

**实 验（实训）报 告**

**项 目 名 称**  进程管理

**所属课程名称**  操作系统

**项 目 类 型**  验证/设计型

**实验(实训)日期**  2024/11/12

**班 级**  22软件1

**学 号**  220110900128

**姓 名**  汤欣玥

**指导教师**  陈伟锋

浙江财经大学教务处制

|  |
| --- |
| **一、实验（实训）概述：（简单复述实验要求，1页以内，红字部分应删除）** |
| **【目的及要求】**  **实验五**  **1.安装gcc**  **2.完成hello.c**  **3.完成fork.c**  **4.完成proceses.c**  **5.完成communication.c**  **实验六**  **用c语言实现对n个进程采用动态优先权算法的调度**  **【基本原理】**  **1.初始优先权：每个进程在创建时会被赋予一个初始优先权。**  **2.动态调整：**  **等待时间：进程等待时间越长，其优先权越高。**  **服务时间：进程已经占用CPU的时间越长，其优先权越低。**  **其他因素：如I/O操作次数、内存使用情况等，也可以影响优先权。**  **3.调度：每次选择优先权最高的进程运行。【实施环境】**  **【软件环境】**  **- 虚拟化软件: VMware Workstation Pro 16.2**  **- Ubuntu版本: Ubuntu 23.10.1 (amd64)**  **Dev c++** |
| **二、实验（实训）内容：** |
| **【实验（实训）过程】（步骤、记录、数据、程序等）**  **1. 完成hello.c**  **打开终端，创建文件夹：cd hello\_world.创建源文件：nano hello.c**  **printf("Hello, World!\n");**  **编译源文件：gcc -o hello hello.c**  **运行程序：输入./hello**    **2. 完成fork.c**  **// 创建一个新的进程**  **pid = fork();**  **if (pid < 0) { // fork失败**  **fprintf(stderr, "Fork Failed");**  **return 1;**  **} else if (pid == 0) { // 子进程**  **printf("This is the child process. PID: %d\n", getpid());**  **} else { // 父进程**  **printf("This is the parent process. PID: %d, Child PID: %d\n", getpid(), pid);**  **}**    **3.完成proceses.c**  **void runProcess(const char \*name) {**  **printf("Running process %s...\n", name);**  **sleep(2); // 模拟进程运行2秒**  **printf("Process %s finished running.\n", name);**  **}**    **4.完成communication.c**  **if (cpid == -1) {**  **fprintf(stderr, "Fork failed");**  **return 1;**  **}**  **if (cpid > 0) { // 父进程**  **close(pipefd[0]); // 关闭读端**  **write(pipefd[1], writeMsg, (strlen(writeMsg) + 1)); // 写消息到管道**  **close(pipefd[1]); // 关闭写端**  **wait(NULL);**  **} else { // 子进程**  **close(pipefd[1]); // 关闭写端**  **read(pipefd[0], readMsg, sizeof(readMsg));**  **close(pipefd[0]); // 关闭读端**  **printf("Received message: %s\n", readMsg);**  **}**    **5.用c语言实现对n个进程采用动态优先权算法的调度**  **建立PCB的结构体--存储进程信息**  **typedef struct PNode{**  **int id;**  **int priority;**  **int cputime;**  **int alltime;**  **int startblock;**  **int fstartblock;/**  **int blocktime;**  **int fblocktime;//与fstartblock同理，设此存储初始状态。**  **int state;//1为就绪，0为阻塞**  **PNode \*next;**  **}PCB;**  **创建就绪与阻塞队列**    **根据进程参数对进程进行删除或移动**        **【结论与讨论】（结果、分析）**  **熟悉了vim的功能和C语言运行的相关操作以及进程调度的含义，通过模拟加深对进程概念和调度过程的理解。** |
| **三、指导教师评语及成绩：** |
| **评语：**  **成绩： 指导教师签名：**  **批阅日期：** |