

FTP服务器开发文档 副本

1. 设计概要

1.1 项目概述

简易FTP服务器项目是一个基于openEuler操作系统的数据传输工具，旨在为用户提供一个稳定、安全的文件上传和下载环境。该服务器能够监听指定端口，支持文件的上传和下载操作，并能同时处理多个并发请求，确保用户在多样化的网络环境中能够高效地进行数据交换。本项目特别强调用户权限管理，支持匿名及账号登录，并根据不同的用户角色实现精细化的权限控制。

在功能实现方面，简易FTP服务器项目具备以下特点：

1. 网络监听：服务器能够绑定到指定的网络端口，等待客户端的连接请求。
2. 文件操作：用户可通过服务器进行文件的上传和下载，满足基本的数据传输需求。
3. 并发处理：服务器设计考虑了多线程或多进程技术，能够同时处理多个用户的请求，提高系统资源利用率。
4. 用户认证与权限管控：服务器内置匿名账户及特定用户账户（user1、user2），实现基于角色的权限管理，确保数据安全。
5. 脚本支持：项目提供run-ftp.sh脚本，实现自动化安装、配置运行环境，以及编译和运行服务器程序。

根据权限配置要求，本项目实现了以下特定功能：

- （1）匿名账户：允许匿名用户查看并下载user1上传的文件，但不支持文件上传操作。
- （2）user1账户：具备查看、下载自己上传文件的能力，同时可以下载user2上传的文件。
- （3）user2账户：仅能查看和下载自己上传的文件，限制了文件访问范围。

简易FTP服务器项目适用于openEuler 24.03 LTS或麒麟信安Kylinsec V3.5.2操作系统，所有系统配置均采用默认设置，确保项目的通用性和易用性。通过本项目，我们旨在为openEuler用户提供一个简易、高效、安全的FTP解决方案，同时促进仓颉语言的普及和应用。

2. 技术方案

2.1 总体架构

简易FTP服务器项目，基于openEuler操作系统，采用仓颉（Cangjie）编程语言进行开发，从而构建一个高效稳定的数据传输平台。以下为该项目的技术方案概述，采用仓颉语言的特点进行设计：

项目核心架构设计围绕以下几个关键组件：

1. **网络通信框架**：利用仓颉语言的网络编程能力，实现服务器与客户端之间的稳定通信。负责监听端口、处理连接请求，并维持稳定的网络会话。
2. **文件操作库**：使用仓颉语言的文件I/O功能，开发文件上传和下载的接口。该库包括文件读写、权限检查、数据流控制等模块，确保文件操作的安全性和高效性。
3. **并发处理机制**：结合仓颉语言的并发编程特性，设计多线程或多进程模型，以支持服务器同时处理多个客户端请求，提升系统的并发处理能力。
4. **认证与权限管理系统**：基于仓颉语言的加密和安全模块，实现用户认证和权限控制。系统包括用户账户管理、密码加密存储、权限分配等功能。
5. **自动化部署脚本**：编写仓颉语言的脚本，用于服务器的自动化部署和启动。脚本将涵盖环境配置、依赖管理、编译构建等流程。

具体技术方案如下：

- **网络通信**：
 - 利用仓颉语言的网络库实现基于 TCP 的可靠数据传输。
 - 开发FTP命令解析器，支持FTP协议的标准命令集。
- **文件操作**：
 - 使用仓颉语言的文件操作API，实现文件的上传和下载逻辑。
 - 处理文件名中的特殊字符和长文件名问题。
- **并发处理**：
 - 利用仓颉语言的并发编程特性，为每个客户端连接创建独立的执行线程或进程。
 - 实现线程或进程池管理，优化资源使用。
- **认证与权限管理**：
 - 设计并实现用户认证机制，支持匿名登录和固定账户登录。
 - 实现密码加密存储和验证，保障账户安全。
 - 根据用户角色设定文件访问权限，实现细粒度的权限控制。
- **自动化部署**：
 - 编写仓颉语言脚本，实现服务器的快速部署和一键启动。
 - 脚本包含错误处理和日志记录，便于维护和问题诊断。
- **系统兼容性**：
 - 确保服务器在 openEuler 及其他兼容操作系统上稳定运行。
 - 支持主流硬件架构，如x86_64和aarch64。

