nuttx文件系统

标签（空格分隔）： 未分类

1.数据结构

```
/* This structure represents one inode in the Nuttx
pseudo-file system */

struct inode
{
  FAR struct inode *i_peer;       /* Pointer to same
level inode */
  FAR struct inode *i_child;      /* Pointer to lower
level inode */
  int16_t           i_crefs;      /* References to inode
*/
  uint16_t          i_flags;      /* Flags for inode   节点
类型是驱动还是块设备还是挂载点*/
  union inode_ops_u u;            /* Inode operations */
#ifdef CONFIG_FILE_MODE
  mode_t            i_mode;       /* Access mode flags
*/
#endif
  FAR void         *i_private;    /* Per inode driver
private data */
  char              i_name[1];    /* Name of inode
(variable) */
};
```

这个结构体代表了nuttx文件系统的文件树(二叉树算法)的一个节点,也就是磁盘上的一个文件对应一个inode,里面包含了各种文件操作:

```
union inode_ops_u
{
  FAR const struct file_operations    *i_ops;  /* Driver
operations for inode */
#ifndef CONFIG_DISABLE_MOUNTPOUNT
  FAR const struct block_operations   *i_bops; /* Block
driver operations */
  FAR const struct mountpt_operations *i_mops; /*
Operations on a mountpoint */
#endif
};
```

其中 i_ops是驱动节点的文件操作

```
i_bops是块设备节点的文件操作
i_mops是挂载点也就是外部存储设备,如sd卡的文件系统相应的文件操作
实际上i_bops都是i_mops中的文件操作用于最终操作硬件的块设备的方法,
如write会先调用i_mops->write,再调用i_bops->write
```

注意,**inode_ops_u**是共同体,一个**inode**只能是**ops/bops/mops**的其中一个,并且**ops**的前**6**个成员**(open,read...)**和**mops**一样,所以系统调用中统一用**ops.** 对于挂载点,相应的**bops**在**mount**时指定

```
struct file
{
  int               f_oflags; /* Open mode flags */
  off_t             f_pos;    /* File position */
  FAR struct inode *f_inode;  /* Driver interface */
  void             *f_priv;   /* Per file driver private
data */
};
```

应用层调用open以后会alloc一个这个结构体,用于描述当前打开的文件,里面有这个文件对应的inode结构体.
查看files_allocate函数,可以知道,其实alloc的方式是结构体

```
struct filelist
{
  sem_t   fl_sem;             /* Manage access to the
file list */
  struct file fl_files[CONFIG_NFILE_DESCRIPTORS];
};
```

中有个file结构体的数据,里面谁的inode指针为空,就代表当前这个file结构体可用,然后返回数组成员号作为fd

2.romfs文件系统初始化

```
os_start->os_bringup->CONFIG_USER_ENTRYPOINT->nsh_main-
>nsh_initialize->nsh_romfsetc->mount
```

px4的nuttx系统中,romfs是挂载到/etc目录下的,里面就是init.d目录

3.SD卡挂载

启动脚本init.d/rcS中调用mount命令

```
mount -t vfat /dev/mmcsd0 /fs/microsd
```

驱动节点所在的/dev不需要mount,似乎在启动是调用devnull_register时创建/dev目录