

# 1. 项目概述

本项目旨在基于openEuler操作系统实现一个简易FTP服务器。该服务器将具备基本的文件上传、下载功能，能够处理多个并发请求，并支持匿名及账号登录，同时实现权限管控。

# 2. 开发环境

- 操作系统：openEuler 24.03 LTS
- 编程语言：Python

# 3. 功能需求

- 监听指定端口
- 文件上传、下载功能
- 处理多个并发请求
- 支持匿名及账号登录
- 实现权限管控

# 4. 权限配置

- 匿名账户：仅可查看并下载user1上传的文件，不可上传文件。
- user1账户：可查看并下载自己上传的文件，及下载user2上传的文件。
- user2账户：仅可查看并下载自己上传的文件。

# 5. 项目结构

```
python-ftp/
|
├── ftp_server.py      # FTP服务器主程序
├── run-ftp.sh         # 运行脚本
└── README.md          # 项目说明文档
```

## 6. 安装与配置

- **安装依赖**: 通过 run-ftp.sh 脚本自动安装Python和 pyftplib 库。
- **配置用户**: 在 ftp\_server.py 中配置匿名账户及user1、user2账户的权限。

## 7. run-ftp.sh 脚本内容

```
#!/bin/bash

# 安装 Python 和 pip
echo "确保 Python 和 pip 已安装..."
if ! command -v python3 &> /dev/null
then
    echo "Python 3 未安装, 尝试进行安装..."
    sudo dnf install -y python3
fi

if ! command -v pip3 &> /dev/null
then
    echo "pip 未安装, 尝试进行安装..."
    sudo dnf install -y python3-pip
fi

# 安装 pyftplib
echo "安装 pyftplib ..."
pip3 install pyftplib

# 检查目录结构
FTP_DIR="/ftp_data"
mkdir -p $FTP_DIR/user1 $FTP_DIR/user2

# 启动 FTP 服务器
echo "启动 FTP 服务器 ..."
python3 ftp_server.py &

echo "FTP 服务器已启动, 端口 21 监听中。"
```

## 8. ftp\_server.py 核心代码

```
# 定义权限控制
class CustomAuthorizer(DummyAuthorizer):
    def __init__(self):
        super().__init__()
        # 匿名用户 (主目录)
        self.add_anonymous('/ftp_data/user1')
        # user1 用户 (主目录)
        self.add_user("user1", "123456", homedir='/ftp_data', perm='elr')
        # user2 用户 (主目录)
        self.add_user("user2", "123456", homedir='/ftp_data/user2', perm='elradfmwMT')
        # user1 可以查看user2的文件
        self.override_perm("user1", "/ftp_data/user1", perm='elradfmwMT', recursive=True)
```

以上代码片段展示了如何配置匿名用户和两个普通用户的权限。其中，`DummyAuthorizer` 类用于自定义授权器，通过继承并重写其方法来实现特定的访问控制逻辑。

我们为不同用户设置了不同的权限，例如匿名用户只能查看并下载 `user1` 上传的文件，而 `user1` 可以查看自己的文件以及 `user2` 的公开文件。通过这种方式使得FTP服务器能够满足基本的权限管理需求。

## 9. 测试

- 使用FTP客户端连接到服务器，测试上传、下载功能。
- 验证不同账户的权限配置是否正确。

## 10. 总结

本项目提供了一个基于openEuler系统的简易FTP服务器实现。通过简单的配置和脚本，实现了基本的FTP服务功能，并满足了权限管理的需求。未来可根据需要扩展更多高级功能。