥的。

（3）修改上述程序，让其中两个子进程各向管道写入2000个字符，父进程从管道中读出，分有锁和无锁的情况。运行程序，分别观察有锁和无锁情况下的写入读出情况。

2.2.3实验报告内容要求

报告中运行结果与分析部分，请回答下列问题。

(1)你最初认为运行结果会怎么样？

(2)实际的结果什么样？有什么特点？试对产生该现象的原因进行分析。

(3)实验中管道通信是怎样实现同步与互斥的？如果不控制同步与互斥会发生什么后果？

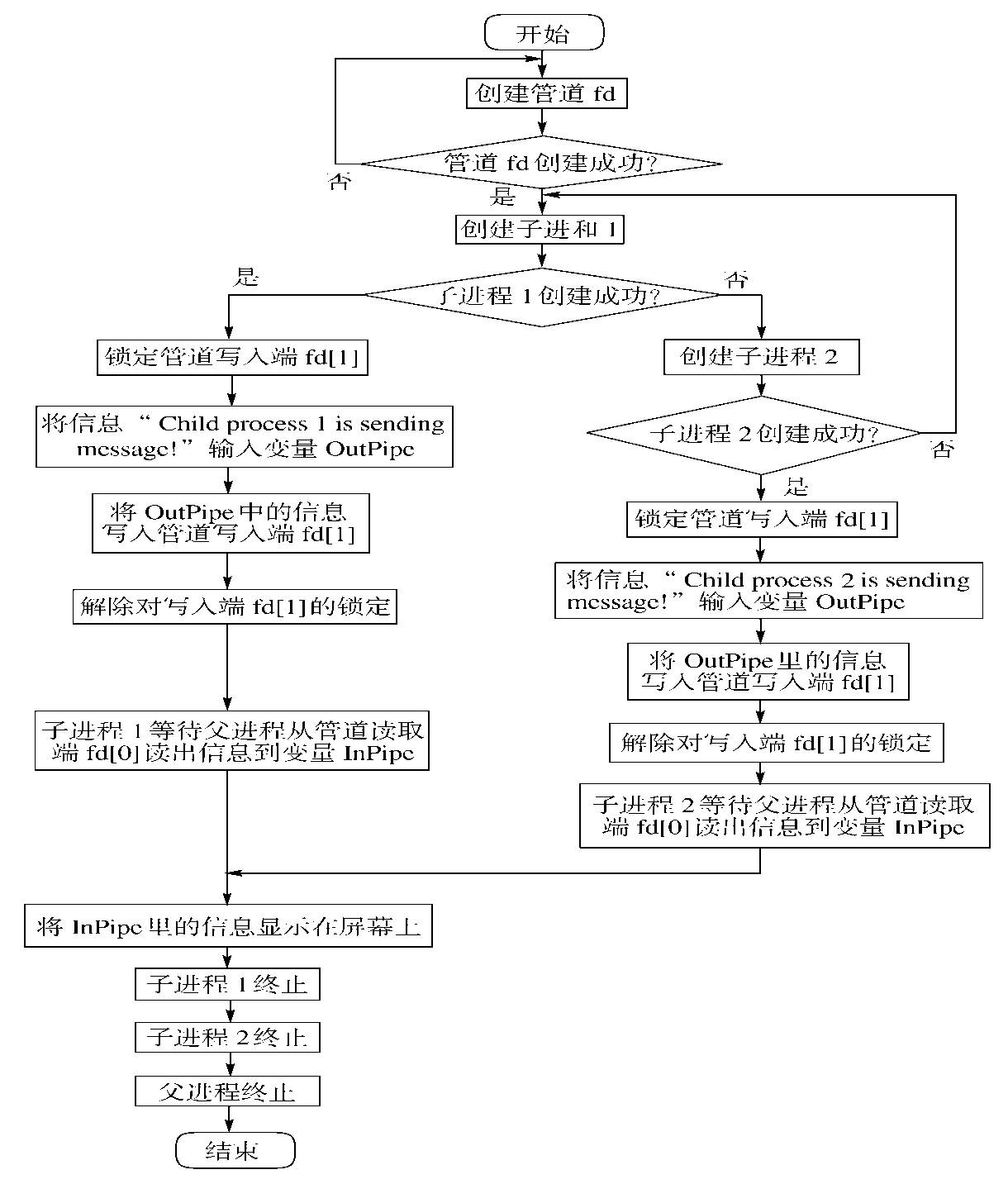


图2.2管道通信程序流程图

管道通信程序残缺版：

**/\*管道通信实验程序残缺版 \*/   
 #include <unistd.h>**

**#include <signal.h>**

**#include <stdio.h>**

**int pid1,pid2; // 定义两个进程变量**

**main( ) {**

**int fd[2];**

**char InPipe[1000]; // 定义读缓冲区**

**char c1=‘1’, c2=‘2’;**

**pipe(fd); // 创建管道**

**while((pid1 = fork( )) == -1); // 如果进程1创建不成功,则空循环**

**if(pid1 == 0) { // 如果子进程1创建成功,pid1为进程号**

**补充； // 锁定管道**

**补充; // 分2000次每次向管道写入字符’1’**

**sleep(5); // 等待读进程读出数据**

**补充; // 解除管道的锁定**

**exit(0); // 结束进程1**

**}**

**else {**

**while((pid2 = fork()) == -1); // 若进程2创建不成功,则空循环**

**if(pid2 == 0) {**

**lockf(fd[1],1,0);**

**补充; // 分2000次每次向管道写入字符’2’**

**sleep(5);**

**lockf(fd[1],0,0);**

**exit(0);**

**}**

**else {**

**补充; // 等待子进程1 结束**

**wait(0); // 等待子进程2 结束**

**补充; // 从管道中读出4000个字符**

**补充; // 加字符串结束符**

**printf("%s\n",InPipe); // 显示读出的数据**

**exit(0); // 父进程结束**

**}**

**}**

**}**

## 2.3内存的分配与回收

2.3.1实验目的

通过设计实现内存分配管理的三种算法（FF，BF，WF），理解内存分配及回收的过程及实现思路，理解如何提高内存的分配效率和利用率。

2.3.2实验内容

（1）理解内存分配FF，BF，WF策略及实现的思路。

（2）参考给出的代码思路，定义相应的数据结构，实现上述3种算法。每种算法要实现内存分配、回收、空闲块排序以及合并、紧缩等功能。

（3）充分模拟三种算法的实现过程，并通过对比，分析三种算法的优劣。

2.3.3实验报告内容要求

在实验总结部分，要包括如下内容：

（1）对涉及的3个算法进行比较，包括算法思想、算法的优缺点、在实现上如何提高算法的查找性能。

（2）3种算法的空闲块排序分别是如何实现的。

（3）结合实验，举例说明什么是内碎片、外碎片，紧缩功能解决的是什么碎片。

（4）在回收内存时，空闲块合并是如何实现的。

## 2.4 页面的置换

2.4.1实验目的

