**广州大学学生实验报告**

**开课学院及实验室：**计算机科学与网络工程学院软件实验室 **2023年 5月 日**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学院** | **计算机科学与网络工程学院** | **年级/专业/班** | **软件211** | **姓名** | 张景致 | **学号** | 32106300004 |
| **实验课程名称** | 操作系统实验 | | | | | **成绩** |  |
| **实验项目名称** | 进程管理与进程通信 | | | | | **指导老师** |  |

**实验一 进程管理与进程通信**

1. **实验目的**

1、掌握进程的概念，明确进程的含义。

2、认识并了解进程并发执行的实质，进程的阻塞与唤醒，终止与退出的过程。

3、熟悉进程的睡眠、同步、撤消等进程控制方法。

4、分析进程竞争资源的现象，学习解决进程互斥的方法 。

5、了解什么是信号，利用信号量机制熟悉进程间软中断通信的基本原理，

6、熟悉消息传送的机理 ，共享存储机制 。

1. **实验环境**

Linux archlinux 6.3.1-zen1-1-zen #1 ZEN SMP PREEMPT\_DYNAMIC Mon, 01 May 2023 17:42:12 +0000 x86\_64 GNU/Linux

gcc (GCC) 13.1.1 20230429

Copyright © 2023 Free Software Foundation, Inc.

1. **实验内容**
2. **实验原理 实验中用到的系统调用函数（包括实验原理中介绍的和自己采用的），实验步骤。**

本实验的用到的主要系统调用函数：

fork, exec, wait, exit, getpid, sleep, lockf, kill, signal, read, write, msgget, msgsnd, msgrcv, msgctl，shmget, shmat, shmdt, shmctl

***·实验步骤·***

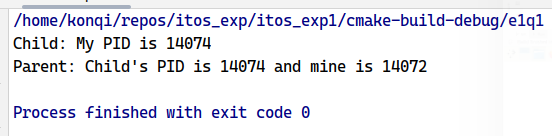
共享头文件common.h：  
#ifndef **ITOS\_EXP1\_COMMON\_H**#define **ITOS\_EXP1\_COMMON\_H**#include <stdio.h>  
#include <string.h>  
#include <stdlib.h>  
  
#include <fcntl.h>  
#include <unistd.h>  
  
#include <sys/wait.h>  
#include <sys/file.h>  
#include <sys/ipc.h>  
#include <sys/msg.h>  
#include <sys/shm.h>  
#include <sys/mman.h>  
  
#endif *//ITOS\_EXP1\_COMMON\_H*

1. 编写一段程序，使用系统调用fork( )创建两个子进程。当此程序运行时，在系统中有一个父进程和两个子进程并发执行，观察实验结果并分析原因。

所用代码：

#include "common.h"  
  
int main() {  
 pid\_t pid = fork();  
 if (pid == 0) {  
 *// child* printf("Child: My PID is %d\n", getpid());  
 } else if (pid > 0) {  
 *// parent* wait(0);  
 printf("Parent: Child's PID is %d and mine is %d\n", pid, getpid());  
 } else {  
 *// fork failed* printf("Fork failed.");  
 }  
 return 0;  
}

运行结果：



原因分析：

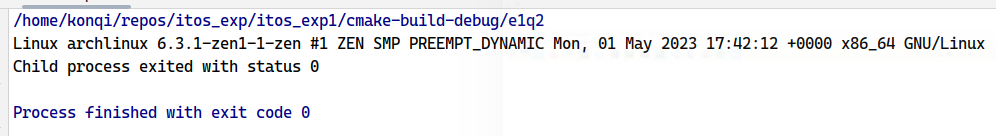
fork系统调用成功发生后，程序分为两个代码上相同的子母进程运行，而母进程从fork系统调用获得了子进程的pid。

1. 用fork( )创建一个进程，再调用exec( )，用新的程序替换该子进程的内容，利用wait( )来控制进程执行顺序，掌握进程的睡眠、同步、撤消等进程控制方法，并根据实验结果分析原因。

所用代码：

#include "common.h"  
  
int main() {  
 pid\_t pid = fork();  
 if (pid == 0) {  
 *// child process* char \*args[] = {"/usr/bin/uname", "-a", NULL};  
 int ret = execv(args[0], args);  
 if (ret == -1) {  
 perror("execv");  
 exit(**EXIT\_FAILURE**);  
 }  
 } else if (pid > 0) {  
 *// parent process* int status;  
 wait(&status);  
 if (**WIFEXITED**(status)) {  
 printf("Child process exited with status %d\n", **WEXITSTATUS**(status));  
 }  
 } else {  
 *// fork failed* perror("fork");  
 exit(**EXIT\_FAILURE**);  
 }  
 return 0;  
}

