**File Processing**

**Programming Project #2**

**컴퓨터공학과**

**20141602**

**황기덕**

1. **프로젝트 목적**

* 이전 프로젝트에서 구현하였던 뉴스 구독 서비스의 회원 관리를 위해 회원 정보와 뉴스 구독 내역에 관한 정보를 처리하고 유지하는 정보 시스템에 관리자 모드 및 인덱스 기능을 추가한다.

1. **프로젝트 기본 가정**

* 1차 프로젝트때 가정했던 모든 조건들을 만족한다.
* 처음 프로그램을 실행하고 7번을 누르면, fileOfMember.dat, fileOfNewsAgency.dat 파일을 기준으로 index 파일을 생성한다. 이 때, fileOfMember.dat 에는 test를 위한 admin과 TestUser의 정보가 미리 담겨있다고 가정한다.
* 이 프로젝트는 visual studio 2017. window sdk 버전 10.0.15063에서 진행하였다.
* 모든 인덱스가 메모리에 저장될 수 있다고 가정한다.
* NewsAgencySubscriptionSystem이 시작 될 때, 각각의 인덱스 파일의 존재 여부를 체크하여인덱스 파일이 존재하지 않는다면 각각의 레코드 파일에 대한 인덱스를 생성하도록 한다. 초기에는 레코드 파일만 존재한다고 가정한다.

1. **method, field 추가구현 / 추가된 파일**

* **Member**

Member class에서 이번 프로젝트를 위하여 추가한 field와 method이다.

|  |  |  |
| --- | --- | --- |
| Data | Type | update |
| key | Variable string | Key |
| level | 2 characters (Numberic) | Get\_level |

Get\_password => instance 의 password를 반환한다.

Get\_level => instance의 level을 반환한다.

Key => primary key인 id를 이용해 key를 만들어 index file에서 사용 가능한 key를 만들어준다.

* **NewsAgency**

NewsAgency class에서 이번 프로젝트를 위해 추가한 field와 method이다.

|  |  |  |
| --- | --- | --- |
| Data | Type | update |
| Key | Variable string | Key |

Key => primary key인 NewsAgency Id를 이용하여 index file에서 사용 가능한 key를 만들어준다.

**추가된 파일**

생성자 / 소멸자 제외.

* **TextIndex**

**Field**

|  |  |  |
| --- | --- | --- |
| **Data** | **Type** | **Function** |
| MaxKeys | Int | 할당한 index 개수.(default = 100) |
| NumKeys | Int | 현재 저장된 index의 개수 |
| Keys | Char\*\* | 현재 저장된 key들의 배열 |
| RecAddrs | Int\* | 현재 저장된 key가 가리키는 address |
| Unique | int | 현재 key가 unique하다는 것을 보장하는 flag |

**Method**

|  |  |  |
| --- | --- | --- |
| **Method** | **Type** | **Function** |
| Insert | Int | Parameter로 받은 key, recaddr을 통해 index 삽입 |
| Remove | Int | Parameter로 받은 key를 index에서 삭제하고 남은 key, recaddr 배열을 앞으로 모음. |
| Search | Int | Parameter로 받은 key의 recaddr를 return |
| Print | Void | 그 index의 정보와 내용을 print |
| Find | Int | Parameter로 받은 key의 index number를 return |
| Init | int | Text index 초기화 |
| FirstRecAddr | int | 가장 작은 키에 대한 참조를 return |
| NextRecAddr | int | 다음 키에 대한 참조를 return |

**Textind.h/ textind.cpp**

* 메모리에서 index를 관리하는 클래스로, 생성자 메소드 textindex()에 정수 값을 넣으면 그만큼을 저장할 수 있는 index 배열이 생성된다. 만약 넣지 않으면 max값은 자동으로 100으로 설정이 된다(추후 수정 예정). Insert method를 사용하여 key와 address를 저장할 수 있으며, 다른 method들에 기본적으로 사용되고있는 Find() 함수에선 binary search를 사용하여 수행 시간을 줄였다.
* **TextIndexBuffer**

