



Melis RTOS 存储

开发指南

版本号: 1.0
发布日期: 2020.08.22

版本历史

版本号	日期	制/修订人	内容描述
1.0	2020.08.22	AWA1379	初始化添加 Melis RTOS 存储开发方法



目 录

1 概述	1
1.1 编写目的	1
1.2 使用范围	1
1.3 相关人员	1
2 文件系统接口	2
3 文件系统配置	3
3.1 术语介绍	3
3.2 DFS 配置	3
3.2.1 littlefs 配置	4
3.2.2 ramfs 配置	4
3.2.3 devfs 配置	4
3.2.4 rootfs 配置	4
3.3 模拟 Linux 的 VFS	5
4 分区配置	6
4.1 注意事项	6
5 分区读写	7
5.1 范例	7
5.2 注意事项	7
6 存储介质开发	8
6.1 NOR Flash	8
6.2 sys_config.fex 配置	8
6.3 NOR Flash 锁保护开发	8
6.3.1 独立块锁保护	8
6.3.2 区间保护	9
6.3.3 注意事项	9
6.4 SDMMC Card	9
7 测试	11
7.1 rwcheck 校验测试	11
7.2 rwspeed 性能测试	11

1 概述

1.1 编写目的

此文档介绍从 Melis 4.0 开始，Melis RTOS 的存储开发方法。

1.2 使用范围

Allwinner 软件平台 Melis 4.0 以上版本。

1.3 相关人员

使用 Melis 4.0 以上系统的文件系统开发者、OTA 开发者

2 文件系统接口

Melis RTOS 支持 C 标准的文件读写接口 (如 fopen/fwrite/fclose)，以及 POSIX 标准的文件读写接口 (如 open/write/close)。

表 2-1: 文件接口表

接口	作用
fopen	打开文件句柄
fclose	关闭文件句柄
fread	读文件
fwrite	写文件
fseek	文件位置跳转
ftell	返回当前位置
open	打开文件句柄
close	关闭文件句柄
read	读文件
write	写文件
lseek	2GB 内文件位置跳转
lseek64	文件位置跳转
unlink	删除文件
remove	删除文件
rename	文件重命名
dup	复制文件描述符
fsync	同步文件
ftruncate	改变文件大小
ioctl	控制文件
mkdir	创建目录
rmdir	删除目录
opendir	打开目录
closedir	关闭目录
readdir	读取目录
stat	获取文件信息
statfs	获取文件系统信息
umount	卸载文件系统
mount	挂载文件系统

3 文件系统配置

3.1 术语介绍

表 3-1: 文件系统名词介绍表

接口	作用
DFS	rt-thread 精简版虚拟文件系统
littlefs	针对嵌入式设备的文件系统
ramfs	内存文件系统
devfs	rt-thread 设备文件系统
layerfs	DFS 转换层文件系统

当前 Melis RTOS 存在两套虚拟文件系统 (VFS)，一套模拟 Linux VFS，另外一套则源自 RT-Thread DFS。模拟 Linux 的 VFS 主要是为了支持 fat32, exfat, ntfs 等文件系统，适用于 TF 卡、U 盘等介质，不使用该类介质时，可以关闭，节省代码大小。采用 RT-Thread DFS 主要是为了使用 littlefs, ramfs, devfs, layerfs 等文件系统。

模拟 Linux 的 VFS 使用时，会通过 layerfs 与 DFS 进行适配。每当模拟 Linux 的 VFS 挂载一个文件系统系统，便会在 DFS 中挂载一个 layerfs 类型的文件系统。通过 layerfs，可以使用 DFS 的接口去使用 fat32, exfat, ntfs 等文件系统，以达到统一接口又兼容旧版本的目的。

3.2 DFS 配置

配置项如下：

```
make menuconfig
Kernel Setup --->
  Subsystem support --->
    RT-Thread DFS Support --->
      [*] rt-thread dfs virtual file system #使能 RT-Thread DFS
```

3.2.1 littlefs 配置

littlefs 主要针对 flash 特性而开发的文件系统，在 Melis RTOS 系统中，主要用来挂载 nor flash 的用户分区，为用户提供 flash 文件读写功能，其配置项如下：

```
make menuconfig
Kernel Setup --->
Subsystem support --->
RT-Thread DFS Support --->
[*] rt-thread dfs virtual file system #使能 RT-Thread DFS
[*] Enable Little file system # 使能 littlefs
```