通过模拟实现页面置换算法（FIFO、LRU），理解请求分页系统中，页面置换的实现思路，理解命中率和缺页率的概念，理解程序的局部性原理，理解虚拟存储的原理。

2.4.2实验内容

（1）理解页面置换算法FIFO、LRU的思想及实现的思路。

（2）参考给出的代码思路，定义相应的数据结构，在一个程序中实现上述2种算法，运行时可以选择算法。算法的页面引用序列要至少能够支持随机数自动生成、手动输入两种生成方式；算法要输出页面置换的过程和最终的缺页率。

（3）运行所实现的算法，并通过对比，分析2种算法的优劣。

（4）设计测试数据，观察FIFO算法的BLEADY现象；设计具有局部性特点的测试数据，分别运行实现的2种算法，比较缺页率，并进行分析。

2.4.3实验报告内容要求

在实验总结部分，要包括如下内容：

（1）从实现和性能方面，比较分析FIFO和LRU算法。

（2）LRU算法是基于程序的局部性原理而提出的算法，你模拟实现的LRU算法有没有体现出该特点？如果有，是如何实现的？

（3）在设计内存管理程序时,应如何提高内存利用率。

# 3 开放实验

本实验不限定题目，学生可以从给定题目中选择一道来完成，也可以自拟一道题目来完成，如果是自拟题目，要求难度和工作量与给定题目相当。

## 3.1 文件系统实验

3.1.1实验目的

通过一个简单文件系统的设计，理解文件系统的内部结构、基本功能及实现。

3.1.2实验思想

在分析 Linux 的文件系统的基础上，基于Ext2的思想和算法，设计一个类Ext2的虚拟多级文件系统，实现Ext2文件系统的一个功能子集。并且用现有操作系统上的文件来代替硬盘进行硬件模拟。设计文件系统应该考虑的几个层次：①介质的物理结构；②物理操作——设备驱动程序完成；③文件系统的组织结构（逻辑组织结构）；④对组织结构其上的操作；⑤为用户使用文件系统提供的接口。本实验只涉及后三个层次的内容。

3.1.3实验内容

（1）分析EXT2文件系统的结构；

（2）基于Ext2的思想和算法，设计一个类Ext2的多级文件系统，实现Ext2文件系统的一个功能子集；

（3）用现有操作系统上的文件来代替硬盘进行硬件模拟。

3.1.4实验报告内容要求

实验报告中，要给出所设计的文件系统架构图、主要数据结构的定义、主要函数的设计思路。

## 3.2 Linux的动态模块与设备驱动

3.2.1实验目的

(1)学习内核通过动态模块来动态加载内核新功能的机制；

(2)理解LINUX字符设备驱动程序的基本原理；

(3)掌握字符设备的驱动运作机制；

(4)学会编写字符设备驱动程序；

(5)学会编写用户程序通过对字符设备的读写完成不同用户间的通信。

3.2.2实验内容

(1)编译、安装与卸载动态模块；

(2)实现系统调用的篡改；

(3)编写一个简单的字符设备驱动程序，以内核空间模拟字符设备，完成对该设备的打开、读写和释放操作；

(4)编写聊天程序实现不同用户通过该设备的一对一、一对多、多对多聊天。

3.2.3实验报告内容要求

报告中要说明驱动程序的实现；用户程序的实现；如何实现读写的同步与互斥；一对一、一对多、多对多的聊天分别是如何实现的。

## 3.3 其他可选题目

(1)设计实现一个最小操作系统内核

开发PC机或笔记本的基于USB盘的引导程序，设计内核任务及其系统调用。

(2)为Linux设计实现一个新的调度程序

在分析Linux内核调度程序的基础上，将其进程调度算法改为随机调度或其他调度算法。

(3) 模拟实现虚拟存储

分析研究虚拟存储的思想，模拟实现虚拟存储（要包括数据缓冲、交换文件的设计）。