**Field**

|  |  |  |
| --- | --- | --- |
| **Data** | **Type** | **Function** |
| maxKeys | Int | Index key의 최대 개수 |
| KeySize | Int | Index key의 size |
| Dummy | Char\* | 임시 가용 공간 |

**Method**

|  |  |  |
| --- | --- | --- |
| **Method** | **Type** | **Function** |
| Pack | Int | Index/ recaddress를 pack한다. |
| Unpack | Int | Index / rec address를 unpack한다. |
| Print | void | 현재 index buffer를 출력한다. |

**tindbuff.h / tindbuff.cpp**

* FixedFieldBuffer를 상속받아 구현되었다. 주로 구현한 text index에 대해 pack, unpack 연산을 지원하여 실제로 ind 파일에 저장할 수 있게끔 도와주는 class이다. 생성될 때 keysize와 maxkeys 값을 받아 fixedfieldbuffer에 그것들이 들어갈만한 공간을 만들어준다.
* **TextIndexFile**

**Field**

|  |  |  |
| --- | --- | --- |
| **Data** | **Type** | **Function** |
| Index | TextIndex | Index를 저장하고 있다. |
| IndexFile | BufferFile | 저장할 file을 저장한다. |
| IndexBuffer | TextIndexBuffer | Index를 pack/unpack 하기 위한 buffer. |
| DataFile | RecordFile<RecType> | 실제 data를 가지고 있는 datafile이다. |
| FileName | Char \* | 해당 파일의 기본 이름(ex: fileOfMember)를 저장한다. |

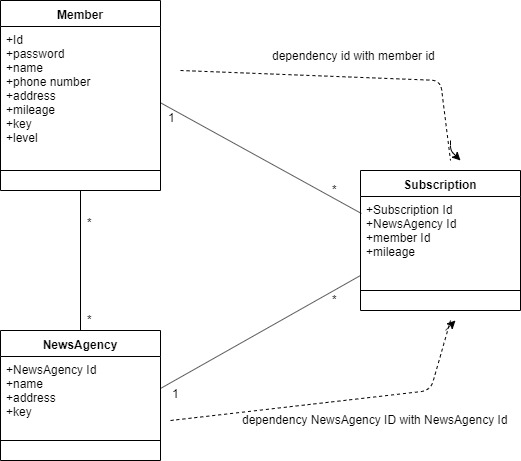
**Method**

|  |  |  |
| --- | --- | --- |
| **Method** | **Type** | **Function** |
| Read | Int | 다음 record를 읽거나, parameter로 넘겨받은 key에 대한 record를 찾아서 읽는다. |
| Append | Int | Record에 대한 것을 index 가장 끝에 삽입 |
| Create | Int | 해당 파일의 기본 이름을 통해 맞는 파일을 생성한다. |
| Open | Int | Parameter로 받은 filename에 해당하는 filestream을 open한다. |
| Close | Int | Open된 stream을 닫는다. |
| Init | Void | Textindexedfile을 초기화한다. |
| Delete | Int | Parameter로 받은 record를 file에서 삭제한다. |
| SetFileName | Int | 해당 파일의 기본 이름을 통해 맞는 이름을 설정. |

**Indfile.h**

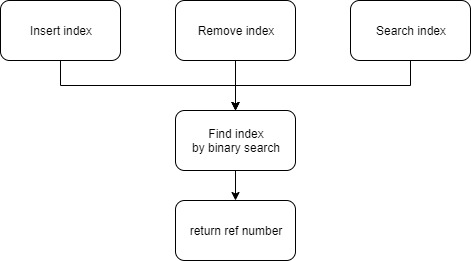
* 데이터 파일과 그에 해당하는 index 파일을 같이 관리하는 file이다. index에서 key를 통해 ref를 읽어와 datafile에서 실제 record를 읽고, index와 datafile 뒤에 실제로 record와 index를 추가하는 등 datafile과 indexfile 을 동시에 handling한다. 이 때 .dat , .ind 파일로 형식이 맞춰져 있어야 한다.

