

自定义Shell解释器的设计与实现

一、项目设计目的

- (1) 通过设计一个自定义的 Shell 解释器，加深对 Linux 系统的底层系统调用和解释器工作原理的理解。
- (2) 综合运用操作系统原理中所学的知识和 Linux 系统实验中锻炼的动手能力完成实验。

二、项目设计内容

- (1) **（基础）**编写程序实现一个自定义的 Linux 的 Shell 命令解释器，要求实现以下基本功能：
 - **命令帮助信息：**类似于 help 命令或 man 命令。该功能提供一个命令，用于显示设计的自定义 Shell 解释器支持的所有命令及其用法。用户可以随时调用此命令，以获取对其他命令的简要描述和正确使用方式的指导。
 - **显示指定目录文件：**类似于 ls 命令。该功能提供一个命令，用于列出指定目录下的所有文件和文件夹。用户可以通过这个命令查看当前工作目录中的内容。
 - **切换工作目录：**类似于 cd 命令。该功能提供一个命令，允许用户改变当前工作目录，切换到指定的路径。
 - **复制文件、文件夹：**类似于 cp 命令。该功能提供一个命令，允许用户将源文件或目录复制到指定的目标文件或目录中。
 - **移动文件、文件夹：**类似于 mv 命令。该功能提供一个命令，用户可以调用它来对文件或目录重新命名，或者将文件从一个目录移到另一个目录中。
 - **删除文件、文件夹：**类似于 rm 命令。该功能提供一个命令，用于删除文件或目录，也可以将某个目录及其下属的所有文件及其子目录递归删除。
 - **显示工作目录：**类似于 pwd 命令。该功能提供一个命令，用于显示当前工作目录的路径。用户可以通过调用此命令立刻获取目前所在的工作目

录的绝对路径名称。

- **显示输入命令历史：**类似于 `history` 命令。该功能提供一个命令，用户可以调用它来显示之前输入的命令历史记录，方便回顾和重用命令。
- **显示进程信息：**类似于 `ps` 命令。该功能提供一个命令，用户可以调用它来获取有关系统中运行进程的详细信息，例如进程 ID、状态和资源使用情况等。
- **显示目录结构：**类似于 `tree` 命令。该功能提供一个命令，以树状结构显示目录的内容。执行命令会列出指定目录下的所有文件，包括子目录里的文件。
- **输入输出重定向：**允许用户使用符号将命令的输入或输出重定向到文件，实现更灵活的输入输出控制。
- **管道：**类似于 `[]` 操作符，该功能允许用户将一个命令的输出作为另一个命令的输入，实现命令之间的数据流传递。
- **后台运行程序：**类似于 `[&]` 操作符，该功能使得某个程序在后台运行，允许用户继续输入其他命令而不阻塞 Shell。

(2) （进阶）在完成上述基础功能后，学有余力的同学可以进一步了解 Shell 解释器的底层设计，尝试在自定义的 Shell 解释器中实现更复杂的功能，如外部指令调用等。

(3) 对实现的功能逐一进行测试，确保程序的完整性和正确性。

三、项目提交

提交内容由三部分组成：项目设计报告、设计源代码、现场验收测试。

(1) 项目设计报告：

- 要求对设计的 Shell 解释器功能原理进行分析，并详细描述实验的设计思路。
- 提供实验运行过程的截图或流程图，报告中的截图应该清晰展示关键步骤，可以提供必要的流程图直观展示设计的整体结构框架。

- 对运行结果进行分析和总结。

(2) 设计源代码：

- 源代码应符合规范的编码风格
- 源代码中应包含必要的注释，解释关键功能、算法或设计决策
- 提供充分的测试用例，覆盖不同的功能和边界条件，确保程序的稳定性。

(3) 现场验收测试：

- 在现场演示中，学生需要演示实现的 Shell 解释器中的各项功能。
- 需要对每个实现的功能进行讲解，解释设计原理、相关算法和使用的系统调用。

四、评分标准

以小组为单位进行评分，评分将综合考虑项目设计报告的深度和完整性、设计源代码的质量和规范性、现场验收测试的演示效果和学生对实现功能原理的理解。