**构建说明文档**

**一、项目概述：**

本项目是一个简单的命令行程序，旨在演示基本的 shell 功能，包括执行命令、支持管道、重定向输入/输出以及更改当前工作目录。

**二、环境要求：**

操作系统：任何支持 POSIX 标准的 Unix-like 系统（如 Linux、macOS）。

编译器：C++ 编译器，如 GCC 或 Clang。

构建工具：CMake（版本 2.8 或更高）。

**三、获取源代码**

使用Git克隆项目源代码到本地：

git clone https://github.com/yourusername/demo-shell.git

**四、安装指南：**

确保你的系统上安装了CMake和C++编译器。

如果没有安装CMake，可以CMake下载并安装。

安装 CMake:访问CMake官网，下载并安装适合您操作系统的 CMake 版本。

安装 C++ 编译器：在 Linux 系统上，可以使用包管理器安装 GCC。

**五、配置步骤：**

本项目不需要额外的配置步骤。CMakeLists.txt 文件包含了所有必要的配置信息。

**六、构建过程：**

克隆或下载项目代码到本地。

打开命令行终端，切换到项目目录。

创建一个构建目录并切换到该目录：mkdir build/cd build

运行 CMake 以生成构建系统：cmake .

使用 make 命令编译项目：make

编译完成后，可执行文件main将位于构建目录中，可使用ls查看

**七、 运行项目：**

构建完成后，你可以运行生成的可执行文件：./main

**八、测试指南：**

在运行shell程序后，尝试执行一些基本命令，例如：

Myshell> ls 列出当前目录的文件和文件夹

Myshell> echo "Hello" 输出文本

Myshell> exit 退出

使用管道：Myshell> ls | grep log 列出所有.cpp文件

或先再目录下创建hello.txt,然后Myshell> echo "Hello" > hello.txt 在hello.txt中输出Hello

使用重定向：Myshell> cat < hello.txt

**九、常见问题：**

**构建时出现编译错误：**

请确保你的编译器支持C++11或更高版本。检查错误信息并根据提示进行修改。**如何添加新功能或修改现有代码：**

直接编辑main.cpp文件，然后重新构建项目。