

```
#ifndef TYPE_H
#define TYPE_H

//this is the header file with all of the definitions for EXT2 file system stuff
//simply copy and pasted from textbook
#include <stdio.h>
#include <stdlib.h>

#include <fcntl.h>
#include <ext2fs/ext2_fs.h>
#include <libgen.h>
#include <string.h>
#include <sys/stat.h>
#include <time.h>
#include <math.h>

typedef unsigned char  u8;
typedef unsigned short u16;
typedef unsigned int   u32;

typedef struct ext2_group_desc  GD;
typedef struct ext2_super_block SUPER;
typedef struct ext2_inode       INODE;
typedef struct ext2_dir_entry_2 DIR;

GD      *gp;
SUPER   *sp;
INODE    *ip;
DIR      *dp;

#define BLOCK_SIZE      1024
#define BLKSIZE         1024

#define BITS_PER_BLOCK  (8*BLOCK_SIZE)
#define INODES_PER_BLOCK (BLOCK_SIZE/sizeof(INODE))

#define SUPERBLOCK      1
#define GDBLOCK         2
#define ROOT_INODE      2

#define BBITMAP          3
#define IBITMAP          4
#define INODEBLOCK       5
#define INODE_START_POS 10

#define DIR_MODE         0040777
#define FILE_MODE        0100644
#define SUPER_MAGIC      0xEF53
#define SUPER_USER       0

#define FREE             0
#define BUSY             1
#define READY           2
#define KILLED          3

#define NMINODE          100
#define NMOUNT           10
#define NPROC            10
#define NFD              10
#define NOFT             50
```

```

typedef struct Oft{
    int mode;
    int refCount;
    struct Minode *inodeptr;
    long offset;
} OFT;

/*
inodeptr is the inode struct from block
everything else if info for the filesystem about the inode

fd[ ] is an array of OFT structs (not necessarily open)
*/

typedef struct Proc{
    int uid;
    int pid;
    int gid;
    int ppid;
    int status;

    struct Minode *cwd;
    OFT *fd[NFD];

    struct Proc *next;
    struct Proc *parent;
    struct Proc *child;
    struct Proc *sibling;
} PROC;

/*
In Minode, the INODE INODE is the inode info
Everything else is info for the filesystem about the inode
*/

typedef struct Minode{
    INODE INODE;
    int dev, ino;

    int refCount;
    int dirty;
    int mounted;
    struct Mount *mountptr;
    char name[128];
} MINODE;

typedef struct Mount{
    int ninodes;
    int nblocks;
    int dev, busy;
    struct Minode *mounted_inode;
    char name[256];
    char mount_name[64];
} MOUNT;

//globals
MINODE minode[NMINODE];
MINODE *root;
PROC proc[NPROC], *running;
MOUNT mounttab[5];

```

```
char names[64][128],*name[64];
char third [64];
OFT OpenFileTable[NOFT];
int fd, dev, n;
int nblocks, ninodes, bmap, imap, inode_start, iblock;
int inodeBeginBlock;
char pathname[256], parameter[256];
char teststr[1024] = "";
int DEBUG;

#endif
```