运行结果：



原因分析：

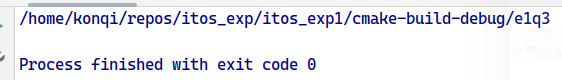
fork调用成功后，子进程execv调用启动了uname -a命令，母进程通过wait等待并获取了进程退出状态。另设有检查系统调用成功与否的结构。

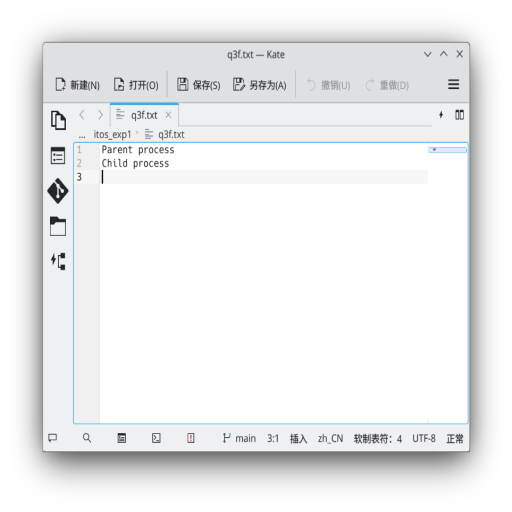
1. 编写一段多进程并发运行的程序，用lockf( )来给每一个进程加锁，以实现进程之间的互斥，观察并分析出现的现象及原因。

所用代码：

#include "common.h"  
  
int main() {  
 pid\_t pid;  
 int fd;  
 char buf[20];  
  
 fd = open("q3f.txt", **O\_WRONLY** | **O\_CREAT**, 0644);  
 if (fd == -1) {  
 perror("open");  
 exit(**EXIT\_FAILURE**);  
 }  
  
 pid = fork();  
 if (pid == -1) {  
 perror("fork");  
 exit(**EXIT\_FAILURE**);  
 } else if (pid == 0) {  
 *//lock* if (lockf(fd, **F\_LOCK**, 0) == -1) {  
 perror("lockf");  
 exit(**EXIT\_FAILURE**);  
 }  
  
 sprintf(buf, "Child process\n");  
 write(fd, buf, strlen(buf));  
  
 *//unlock* if (lockf(fd, **F\_ULOCK**, 0) == -1) {  
 perror("lockf");  
 exit(**EXIT\_FAILURE**);  
 }  
  
 exit(**EXIT\_SUCCESS**);  
 }  
  
 *//lock* if (lockf(fd, **F\_LOCK**, 0) == -1) {  
 perror("lockf");  
 exit(**EXIT\_FAILURE**);  
 }  
  
 sprintf(buf, "Parent process\n");  
 write(fd, buf, strlen(buf));  
  
 *// unlock* if (lockf(fd, **F\_ULOCK**, 0) == -1) {  
 perror("lockf");  
 exit(**EXIT\_FAILURE**);  
 }  
  
 exit(**EXIT\_SUCCESS**);  
}

运行结果：





原因分析：

首先一个文件被打开，然后子进程被创建。子母进程分别在尝试使用sprintf写入内容前使用lockf锁定了文件，使用完之后释放之。

4、编写程序：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来的中断信号（即按^c键）；捕捉到中断信号后，父进程用系统调用kill( )向两个子进程发出信号，子进程捕捉到信号后分别输出下列信息后终止：

Child process1 is killed by parent!

Child process2 is killed by parent!

父进程等待两个子进程终止后，输出如下的信息后终止：

Parent process is killed!

分析利用信号量机制中的软中断通信实现进程同步的机理。

5、使用系统调用msgget( ),msgsnd( ),msgrev( ),及msgctl( )编制一长度为1k的消息发送和接收的程序，并分析消息的创建、发送和接收机制及控制原理。

6、编制一长度为1k的共享存储区发送和接收的程序，并设计对该共享存储区进行互斥访问及进程同步的措施，必须保证实现正确的通信。

1. **实验结果分析（截屏的实验结果，与实验结果对应的实验分析）**

1、实验结果与实验程序、实验步骤、实验原理、操作系统原理的对应分析；

2、不同条件下的实验结果反应的问题及原因；

3、实验结果的算法时间、效率、鲁棒性等性能分析。

**六、实验总结**

**七、实验数据及源代码（学生必须提交自己设计的程序源代码，并有注释，源代码电子版也一并提交），包括思考题的程序。**

**注意： 实验报告文件名 学号-姓名-实验1-班级**

**实验数据与源代码 一个压缩包，名字和实验报告规则一样，需要有一个说明文件解释各个文件是什么文件。**

学委应把实验报告放到一个文件夹中，做成一个压缩包提交；实验数据和源代码，一个同学一个压缩包，形成一个文件夹，压缩后提交