通过采用仓颉语言开发，简易FTP服务器项目将展现其独特的编程优势，提供稳定可靠的数据传输服务，满足用户在 openEuler 系统上的文件传输需求，并确保系统的安全性和易用性。

2.2 功能设计

1. 网络通信模块

- **端口监听：**
 - 服务器应能够监听一个或多个预配置的端口。
 - 支持配置文件中设置监听端口。
- **连接管理：**
 - 接受来自客户端的连接请求。
 - 维护一个活跃连接列表，用于管理多个并发会话。
 - 限制最大连接数，防止资源耗尽。

2. 用户认证模块

- **匿名登录：**
 - 允许匿名用户登录，但限制其操作权限，如仅能下载公共文件。
- **账号登录：**
 - 支持用户名和密码认证。
 - 用户信息存储在配置文件或数据库中。
- **权限控制：**
 - 根据用户角色分配不同的文件操作权限。
 - 管理员用户拥有所有权限，普通用户权限受限。

3. 文件操作模块

- **文件上传：**
 - 客户端能够将文件上传到服务器指定目录。
 - 支持断点续传功能。
- **文件下载：**
 - 客户端能够从服务器下载文件。
 - 支持文件传输的压缩和解压缩。
- **文件管理：**
 - 列出服务器上的文件和目录。
 - 支持创建目录、删除文件、重命名文件等操作。

4. 并发处理模块

- **多线程/多进程支持：**

- 为每个客户端连接创建独立的线程或进程。
- 线程或进程池管理，以优化资源使用。
- **同步机制：**
 - 确保文件操作的原子性，避免并发访问导致的数据损坏。

5. 日志和监控模块

- **日志记录：**
 - 记录所有用户操作和系统事件。
 - 支持日志级别设置，便于调试和审计。
- **系统监控：**
 - 监控服务器性能指标，如CPU使用率、内存使用情况等。
 - 实时反馈服务器状态。

6. 安全性模块

- **数据加密：**
 - 支持FTP数据传输的加密，如FTP over SSL/TLS。
- **访问控制：**
 - 限制对特定目录的访问。
 - 防止目录遍历攻击。

7. 配置和管理模块

- **服务器配置：**
 - 通过配置文件设置服务器参数，如监听端口、最大连接数等。
 - 支持在线更新配置，无需重启服务器。
- **用户管理：**
 - 管理员能够添加、删除和修改用户账户。
 - 支持用户权限的动态调整。

3. 环境配置

- 查看openEuler是否符合仓颉工具链的系统环境要求

```
[root@openeuler ~]# ldd --version
ldd (GNU libc) 2.28
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Written by Roland McGrath and Ulrich Drepper.
[root@openeuler ~]# uname -r
4.19.90-2003.4.0.0036.oe1.x86_64
[root@openeuler ~]# rpm -qa | grep libstdc++
libstdc++-7.3.0-20190804.h31.oe1.x86_64
[root@openeuler ~]#
```

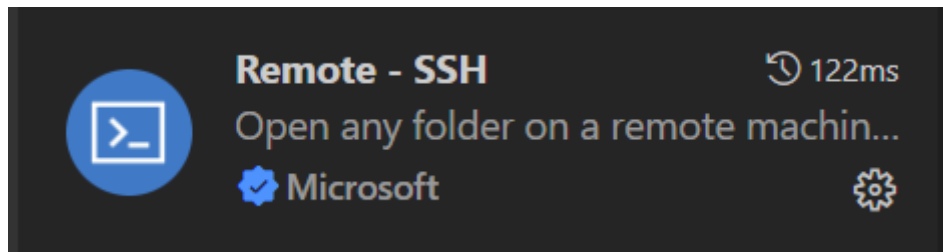
- 安装仓颉工具链，openEuler似乎并不需要安装仓颉工具链依赖。
- 运行第一个仓颉程序

```
[root@openeuler cangjieCode]# vim hello.cj
[root@openeuler cangjieCode]# cjc hello.cj -o hello
[root@openeuler cangjieCode]# ls
default.bchir2  default.cjo  hello  hello.cj
[root@openeuler cangjieCode]# hello
-bash: hello: command not found
[root@openeuler cangjieCode]# ./hello
你好，仓颉
```

