数据结构 算法流程

3. 程序中所用的数据结构及说明

3.1 文件结构体 (File)

3.2 空闲块组结构体 (BlockGroup)

3.3 超级块结构体 (SuperBlock)

4. 算法及流程

4.1 初始化 (initSuperBlock)

4.2 分配块 (allocateBlocks)

4.3 释放块 (releaseBlocks)

4.4 添加文件 (addFile)

4.5 删除文件 (deleteFile)

4.6 列出所有文件 (listFiles)

4.7 主函数 (main)

3. 程序中所用的数据结构及说明

程序使用了几个关键的数据结构来模拟一个简单的文件系统，包括超级块（SuperBlock）、空闲块组（BlockGroup）和文件（File）。以下是每个数据结构的详细说明：

3.1 文件结构体（`File`）

```
typedef struct {
    char name[256];          // 文件名的最大长度为255字符加一个终止符
    int firstBlock;          // 文件的第一个块号
    int blockCount;          // 文件占用的块数
} File;
```

- `name`：存储文件的名称。使用数组 `char` 来保存字符串，并预留足够空间以确保不会发生溢出。
- `firstBlock`：表示该文件在磁盘上的起始块号。
- `blockCount`：记录该文件占用了多少个物理块。

3.2 空闲块组结构体（`BlockGroup`）

```c
typedef struct BlockGroup {
    int blocks[GROUP_SIZE];  // 存储本组空闲块号的数组
    int count;               // 当前组内剩余空闲块数量
} BlockGroup;
```

- `blocks`：一个固定大小的数组，用于存储当前组内的所有空闲块号。每个元素代表一个可用的块号。
- `count`：表示当前组内还有多少个空闲块未被分配出去。

3.3 超级块结构体（`SuperBlock`）

```c
typedef struct {
    BlockGroup *groups;      // 动态分配的空闲块链数组指针
    int currentGroup;        // 当前可用的最后一组索引
    int groupCount;          // 总组数
} SuperBlock;
```

- `groups`：指向一组 `BlockGroup` 类型的数组，动态分配内存来存储所有的空闲块信息。
- `currentGroup`：指示当前正在使用的最后一个包含空闲块的组的索引位置。
- `groupCount`：记录总共创建了多少个空闲块组。

4. 算法及流程

4.1 初始化（`initSuperBlock`）

- **输入参数**：磁盘大小、空闲块总数和每组空闲块的数量。
- **逻辑**：
  - 根据磁盘大小和每组空闲块的数量计算需要多少个空闲块组。
  - 动态分配内存给这些空闲块组，并初始化它们的 `blocks` 数组和 `count` 成员。
  - 将每个空闲块组的信息写入对应的文本文件中，模拟实际磁盘操作。

4.2 分配块（`allocateBlocks`）

- **输入参数**：请求分配的块数。
- **逻辑**：
  - 遍历超级块中的空闲块组，寻找可以满足请求的组。
  - 如果找到合适的组，则更新该组的 `blocks` 数组和 `count` 成员，返回分配的起始块号。

- 如果没有足够的空闲块，返回-1表示分配失败。

4.3 释放块（`releaseBlocks`）

- **输入参数**：要释放的起始块号和块数。
- **逻辑**：
  - 计算要释放的块属于哪个空闲块组。
  - 更新该组的 `blocks` 数组和 `count` 成员，将释放的块重新加入到空闲列表中。
  - 更新相应的文本文件，模拟实际磁盘操作。

4.4 添加文件（`addFile`）

- **逻辑**：
  - 检查文件系统是否已满。
  - 获取用户输入的新文件名和所需块数。
  - 调用 `allocateBlocks` 尝试分配指定数量的块。
  - 如果分配成功，更新文件列表并打印当前超级块状态；否则提示用户无法创建文件。

4.5 删除文件（`deleteFile`）

- **逻辑**：
  - 获取用户输入的要删除的文件名。
  - 查找匹配的文件，调用 `releaseBlocks` 释放其占用的块。
  - 更新文件列表并打印当前超级块状态；如果找不到文件，则提示用户文件不存在。

4.6 列出所有文件（`listFiles`）

- **逻辑**：
  - 遍历文件列表，打印每个文件的名字、首块号以及它所占用的所有物理块号。

4.7 主函数（`main`）

- **逻辑**：
  - 提供命令行界面，允许用户选择不同的操作（创建文件、删除文件、列出所有文件或退出）。
  - 根据用户的选择调用相应的函数。
  - 在用户选择退出时，输出统计信息（如分配的物理块总数和读磁盘次数），然后清理资源并结束程序。

通过上述算法和流程，程序实现了对文件系统的简单模拟，包括文件的创建、删除和查询功能，同时管理着磁盘上空闲块的分配与回收。