1. **클래스 다이어그램**

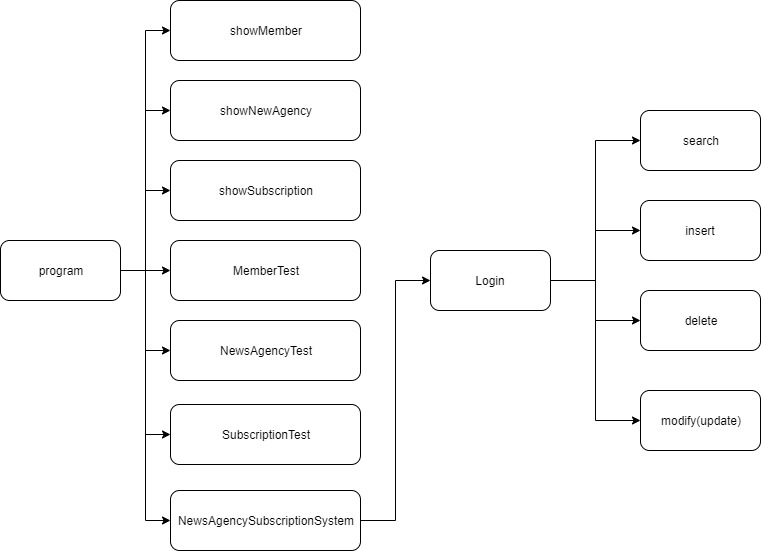


1. **Flow chart**

**find index**



**전체 structure**



1. **추가된 함수 설명**

* **memLogin**

저번 프로젝트완 달리 user mode와 admin mode를 나누어 두었기 때문에 NewsAgencySubscriptionSystem을 실행할 때 login을 하여 level에 따라 user mode와 admin mode를 나누어 주었다. 이 때 get\_id, get\_password, get\_level등을 통하여 id와 passsword가 맞는지, level은 어떠한지를 확인해 주었고 level 별로 할 수 있는 연산 (추가, 삭제) 들을 다르게 바꾸어 주었다.

* **MakeMemberIdx/ MakeAgencyIdx**

이번 프로젝트에선 index를 사용하였다. 그러므로 index file을 만들기 위해 추가하였고, NewsAgencySubscriptionSystem이 실행될 때 처음에 한 번 index file을 기본적으로 만들어지게끔 해 주었다.

**- FirstRecAddr / NextRecAddr , Find, Delete, Init**

->7장 연습문제

1. **7장 연습문제**

* **13**

**키 순서로 인덱스를 통한 반복을 지원하기 위해 TextIndex 클래스에 메소드를 첨가하라. 한가지 가능한 전략은 두 개의 메소드를 정의하는 것이다.**

Key 순서로 인덱스를 반복해야 하기 때문에, firstrecaddr / nextrecaddr 함수를 추가해 주었다. 이를 통해 첫번째 index의 시작 recAddr, 현재 가리키고 있는 index의 그 다음 index의 recAddr을 return시켜 각각의 recAddr을 얻을 수 있었다.

* **16**

**키 배열에서 이진 탐색을 실행하도록 TextIndex::Search 메소드를 수정하라.**

TextIndex에서는 insert, search, remove에서 find라는 함수를 사용하여 index를 검색한다. 이 때 이 find 함수를 순차 검색에서 binary search를 사용하게끔 변경해 주었다.

* **17**

**클래스 TextIndexedFile의 삭제 메소드를 구현하라.**

Delete method를 추가해 구현하였다. 현재 유지하고 있는 Index에서 삭제하려는 record를 remove method를 이용해 뺀 다음, indexfile을 다시 새로 열어서 빠진 index를 다시 indexfile에 작성해 주었다.

* **18**

**단순 데이터 파일로부터 인덱스의 생성을 지원하기 위해 클래스 TextIndexedFile를 확장하라. 즉 존재하는 데이터 파일을 열고 읽고, 파일내의 레코드로부터 인덱스를 생성함으로써 TextIndexedFile 객체를 초기화하는 메소드를 첨가하라.**

Init method를 추가해 구현하였다. Init 함수를 부르면 먼저 datafile의 pointer를처음으로 돌린 다음, 하나하나 읽으며 index.insert를 통해 index를 새롭게 넣어준다. 그렇게 해서 레코드로부터 인덱스를 새롭게 생성해 주었다.