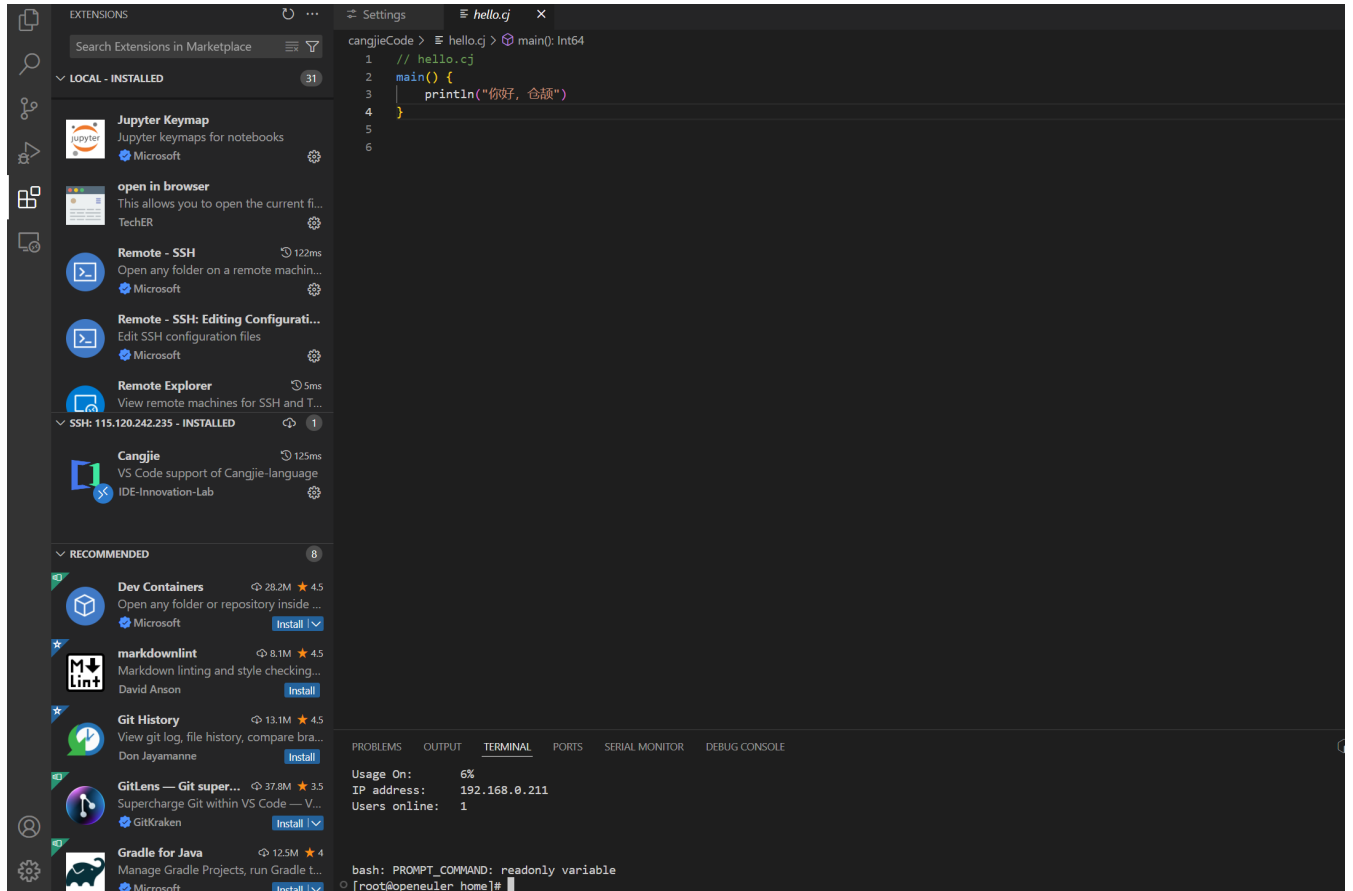
- 使用vscode搭建编码环境
 - 修改 `/etc/ssh/sshd_config` 文件中的 `AllowTcpForwarding` 为 `yes`

```
#CheckUserSplash yes
PermitRootLogin yes
PasswordAuthentication yes
UseDNS no
Protocol 2
LogLevel VERBOSE
PubkeyAuthentication yes
RSAAuthentication yes
IgnoreRhosts yes
RhostsRSAAuthentication no
HostbasedAuthentication no
PermitEmptyPasswords no
PermitUserEnvironment no
Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.c
om,chacha20-poly1305@openssh.com
ClientAliveCountMax 0
Banner /etc/issue.net
> MACs hmac-sha2-512,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha2-256-etm@ope
nssh.com,hmac-sha1,hmac-sha1-etm@openssh.com
StrictModes yes
35 AllowTcpForwarding yes
AllowAgentForwarding no
GatewayPorts no
PermitTunnel no
XKexAlgorithms curve25519-sha256,curve25519-sha256@libssh.org,diffie-hellman-group14-
sha1,diffie-hellman-group-exchange-sha1,diffie-hellman-group-exchange-sha256
```

