

**实 验（实训）报 告**

**项 目 名 称**  （实验名称）

**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**  2024.10.18

**班 级**  22软件工程2

**学 号**  22011090135

**姓 名**  谢尚钦

**指导教师**  陈伟锋

浙江财经大学教务处制

|  |
| --- |
| **一、实验（实训）概述：** |
| **【目的及要求】**  一、实验5：  （1）安装gcc  （2）完成hello.c  （3）完成fork.c  （4）完成proceses.c  （5）完成communication.c  二、实验6：进程调度模拟  用C语言实现对N个进程采用动态优先权算法的调度  默认输出进程调度序列的id，以空格分隔  当输入的最后一行为1时，输出每一步的系统状态，示例如下  1730267356329  **【基本原理】**  **【实施环境】**  **Linux Ubuntu,** |
| **二、实验（实训）内容：** |
| **【实验（实训）过程】（步骤、记录、数据、程序等）**  一、实验5：  （1）安装gcc  安装gcc,这里我们实际上安装的是"build-essential"，它包含了 GNU 编辑器集合，GNU 调试器，和其他编译软件所必需的开发库和工具。下面这个命令将会安装一系列软件包，包括gcc，g++，和make。  执行  1729058587188  结果1729058670948  检查验证 GCC 编译器是否被成功地安装：  1729058712893   1. 完成hello.c   第一步：vi创建 hello.c  1729061065450  第二步：为hello.c输入以下内容  1729061102990  第三步：编译程序。执行命令：  **1729061160808**  第四步：**执行程序，也就是运行程序。输入命令：**  **1729061205623**   1. 完成fork.c   在终端中输入vim fork.c  同样输入  1729662328682  然后·esc 输入：wq    然后1729662381957  再输入1729662405922  得到结果  1729662438602   1. 完成proceses.c   终端中输入 vim processes.c  输入1729662528994  退出保存esc :wq  输入+得到1729662654958  可以用ctrl+c中断  1729662700264  （5）完成communication.c  输入vim communication.c  输入1729662825632  1729662846841  退出保存esc ：wq  1729662883007  输入gcc communication.c -o communication  ./communication  1729662992717  二、实验6：进程调度模拟  用C语言实现对N个进程采用动态优先权算法的调度  默认输出进程调度序列的id，以空格分隔  1.首先设计定义变量  进程标识号：PID  进程名：PNAME  进程优先数SUPER，并规定优先数越大的进程，其优先权越高；假定在调度过程中，进程每运行一个时间片，其优先数减2；进程每在就绪队列中待一个时间片，则其优先数加1。(可自行设定动态优先数的变化规律)  进程已占用的CPU时间RUNTIME（rtime）。  进程最大需占用的CPU时间NEEDTIME(ntime)。当RUNTIME等于NEEDTIME时，进程运行完毕。  进程状态STATE。假设实验中的进程只有三种状态：就绪（Wait）、运行（Running）和完成（Finished）。  ————————————————  2.设计所需结构体和函数  定义进程控制块PCB  对进程进行优先级排列函数  输入进程控制块函数  获取就绪状态的进程数  展示进程各个变量的具体内容  查看进程信息，显示当前处于运行态的进程和处于就绪队列的进程  建立进程就绪函数(进程运行时间到，置就绪状态)  3.具体操作  （1）建立chap4.c并打开  1730268589772  （3）运行代码  1730268626866  (4)在C:/a.txt中查看结果  在文件中找到该文件  1730268746597  输出的结果  1730268773128  **【结论与讨论】（结果、分析）**  通过完成 hello.c、fork.c、processes.c 和 communication.c 四个程序，我成功地实现了进程的创建、执行、通信等操作，并加深了对进程概念的理解。  本次实验成功地实现了动态优先权算法的进程调度模拟。通过观察 C:/a.txt 文件中的输出结果，我们可以清晰地看到每个时间片内进程的状态变化、优先级调整以及调度顺序。实验结果表明，动态优先权算法能够有效地根据进程的优先级和运行时间进行调度，从而提高系统资源的利用率。 |
| **三、指导教师评语及成绩：** |
| **评语：**  **成绩： 指导教师签名：**  **批阅日期：** |