

**实 验（实训）报 告**

**项 目 名 称**  进程通信与进程调度

**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

**班 级**  22软件1班

**学 号**  220110900530

**姓 名**  徐玉清

**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】**  1.安装gcc/g++  2.完成hello.cpp  3.完成fork.cpp  4.完成process.cpp  5.完成communication.cpp  6.用C语言实现对N个进程采用动态优先权算法的调度  **b674a15e752223866c5459732dac92e**  **【基本原理】**  1.gcc 是一个用于编译 C 和 C++ 程序的编译器，  在ubuntu系统中安装gcc  `  sudo add-apt-repository ppa:ubuntu-toolchain-r/test  sudo apt update  sudo apt install g++-7 -y  `  2.fork.cpp 是一个使用 fork() 系统调用的 C 程序，用于创建一个新的进程  3.动态优先权调度算法是一种基于进程优先级的调度算法，其中进程的优先级可以随着时间和进程状态的变化而变化，用于确保重要的进程能够获得更多的CPU时间。  基本原理:  - 每个进程都有一个与之关联的优先级。  - 调度器选择优先级最高的进程来执行。  - 进程的优先级可以根据特定的规则动态变化，例如，等待时间越长，优先级越高。  - 当进程完成其任务时，它的优先级会被重置或降低。  **【实施环境】**  **Ubuntu20.04.6LTS**  **Vscode gcc** |
| **二、实验（实训）内容：** |
| **【实验（实训）过程】（步骤、记录、数据、程序等）**  **实验五：**  1.安装gcc/g++  `  sudo add-apt-repository ppa:ubuntu-toolchain-r/test  sudo apt update  sudo apt install g++-7 -y  `    2.完成hello.cpp  `vim hello.cpp` --创建hello.cpp文件，并进入命令行编写    `g++ hello.cpp` --用g++编译器编译运行hello.cpp文件，生成a.out文件  `a,out` 执行a.out文件，输出”Hello World!“    3.完成fork.cpp  pid\_t 是一个在 POSIX 标准中定义的数据类型，用于存储进程标识符  此代码创建了一个新进程，如果fork()成功，子进程将执行ls命令，而父进程则等待子进程完成。如果fork()失败，则会打印错误信息并退出程序，如果execlp()失败，子进程也会打印错误信息并退出    运行结果：  执行了`ls`命令    4.完成process.cpp  此代码会导致父进程和子进程中的 value 变量独立变化，因为它们在各自的地址空间中。由于 fork() 调用，子进程会复制父进程的内存空间，所以初始时 value 的值和地址在两个进程中是相同的。随着循环的进行，value 的值会在两个进程中独立变化，value的地址是相同的    运行结果：    5.完成communication.cpp  pipe函数用于创建一个管道，它允许两个进程进行双向通信。管道是一种特殊的文件类型，它具有两个文件描述符：一个用于读，一个用于写  这段代码通过信号和管道实现了父子进程间的简单通信。父进程创建两个子进程，子进程1向管道写入信息，子进程2从管道读取信息。父进程等待两个子进程结束。同时，程序还处理了SIGINT和SIGUSR1信号，用于控制子进程的结束        运行结果：  b67bd956aa948af676a55127c55c6bf  6.用C语言实现对N个进程采用动态优先权算法的调度  **定义进程状态、进程控制块PCB结构体：**    **main()主函数**    **更新进程优先级和状态：**    **动态优先调度算法：**    **运行测试：**    这些输入对应于5个进程，每个进程的 priority, cputime, alltime, startblock, blocktime，以及最后的时间片大小    **程序将根据输入的进程属性和时间片大小执行动态优先权调度算法，并打印出每个进程的运行顺序和优先级**  **【结论与讨论】（结果、分析）**  通过这次操作系统进程通信与进程调度实验，我成功在ubuntu系统中配置了gcc环境，并完成了hello.cpp 的运行输出。学会了fork()系统调用的基本用法，  成功创建了子进程，并在父进程和子进程中分别打印了相应的信息，在process.cpp程序中，通过管道成功实现了父子进程间的通信。然后通过模拟动态优先权调度算法，了解了算法程序如何根据进程的优先级进行调度，以及优先级如何动态变化。动态优先权调度算法考虑了进程的优先级，这使得系统能够根据进程的重要性或紧迫性来分配 CPU 时间。对通过这些实验，自己对Linux的了解又多了一点，为以后在Linux游玩奠定基础。 |
| **三、指导教师评语及成绩：** |
| **评语：**  **成绩： 指导教师签名：**  **批阅日期：** |