

**proj235-ota-upgrade-system**

# **说明书**

选题：proj235

学校：中北大学

队员：冯琦雁 刘晓敏 郭冰芯

指导老师：张静

## 目录

1. 项目描述 .....	3
2. 项目目的 .....	3
3. 项目结构 .....	4
4. 源代码说明 .....	4
5. 脚本说明 .....	5

# 1.项目描述

在工业控制、机器人控制领域中越来越多使用 Linux 嵌入式操作系统，但嵌入式 linux 系统在终端设备上部署之后，在更新和升级系统镜像的时候， 还需要重新物理部署，不能确保最佳的操作体验和系统正常的运行时间。为了改善 linux 的升级体验， 可以参考 ostree，改造 linux 的升级系统，以能够执行安全的远程 OTA 升级，执行安全的离线升级。

## 2.项目目的

### 第一题：升级模块基础框架实现

1. 使用 ostree 创建 repo 仓库
2. 脚本能将编译出来的 rootfs 和内核等发布到 ostree repo 仓库
3. linux 系统能够使用受 ostree 管理的 rootfs 启动正常(grub 方式)

### 第二题：升级模块更新功能实现

以题目一为基础，工具需要追加下列功能：

1. linux 系统正常启动后，可以获取远程 repo 仓库最新版本，执行 ostree 升级
2. linux 系统升级失败时，能回退到升级前的系统中
3. 三次启动失败，能够自动回退到升级前版本
4. 支持 A/B 分区，启动失败时可以从 B 分区启动

### （可选）第三题：升级模块扩展功能实现

以题目二为基础，工具需要追加下列功能：

1. ostree 客户端从 ostree repo 仓库获取最新版本时，要有安全认证，要能够支持并发测试
2. 只有持有指定密钥才能够向 ostree repo 仓库提交系统
3. ostree repoo 仓库可以接管多个操作系统
4. ostree repoo 仓库提供 webUI 进行管理
5. 支持 uboot 方式启动的 linux 系统

## 3.项目结构

该项目包含源代码、测试脚本和相关辅助脚本。`src/`目录存放源代码文件，包括程序的主要功能实现和工具函数；`tests/`目录存放测试脚本，用于验证各个功能模块的正确性；`scripts/`目录存放辅助脚本，用于执行具体的操作如创建 OSTree 仓库、发布内容、系统升级、回滚和分区切换等。通过这种结构，项目的各个部分职责明确，便于管理和维护。

## 4.源代码说明

### (1) main.c:

`main.c` 是程序的入口点，包含了所有模块的调用逻辑。

首先，打印出 "OSTree Upgrade Module"。

然后，调用 `authenticate` 函数进行用户认证。

如果认证成功，调用 `perform_upgrade` 和 `perform_rollback` 函数。

使用 `log_message` 函数记录操作的结果。

### (2) upgrade.c

`upgrade.c` 包含系统升级的具体实现。

`perform_upgrade` 函数执行升级脚本 `./scripts/perform_upgrade.sh`。

使用 `log_message` 函数记录升级的结果。

返回 1 表示成功，0 表示失败。

### (3) rollback.c

`rollback.c` 包含系统回滚的具体实现。

`perform_rollback` 函数执行回滚脚本 `./scripts/rollback.sh`。

使用 `log_message` 函数记录回滚的结果。

返回 1 表示成功，0 表示失败。

#### (4) `auth.c`

`auth.c` 包含用户认证的具体实现。

`authenticate` 函数检查用户名和密码是否匹配。

如果用户名是 “admin” 且密码是 “password”，返回 1 表示认证成功，否则返回 0。

#### (5) `utils.c`

`utils.c` 包含一个简单的日志记录函数。

`log_message` 函数打印传入的消息。

## 5.脚本说明

#### (1) `create_repo.sh`

该脚本用于创建一个新的 OSTree 仓库。

检查并创建仓库目录，然后初始化仓库。

如果操作成功，输出成功信息，否则输出失败信息并退出。

#### (2) `publish_to_repo.sh`

该脚本用于将新的提交发布到 OSTree 仓库。

检查提交路径是否存在，然后将提交发布到指定分支。

如果操作成功，输出成功信息，否则输出失败信息并退出。

(2) `perform_upgrade.sh`

该脚本用于执行系统升级。

从指定的仓库和分支中拉取最新的提交。

如果操作成功，输出成功信息，否则输出失败信息并退出。

(3) `rollback.sh`

该脚本用于执行系统回滚。

回滚到之前的提交。

如果操作成功，输出成功信息，否则输出失败信息并退出。

(4) `check_failure.sh`

该脚本用于检查系统日志中的失败消息。

检查日志文件中是否包含“ERROR”关键字。

如果找到失败消息，输出失败信息并退出，否则输出成功信息并退出。

(5) `switch_partition.sh`

该脚本用于切换到不同的分区。

卸载当前分区并挂载新的分区。

如果操作成功，输出成功信息，否则输出失败信息并退出。

(6) `test_upgrade.sh`

备份升级前的系统状态。

执行升级脚本并检查其退出状态码。

通过比较升级前后的系统状态来验证升级是否成功。

清理备份文件。

#### (7) test\_rollback.sh

备份回滚前的系统状态。

执行回滚脚本并检查其退出状态码。

通过比较回滚前后的系统状态来验证回滚是否成功。

清理备份文件。

#### (8) test\_auth.sh

编译 main 程序（如果尚未编译）。

使用正确的凭据运行 main 程序，并检查输出是否包含 "Authentication successful"。

使用错误的凭据运行 main 程序，并检查输出是否包含 "Authentication failed"。

#### (9) test\_failure.sh

通过向日志文件中添加错误消息来模拟系统失败。

执行失败检查脚本并检查其退出状态码。

如果检测到错误，测试通过；否则，测试失败。

清理模拟的错误消息。

#### (10) test\_partition.sh

备份当前挂载的分区状态。

执行分区切换脚本并检查其退出状态码。

通过检查新的挂载状态来验证分区切换是否成功。

可选地，切换回原始分区。