File Manager 设计报告 

> by 李易非

一、设计简介

由于 Record Manager 与 Catalog Manager 都要对数据文件进行操作，我们在 Record Manager / Catalog Manager 与 Buffer Manager 之间加入 File Manager，理由如下：

- 减少各个 Manager 交互；
- 逻辑意义上更加清晰，Record Manager 与 Catalog Manager 不必亲自提供 / 调用各个文件的块位置，逻辑上只对对应 " 文件 " 进行操作。

  File Manager 具有的功能如下 (简化版) ：

```
class FileManager
{
    private:
    string file_name;
    int record_length;           // 记录长度
    int first_free_record_addr; // free list 表头
    int record_count;            // 记录总数
    int pointer;                 // 文件中的“记录指针”

    public:

    FileManager(文件名);

    const char* get_record(记录地址); // 通过地址获得记录内容

    const int add_record(记录的rawdata); // 添加一条记录

    bool delete_record_ByAddr(记录地址); // 通过地址删除记录

    int getNextRecord(char * rawdata);  // 获得下一条record


};
```

二、具体说明

1、成员变量说明

- `file_name`: file_name 为该 FileManager 正在接管的文件；
- `Record_length`: record_length 为 FileManager 管理的文件的记录长度；
- `First_free_record_addr` 为 free_list 链表头；
- `Record_count` 为文件中记录总数；
- `pointer` 为文件内部指针，指向一条记录；

2、成员函数说明

- `FileManager(文件名)`
  - 通过文件名打开一个文件，读取第一个 Block 的 Metadata信息，存入成员变量中；
  - Pointer 指向第一条记录的“前一条记录”，因为 `getNextRecord` 的函数需要我们不断获得下一条记录，所以 Pointer 的初始值是第一条记录的“前一条记录”。
- `const char* get_record(记录地址);`
  - 先判断记录地址是否大于 **EOF**，若超出文件范围，弹出错误；
  - 通过记录地址与记录长度，判断记录存在的块，以及记录在块中的相对位置；
  - 调用 Buffer Manager 获得记录内容并返回。

```
if 记录地址 > EOF
    throw err

Block ID = 记录地址 / 4096;
相对位置 = 记录地址 - Block ID * 4096;

Block = BufferManager.GetBlock( Block ID );
return Block.( content + 相对位置 )
```

- `const int add_record(记录的rawdata);`
  - 若当前 free_list 为空，则插入文件末尾；
  - 若当前 free_list 不为空，则插入 free_list 所指向的地址，并更新 free_list
  - 插入完毕后，更新文件 Meta Data (第一个 Block)

```
if first_free_record_addr = -1
    Add to file EOF;
else
    Add by free_list;

Update Meta Data;
```

- `bool delete_record_ByAddr(记录地址);`
  - 先判断记录地址是否大于 **EOF**，若超出文件范围，弹出错误；
  - 通过记录地址与记录长度，判断记录存在的块，以及记录在块中的相对位置；
  - 调用 Buffer Manager 获得记录内容，重置记录内容的前四位，将其指向 free_list_head 指向的地址，free_list_head 指向本次调用所需删除的记录；
  - Block 标记 dirty;

- 更新文件 Meta Data (第一个 Block)

```
if 记录地址 > EOF
    throw err

Block ID = 记录地址 / 4096;
相对位置 = 记录地址 - Block ID * 4096;

Block = BufferManager.GetBlock( Block ID );

Block.Content 前四位 = fisrt_free_record_addr;
first_free_record_addr = 记录地址;
Block.setDirty();

Update Meta Data;
return true;
```

- `int getNextRecord(char * rawdata)`
  - 首先将 `pointer` 自增，判断是否 **EOF** ，若是，则返回 -1；
  - 判断 `pointer` 所指的记录是否是有效记录 (判断句尾有效位)；
  - 一直进行上述过程，直到找到一个有效记录，将有效记录赋给 `rawdata`；

```
PointerIncrement();

while(记录无效)
    PointerIncrement();
    if(Pointer > EOF)
        return -1;

rawdata = Pointer.Content;
return Pointer;
```