**四、文件管理**

**4.1目的**

用磁盘文件实现操作系统的文件管理功能，主要包括目录结构的管理、外存空间的分配与释放以及空闲空间管理三部分。

**4.2内容**

模拟windows的DOS命令，实现md、mk、cd、rd、dir、tree、cd ..、del、help、cls、exit命令。

**4.3数据结构**

#define beginsize 5

#define LENGTH 3

typedef struct // 一次间址

{

int data[LENGTH];

}Indireone;

typedef struct //二次间址

{

Indireone \* first[LENGTH];

}Indiretwo;

typedef struct //三次间址

{

Indiretwo \* second[LENGTH];

}Indirethree;

typedef struct Node

{

int begin[beginsize]; //直接地址

Indireone \* one;

Indiretwo \* two;

Indirethree \* three;

}Mixtab; //混合索引节点

typedef struct NODE

{

char name[50];

int type;//是文件还是目录,1代表文件，0代表目录

int size;//如果是文件给出大小

struct NODE \*next;//兄弟结点

struct NODE \* sub;//子节点

struct NODE \* father;//父节点

Mixtab \* table;

}FCB;

//文件控制块

FCB \* root;

FCB \* present;

FCB \* finding;

char stringname[300];

int Bitmap[16][16];//位示图

int leftbit=0; // 空的块数

**4.4算法设计及流程图**

初始化root节点，产生位示图，记录位示图里空闲块数。

创建目录时先判断其是否存在，若存在则提示错误信息，若不存在则创建FCB，初始化FCB把它添加到当前目录的子节点下。

创建文件时先判断其是否存在，若存在则提示错误信息；接下来判断剩余空闲块数是否足够，若不够提示错误信息；接下来创建FCB并初始化，若文件初始长度不为空，那么创建混合索引节点，并把从空闲区占用的块号存入索引节点。

改变路径时判断当前是否在root目录，若在root里，那执行cd .. 操作将不做反应。若做其他跳转，先判断跳转的目录是否存在，若不存在给出错误提示，若存在则进行跳转，更新present的值；若不在root里,大体同上，只是在cd ..处理上，这种情况下可以进行向前跳转，并且cd \\ 可以直接跳回root目录。

删除目录时先判断目录是否存在，不存在给出错误提示；若存在那么判断它的子节点是否为空，不为空给出错误提示，为空的话删除此目录。

删除文件时先判断文件是否存在，不存在给出错误提示。若存在去找索引节点，依次归还占用的位示图的块号。

列出目录内容时通过present节点，先获取时间，接下来若是目录的话，列出目录名，若是文件的话，列出文件大小、名字。最后列出汇总结果。

树形目录显示首先判断它是否有子节点，若没有给出错误提示；若有则进行深搜显示。



**4.5小结**

树形显示时遇到很多问题，格式控制等细节处理很繁琐。混合索引的处理实现的也有些麻烦，写了很多重复的代码，若将其优化成函数调用，可能会节省代码量。