山东大学 软件 学院

**操作系统** 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202200300092 | 姓名：郑睿杰 | | 班级：大数据2班 |
| 实验编号：实验1 | | | |
| 实验题目：进程控制实验 | | | |
| 实验学时：4 | | 实验日期：2024.4.21 | |
| 实验目的：  加深对于进程并发执行概念的理解。实践并发进程的创建和控制方法。观察和  体验进程的动态特性。进一步理解进程生命期期间创建、变换、撤销状态变换的过程。掌握进程控制的方法，了解父子进程间的控制和协作关系。练习 Linux 系统中进程创建与控制有关的系统调用的编程和调试技术。 | | | |
| 硬件环境：  机型：ThinkBook-13x-ITG  CPU：11th Gen Intel(R) Core(TM) i7-1160G7 @ 1.20GHz  内存：16150988 kB | | | |
| 软件环境：  Ubuntu 9.4.0-1 ubuntu1~20.04.2 + vscode2022 | | | |
| 实验步骤与内容：  【实验思路】  实验的核心思想是利用UNIX系统中的`fork()`系统调用来创建子进程，并在子进程中使用`opendir()`、`readdir()`和`closedir()`函数来周期性地读取并打印当前目录下的所有文件和目录。父进程则等待子进程结束，虽然在正常操作下子进程将持续运行直至外部干预（如使用CTRL+C中断）。  【实验模型描述】  实验模型包括两个主要组件：  1. \*\*父进程\*\*：  - 负责创建子进程。  - 等待子进程完成，回收资源。  2. \*\*子进程\*\*：  - 执行无限循环，每3秒执行一次目录读取和打印操作。  【主要算法代码分析】      在子进程中，list\_files函数负责打开当前目录，读取所有文件和子目录的名称，并打印它们。使用DIR类型和struct dirent结构体来操作目录。代码处理了错误情况，如无法打开目录。  【运行结果】    程序每3秒重复显示当前目录列表  【实验结果的分析综合】  实验成功实现了预定的目标：父进程创建了子进程，并由子进程周期性地列出当前目录中的文件和目录。这个简单的示例演示了进程创建、文件目录的访问以及基本的时间控制（通过`sleep`函数）。此外，程序在实际运行中不会自行终止，需要外部干预来结束子进程的运行。  总体来看，通过这次实验，可以加深对操作系统中进程控制和文件系统操作的理解。对于进一步的学习，可以探索进程间通信（IPC）机制，以及如何优雅地管理和终止子进程。 | | | |
| 结论分析与体会：  【实验结果达到设计目标的程度】  本实验成功实现了设计目标：创建一个父进程，它能生成一个子进程，子进程周期性地列出当前目录中的文件和文件夹。父进程正确地等待子进程结束（尽管实际操作中子进程设计为不会自行结束）。  【可进行的改进】  1. 优雅的中断处理：  - 可以为程序添加信号处理功能，使得子进程能在接收到特定信号（如 SIGINT）时安全地清理并退出，而不是仅仅依赖于外部强制终止。  2. 用户界面友好性：  - 增加参数解析，允许用户指定不同的目录或调整显示文件列表的时间间隔。  - 提供更多用户反馈信息，如打印出程序正在等待下一次目录扫描的提示。  3. 日志功能：  - 实现日志记录功能，将文件列表和可能的错误信息记录到文件中，便于事后分析和调试。  【实验收获与启发】  1. 深化理解进程控制：  - 通过亲自实现和调试父子进程的交互，加深了对进程生命周期和系统调用（如 fork(), wait()）的理解。  2. 文件系统操作实践：  - 通过使用 opendir(), readdir(), closedir()等函数，增强了对文件系统操作的实践经验。 | | | |