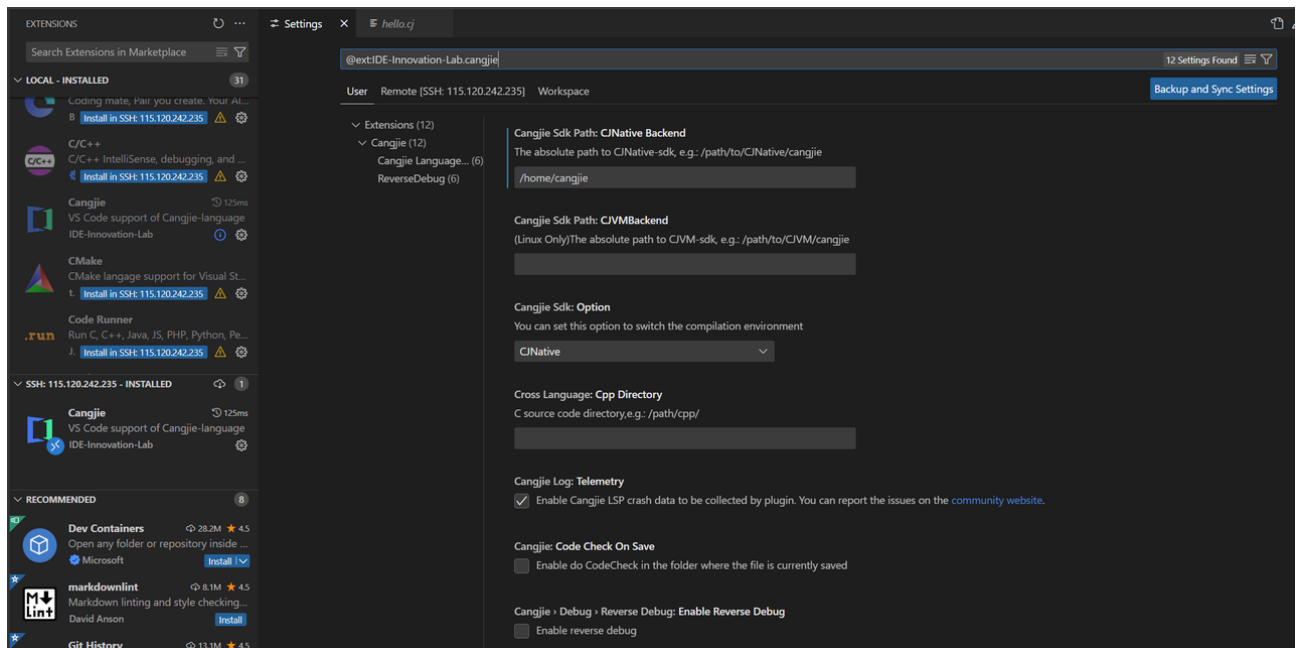
- 本地vscode安装相关远程连接插件



- 进行远程连接连接到远端的openEuler操作系统



- 为了更好的使用仓颉语言，还需安装仓颉插件
 - <https://cangjie-lang.cn/download/0.53.13>
 - 在上述安装地址中下载VScode Plugin并将其上传到openEuler系统，解压后通过VScode安装本地插件并配置插件的SDK的路径为此前下载解压的仓颉的路径



- 至此，我们在openEuler操作系统上搭建Cangjie语言的开发环境就已经完成。