3.2.2 ramfs 配置

ramfs 的挂载介质为内存块，提供临时性的文件读写功能，配置项如下：

```
make menuconfig
Kernel Setup --->
Subsystem support --->
RT-Thread DFS Support --->
[*] rt-thread dfs virtual file system #使能 RT-Thread DFS
[*] Enable RAM file system
(/tmp) Ramfs mount path (NEW) #ramfs挂载路径
(0x800000) Ramfs filesystem size (NEW) #ramfs大小
[*] Ramfs support directory (NEW) #ramfs支持目录
[*] Ramfs data slice (NEW) ramfs支持数据分片
```

3.2.3 devfs 配置

devfs 是设备文件系统，主要用于管理设备，提供使用文件系统接口操作设备的功能，配置项如下：

```
make menuconfig
Kernel Setup --->
Subsystem support --->
RT-Thread DFS Support --->
[*] rt-thread dfs virtual file system #使能 RT-Thread DFS
[*] Using devfs for device objects #使能 devfs
```

3.2.4 rootfs 配置

rootfs 为根目录挂载文件系统，其管理的数据为文件系统挂载表。通过 rootfs，可以获取当前挂载的所有文件系统，其配置项如下：

```
make menuconfig
Kernel Setup --->
    Subsystem support --->
        RT-Thread DFS Support --->
            [*] rt-thread dfs virtual file system #使能 RT-Thread DFS
            [*] Mount Melis rootfs file system for root directory
```

3.3 模拟 Linux 的 VFS

该 VFS 主要用于支持 U 盘、SD 卡等介质上的 fat,exfat, ntfs 等文件系统，其配置项如下：

```
make menuconfig
Kernel Setup --->
    File system support --->
        [*] Support melis virtual file system #使能VFS
        [*] Support devfs file system #使能对应的devfs
        [ ] Support minfs file system
        [ ] Support cdfs file system
        [ ] Support cd9660 file system
        [ ] Support exFAT file system
        [ ] Support ntfs file system
        [ ] Support fat file system
```

除此之外，如需支持原 Melis 4.0 以下的设备管理框架，还需打开以下配置：

```
make menuconfig
Kernel Setup --->
    Drivers Setup --->
        Melis Legacy Support --->
            [*] Support Melis Legacy Driver Management #使能Melis 4.0以下系统的设备管理接口
            [*] Support Melis Legacy DMS Device # 添加devfs虚拟设备
```

如需通过 DFS 接口访问 ntfs, fatfs, exfat 等文件系统，需要使能 layerfs，其配置项如下：

```
make menuconfig
Kernel Setup --->
    Subsystem support --->
        RT-Thread DFS Support --->
            [*] Mount Melis FS AS Dir in DFS (NEW) # 使能layerfs
            (/mnt/) Melis Layerfs Dir Path in DFS (NEW) # 设置挂载路径
```

4 分区配置

分区配置是通过方案目录下的 sys_partition.fex 文件进行配置。如需新增分区，可参考下列案例，添加一个表项：

```
[partition]
name      = ROOTFS      //分区名
size      = 8192        //分区大小，单位为512 byte
downloadfile = "rootfs.fex" //填充分区的文件，该文件需要放置到workspace/suniv/beetles/目录下，同时必须以.fex为文件后缀
user_type  = 0x8000
```

系统启动时，会获取系统分区信息，然后为每个分区在设备文件系统中创建一个设备节点，设备节点名格式为 “/dev/xxx”，xxx 表示在 sys_partition.fex 中配置的分区名称。通过该设备节点，开发人员可以直接读写分区。

4.1 注意事项

配置分区时，需要注意分区大小与文件系统制作工具参数相适配。

5 分区读写

对于裸分区读写，可直接通过设备节点的方式进行读写。

5.1 范例

```
int fd = open("/dev/UDISK", O_RDWR); //通过设备节点打开分区  
write(fd, data, size);  
close(fd);
```

5.2 注意事项

对于分区写，只能进行顺序写，无法进行随机写，也不可使用 lseek 进行跳转写入。

6 存储介质开发

6.1 NOR Flash

NOR Flash 驱动核心代码位于 `ekernel/drivers/hal/source/spinor/hal_spinor.c` 中，如需增加对新物料的支持，需要在 `nor_ids` 数组中新增物料信息，如物料名称，jedec id，擦除块大小，容量等。

6.2 sys_config.fex 配置

暂不支持根据物料信息来适配物料读写命令模式，当前仅支持通过 `sys_config.fex` 设置 NOR Flash 读写模式和频率。找到对应方案的 `sys_config.fex`，按下列模板进行修改：

```
# spinor_para配置项获取spinor信息
[spinor_para]
;read_mode: when read_mode=4 use 4 quad mode;equal 2 use dual mode;equal 1 use single fast
    mode
;write_mode: when write_mode=4 use 4 quad mode;equal 2 use dual mode;equal 1 use single
    fast mode
;frequency: frequency

write_mode = 4 # 配置写模式
read_mode = 4 # 配置读模式
spi_port = 0 # 使用的spi控制器
frequency = 100000000 # 配置spi时钟频率
```

