```
#ifndef _EXT2_FUNC_H
#define _EXT2_FUNC_H

#define VOLUME_NAME     "EXT2FS"    // 卷名
#define BLOCK_SIZE 512          // 块大小
#define DISK_SIZE       4612        //磁盘总块数

#define DISK_START 0            // 磁盘开始地址
#define SB_SIZE 128                 //超级块大小是128

#define GD_SIZE 32                  // 块组描述符大小是32B
#define GDT_START       (0+512)     // 块组描述符起始地址

#define BLOCK_BITMAP (512+512) // 块位图起始地址
#define INODE_BITMAP (1024+512)// inode 位图起始地址

#define INODE_TABLE (1536+512) // 索引节点表起始地址 4*512
#define INODE_SIZE 64           // 每个inode的大小是64B
#define INODE_TABLE_COUNTS      4096 // inode entry 数

#define DATA_BLOCK (263680+512) // 数据块起始地址 4*512+4096*64
#define DATA_BLOCK_COUNTS       4096    // 数据块数

#define BLOCKS_PER_GROUP        4612 // 每组中的块数

#define USER_NAME "root" //用户名
#define PASSWORD "123456" //密码

struct super_block // 32 B
{
        char sb_volume_name[16]; //文件系统名
        unsigned short sb_disk_size; //磁盘总大小
        unsigned short sb_blocks_per_group; // 每组中的块数
        unsigned short sb_size_per_block;       // 块大小
    char username[10]; //用户名
        char password[10]; //密码
};

struct group_desc // 32 B
{
    char bg_volume_name[16]; //文件系统名
    unsigned short bg_block_bitmap; //块位图的起始块号
```

```
    unsigned short bg_inode_bitmap; //索引结点位图的起始块号
    unsigned short bg_inode_table; //索引结点表的起始块号
    unsigned short bg_free_blocks_count; //本组空闲块的个数
    unsigned short bg_free_inodes_count; //本组空闲索引结点的个数
    unsigned short bg_used_dirs_count; //组中分配给目录的结点数
    char bg_pad[4]; //填充(0xff)
};
struct inode // 64 B
{
    unsigned short i_mode;   //文件类型及访问权限
    unsigned short i_blocks; //文件所占的数据块个数
    unsigned int i_size;    // 文件或目录大小(单位 byte)
    unsigned long i_atime;   //访问时间
    unsigned long i_ctime;   //创建时间
    unsigned long i_mtime;   //修改时间
    unsigned long i_dtime;   //删除时间
    unsigned short i_block[8]; //直接索引方式 指向数据块号
    char i_pad[8];           //填充(0xff)
};
struct dir_entry //16B
{
    unsigned short inode; //索引节点号
    unsigned short rec_len; //目录项长度
    unsigned short name_len; //文件名长度
    char file_type; //文件类型(1 普通文件 2 目录.. )
    char name[9]; //文件名
};


static unsigned short last_alloc_inode; // 最近分配的节点号 */
static unsigned short last_alloc_block; // 最近分配的数据块号 */
static unsigned short current_dir;   // 当前目录的节点号 */

static unsigned short current_dirlen; // 当前路径长度 */

static short fopen_table[16]; // 文件打开表 */

static struct super_block sb_block[1];  // 超级块缓冲区
static struct group_desc gdt[1];        // 组描述符缓冲区
static struct inode inode_area[1];  // inode缓冲区
static unsigned char bitbuf[512]={0}; // 位图缓冲区
static unsigned char ibuf[512]={0};
static struct dir_entry dir[32];   // 目录项缓冲区 32*16=512
```

```
static char Buffer[512];  // 针对数据块的缓冲区
static char tempbuf[2*1024*1024];        // 文件写入缓冲区
static FILE *fp;         // 虚拟磁盘指针


extern char current_path[256];    // 当前路径名 */

static unsigned long getCurrentTime();//得到当前时间
static void update_super_block(void);   // 更新超级块内容
static void reload_super_block(void);   //加载超级块内容
static void update_group_desc(void);    //更新组描述符内容
static void reload_group_desc(void);    //加载组描述符内容
static void update_inode_entry(unsigned short i); //更新indoe表
static void reload_inode_entry(unsigned short i); //加载inode表
static void update_block_bitmap(void);  //更新块位图
static void reload_block_bitmap(void);  //加载块位图
static void update_inode_bitmap(void);  //更新inode位图
static void reload_inode_bitmap(void);  //加载inode位图
static void update_dir(unsigned short i);//更新目录
static void reload_dir(unsigned short i);//加载目录
static void update_block(unsigned short i);//更新数据块
static void reload_block(unsigned short i);//加载数据库
static int alloc_block(void);//分配数据块
static int get_inode(void); //得到inode节点
static unsigned short reserch_file(char tmp[9],int file_type,unsigned short *inode_num,unsigned
static void dir_prepare(unsigned short tmp,unsigned short len,int type);
static void remove_block(unsigned short del_num);//删除数据块
static void remove_inode(unsigned short del_num);//删除inode节点
static unsigned short search_file(unsigned short Ino);//在打开文件表中查找是否已打开文件
static void sleep(int k);
static void initialize_disk(void);//初始化磁盘


#endif
```