

# x-Eris 调研报告

小组成员：胡天羽，罗胤玻，李润时，万辰希，吴书让

2023 年 3 月 30 日

## 1 项目概述

## 2 项目背景

### 2.1 嵌入式操作系统

#### 2.1.1 概述

#### 2.1.2 小结

### 2.2 非虚拟文件系统

#### 2.2.1 概述

#### 2.2.2 小结

### 2.3 虚拟文件系统 - Linux

#### 2.3.1 概述

#### 2.3.2 小结

### 2.4 虚拟文件系统 - 嵌入式

#### 2.4.1 概述

本部分主要介绍了 **FreeRTOS-Plus-FAT** 以及 **RT-Thread** 中的文件系统实现细节。

在早期的嵌入式系统中，需要存储的数据比较少，数据类型也比较单一，往往使用直接在存储设备中的指定地址写入数据的方法来存储数据。然而随着嵌入式设备功能的发展，需要存储的数据越来越多，也越来越复杂，这时仍使用旧方法来存储并管理数据就变得非常繁琐困难。因此需要新的数据管理方式来简化存储数据的组织形式，需要新的文件系统设计。

#### 2.4.2 FreeRTOS-Plus-FAT

通过阅读源代码，可以简单分类此项目的文件组织如下。

- 主要
  - **ff\_dir**: 用于访问文件夹中内容
  - **ff\_fat**: 用于访问 **FAT** 文件系统

- `ff_file`: 用于文件读写
  - `ff_ioman`: 管理缓存和挂载读写对象（介质）
  - `ff_format`: 格式化或分区介质
  - `ff_locking`: 加锁?
  - `ff_memory`: 从内存读取数据
  - `ff_stdio`: 用于文件管理（统计），相对路径转换
  - `ff_sys`: 用于映射文件系统到根目录
- 辅助
    - `ff_headers`: 管理所有头文件
    - `ff_time`: 获取时间
    - `ff_error`: 用于错误处理
    - `ff_crc`: 用于计算 CRC（循环检验码）
    - `ff_string`: 字符串库
  - 驱动
    - 各处理器相应的文件系统驱动

**ERRNO 值** FreeRTOS-Plus-FAT 文件系统的标准 API 与标准 C 库使用相同的 `errno` 值。标准 C 库中的文件相关函数返回 0 表示通过，返回 -1 则表示失败。如果返回 -1，则失败的原因存储在名为 `errno` 的变量中，须单独检查。同样，FreeRTOS-Plus-FAT 的标准 API 返回 0 表示通过，返回 -1 则表示失败，该 API 还会针对各项 RTOS 任务维护 `errno` 变量。

### 2.4.3 RT-Thread

此文件系统有较多简介，可供设计参考，它的特点是：

- 为应用程序提供统一的 POSIX 文件和目录操作接口：`read`、`write`、`poll/select` 等。
- 支持多种类型的文件系统，如 `FatFS`、`RomFS`、`DevFS` 等，并提供普通文件、设备文件、网络文件描述符的管理。
- 支持多种类型的存储设备，如 `SD Card`、`SPI Flash`、`Nand Flash` 等。

**POSIX 文件系统接口** POSIX 表示可移植操作系统接口，POSIX 标准定义了操作系统应该为应用程序提供的接口标准，是 IEEE 为要在各种 UNIX 操作系统上运行的软件而定义的一系列 API 标准的总称。

此标准意在期望获得源代码级别的软件可移植性。换句话说，为一个 POSIX 兼容的操作系统编写的程序，应该可以在任何其它 POSIX 操作系统（即使是来自另一个厂商）上编译执行。RT-Thread 支持 POSIX 标准接口，因此可以很方便的将 Linux/Unix 的程序移植到 RT-Thread 操作系统上。

**虚拟文件系统层** 用户可以将具体的文件系统注册到 DFS 中，如 `FatFS`、`RomFS`、`DevFS` 等，具体介绍请参考其他小节。

**设备抽象层** 设备抽象层将物理设备如 SD Card、SPI Flash、Nand Flash，抽象成符合文件系统能够访问的设备，例如 FAT 文件系统要求存储设备必须是块设备类型。

不同文件系统类型是独立于存储设备驱动而实现的，因此把底层存储设备的驱动接口和文件系统对接起来之后，才可以正确地使用文件系统功能。

#### 2.4.4 小结

设计 VFS 时，可以参考 RT-Thread 的三层结构组织形式、FreeRTOS-Plus-FAT 的 API 函数设计理念，将 FreeRTOS-Plus-FAT 继续拓展优化。

例如，可通过以下几点提高兼容性。（利用 POSIX 文件系统接口和与 FreeRTOS-Plus-POSIX 的结合将有可能方便地运行从 Linux/Unix 上移植的程序。虽然目前 FreeRTOS-Plus-POSIX 对 POSIX API 的支持还不够充分，但是随着开发实用价值将逐渐上升。）

- 继续采用标准 errno 值
- 添加 POSIX 文件系统接口
- 添加支持的文件系统（磁盘文件系统、闪存文件系统）
- 拓展支持的实用函数

## 2.5 文件系统开发

### 2.5.1 概述

### 2.5.2 小结

## 3 立项依据

## 4 前瞻性/重要性分析

### 4.1 ...

## 5 相关工作

### 5.1 FreeRTOS-Plus-FAT

是本项目的基础，主要使用 C 语言写成。源代码组织简洁，内容完整。但没有文档支持，建议直接阅读代码。其主要可借鉴特性：

- 采用标准 errno 值
- 项目结构组织清晰

### 5.2 RT-Thread/DFS

是 RT-Thread 的虚拟文件系统实现，文档详尽，可与上一个项目结合理解虚拟文件系统的一些设计理念。其主要可借鉴特性：

- 采用 POSIX 接口

- 目录管理相关使用函数较多
- 有关于文件系统配置方面参数的介绍

### 5.3 相关工作二

### 5.4 相关工作三

### 5.5 相关工作四

### 5.6 相关工作五

## 6 参考资料

- 1.