**C++课程设计**

**项目简要开发文档**

**专 业：软件工程**

**班 级：2021级**

**姓 名：王晓鑫**

**学 号：2021051304031**

**2024年6月**

1. **项目概述**

**项目名称：** Shell程序

**目标：** 实现一个基本的命令行界面Shell，支持命令输入与执行，脚本执行以及基本的IO重定向和管道功能。

**开发语言：** C++

**开发环境：**Linux Qt

1. **功能需求**

* **命令的输入与执行：** 用户可以输入命令，程序能够解析、执行并显示执行结果。
* **脚本执行：** 提供可shell编程的功能，支持通过**source**命令执行简单的shell脚本文件。
* **IO重定向：** 支持输入输出重定向，使用**>**和**<**符号实现。
* **管道支持：** 使用**|**符号实现命令之间的管道操作。

1. **技术实现**

**使用库：**<iostream>: 提供输入输出流功能。

<string>: 提供字符串处理功能。

<vector>: 提供动态数组功能。

<unistd.h>: 提供POSIX API，包括文件描述符和进程控制。

<sys/wait.h>: 提供进程等待相关功能。

<fcntl.h>: 提供文件控制功能。

<fstream>: 提供文件流操作。

<sstream>: 提供字符串流操作。

**进程管理：**fork(): 用于创建子进程。

execvp(): 在子进程中执行外部命令。

waitpid(): 等待子进程执行完毕，获取进程状态。

**文件操作：**

重定向：使用 open() 函数打开文件并获取文件描述符。

使用 dup2() 函数将文件描述符重定向到标准输入/输出。

关闭原文件描述符以释放资源。

**管道操作：**pipe(): 创建一个管道，返回一个包含读写端的文件描述符数组。

dup2(): 将管道的写端或读端重定向到标准输出/输入。

close(): 关闭不再使用的文件描述符。

1. **系统设计**

**4.1 主要组件**

**主函数 (main):** 程序入口，启动shell循环。

**Shell循环 (shell):** 处理用户输入，调用命令解析和执行。

**命令解析 (parseCommand):** 将字符串输入转换为命令和参数的向量。

**命令执行 (executeCommandWithRedirectionAndPipes):** 根据解析结果执行命令，管理I/O重定向和管道。

**脚本执行 (executeScript):** 从文件读取命令并执行。

**4.2 用例图**

**用户：** 输入命令、查看执行结果、使用重定向和管道功能、执行shell脚本以及退出shell程序。

图示

描述已自动生成

**4.3 流程图**

图示

描述已自动生成

流程图说明：

1. 入口点：程序开始执行，进入 shell 函数。
2. 用户交互：显示提示字符 "myshell> " 并等待用户输入命令。
3. 用户输入处理：当用户输入命令后，程序会读取整行输入并存储在 input 字符串中。
4. 命令解析：程序调用 parseCommand 函数将用户的输入解析成字符串数组 args。
5. 命令执行：程序根据用户输入决定是否执行命令。
   * 如果是 “exit” 命令，程序退出。
   * 如果是 “source” 命令，程序需要处理一个脚本文件。
   * 如果是其他命令，程序调用 executeCommandWithRedirectionAndPipes 函数执行该命令。
6. 执行解析的命令：executeCommandWithRedirectionAndPipes 函数根据命令参数中的重定向符号（> 和 <）及管道符号（|）来设置文件描述符（in\_fd 和 out\_fd）。
7. 并行执行或管道设置：检测是否有管道符号（|），如果有则创建管道并启动子进程，用于左侧的命令，同时关闭父进程的写入端，然后调用 waitpid 等待子进程完成。对于右侧的命令，父进程变为等待状态的子进程，并重定向标准输入到管道读取端。如果未检测到管道符号，则直接调用 fork 并执行子进程的命令，然后等待子进程结束。
8. 退出当前循环并返回：如果命令处理和执行完成，waitpid 返回，程序返回到用户交互步骤，等待下一次输入。
9. 程序结束：用户输入 “exit” 命令时，程序退出循环并结束执行。
10. **测试计划**

* **单元测试：** 对每个主要功能（命令解析、命令执行、IO重定向、管道操作等）进行单元测试。
* **集成测试：** 测试功能间的交互，如脚本执行中的命令序列、命令与IO重定向的组合等。
* **系统测试：** 测试整个系统的稳定性和性能。