6.3 NOR Flash 锁保护开发

6.3.1 独立块锁保护

宏`CONFIG_SUPPORT_INDIVIDUAL_BLOCK_MEMORY_PROTECTION`用于配置 NOR Flash 独立块锁保护，如该款物料支持独立块锁保护，请按下列步骤进行开发。

1. 在 `nor_factory_init` 函数中，对接 Flash 芯片的锁保护配置函数，配置为独立块锁保护。
2. 在 `nor_blk_lock` 和 `nor_blk_unlock` 函数中，对接加锁和解锁功能。
3. 在 `nor_ids` 中 Flash 芯片对应的表项中，在 `flag` 中添加 `EN_INDIVIDUAL_PROTECT_MODE` 信息。

6.3.2 区间保护

宏 `CONFIG_SUPPORT_STATUS_REGISTER_MEMORY_PROTECTION` 用于配置 NOR Flash 区间锁保护，如该款物料不支持独立块锁保护，仅支持区间锁保护，请按下列步骤进行开发。

1. 在 `nor_factory_init` 函数中，对接 Flash 芯片的锁保护配置函数，配置为区间锁保护。
2. 在 `nor_bp_unprotect` 里面，对接加锁和解锁功能。
3. 根据 Flash 文档，建立其保护区间数组。
4. 在 `nor_ids` 中 Flash 芯片对应的表项中，设置 `pt` 和 `pt_len` 的值为保护区间数组及其数组长度。

6.3.3 注意事项

对于新款 Nor Flash 物料的锁保护开发，除了需要在 Melis RTOS 系统的 Flash 驱动进行适配，还需要在 bootloader 中进行适配，否则将无法正常烧录和启动。在 bootloader 中的适配，请联系全志工程师予以协助。

6.4 SDMMC Card

SDMMC Card 使能配置如下：

```
# 选上SDMMC Card驱动
make menuconfig
    Kernel Setup --->
        Drivers Setup --->
            Melis Source Support --->
                [*] Support SDMMC Driver # 使能SDMMC Card驱动

# SDMMC Card设备注册方式较为特殊，需打开以下配置
make menuconfig
    Kernel Setup --->
        Drivers Setup --->
            Melis Legacy Support --->
                [*] Support Melis Legacy Driver Management #使能Melis 4.0以下系统的设备管理接口
                [*]     Support Melis Legacy DMS Device # 添加devfs虚拟设备

# 选上VFS支持，如需挂载fat或者ntfs等文件系统，请自行选上
make menuconfig
    Kernel Setup --->
        File system support --->
            [*] Support melis virtual file system #使能VFS
            [*] Support devfs file system #使能对应的devfs
```

除此之外，提供了 SDMMC Card 的缓存功能，其实现了预读、读写相邻块合并、数据缓存等功能，配置项如下：

```
make menuconfig
Kernel Setup --->
  Drivers Setup --->
    Melis Source Support --->
      [*] support sdmmc cache # 使能SDMMC数据缓存
      (1024) sdmmc cache size (KB) (NEW) #设置数据缓存大小
      (14)  sdmmc cache writeback thread priority (NEW) #缓存数据回刷任务的优先级
      [ ]   support sdmmc cache information command (NEW) # 数据缓存调试命令
```



7 测试

7.1 rwcheck 校验测试

rwcheck 命令用于测试文件系统、存储驱动读写数据的正确性。配置方法：

```
make menuconfig
  Kernel Setup  --->
    Subsystem support  --->
      Command shell  --->
        Commands  --->
          Iobox Command  --->
            [*] Rwcheck command
```



其使用方法如下：

```
rwcheck <-d dir> [-h] [-t times] [-s size] [-p percent] [-m jobs]
-d : 测试目录
-h : 帮助信息
-t : 测试次数
-s : 测试文件大小
-p : 测试数据占文件系统百分比
-m : 测试任务个数
```

7.2 rwspeed 性能测试

rwspeed 命令用于测试文件系统、存储驱动读写数据的性能。配置方法如下：

```
make menuconfig
  Kernel Setup  --->
    Subsystem support  --->
      Command shell  --->
        Commands  --->
          Iobox Command  --->
            [*] Rwspeed command
```

其使用方法如下：

```
rwspeed <-d dir> [-h] [-t avg] [-s size]
-d : 测试目录
-h : 帮助信息
-t : 测试次数
-s : 测试文件大小
-b : 每笔读写数据大小
```

-m : 测试任务个数



著作权声明

版权所有 © 2021 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明

  **全志科技**  (不完全列举) 均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。