

HW7. Cosequential Processing

부산대학교 정보컴퓨터공학부

2020-55645

신세환

제출일: 2024-05-22

1. 문제정의

4장, 5장, 8장 코드 최대한 활용해서 Student클래스, CourseRegistration클래스 객체 정보로 예제의 ledger와 journal처럼 학생 id 기준으로 매핑해서 보고서 형태로 출력가능하게 만들어보자.

2. 클래스 정의 설명

템플릿 클래스 CosequentialProcess를 <int>형식으로 상속받아서 mastrans.h를 만들고 이를 다시 상속받아서 ledgerprocess클래스를 만든다.

3. 클래스의 주요 함수 설명

Student 클래스의 id로 매핑시켜야 하기 때문에 courseRegistration에서도 특정 student id에 해당하는 정보만 빼 올 수 있는 메소드가 필요하다. 그리고 이를 관리할 ledgerprocess클래스도 이에 맞게 수정이 필요하다.

1) courseRegistration

함수 부분

```
int CourseRegistration::getNextSID()
{
    if(studentIndex < studentID.size())
    {
        return studentID[studentIndex++];
    }
    else
    {
        return -1;
    }
}
```

특정 인덱스에 해당하는 학생 성적만 참조 가능하게 만들기 위해서 인덱스를 반환하는 getNextSID()를 추가로 정의해주었다.

2) LedgerProcess 메소드 수정

함수 부분

```
int LedgerProcess::NextItemInJournal ()
{
    int nextSID = cr.getNextSID();
    if (nextSID == -1)
    {
        int res = CRFile.Read(cr);
        if (res <= 0)
        {
            AccountNumber[2] = HighAcct;
            return FALSE;
        }
        nextSID = cr.getNextSID();
    }
    if (nextSID < AccountNumber[2])
    {
        cerr<<"course registration out of order"<<endl; exit(0);}
    AccountNumber[2]=nextSID;
    return TRUE;
}
```

getNextSID()를 사용해서 학생 id 인덱스를 반환받고 -1이 아니라면 다음 학생 정보가 남아있다고 판단하고 사용한다. -1인 경우 해당 객체 내에서는 다음 학생 정보가 없는 것이기 때문에 다음 객체로 넘어간다.

3) courseRegistration::printStudentLine()

함수 부분

```
ostream& CourseRegistration::printStudentLine(ostream& stream, int sid, int smst)
{
    int target = -1;

    for(int i = 0; i < studentID.size(); i++)
```

```

    {
        if(studentID[i] == sid && (smst == semester || smst == -1)){
            target = i;
            break;
        }
    }

    if(target >= 0)
    {
        stream << "courseID : Wt" << courseID << "WtcreditHours : Wt" <<
        creditHours << "Wtsemester : Wt" << semester
            << "WtstudentID : Wt" << studentID[target] << "Wtgrade : Wt" <<
        courseGrade[target]
            << endl;
    }
}

```

특정 ID, 특정 학기에 해당하는 정보가 있을 때 출력할 수 있도록 courseRegistration에 printStudentLine()함수를 추가적으로 정의해주었다.

4) Setting()

```

함수 부분
int setting()
{
    LengthFieldBuffer Lbuffer;
    LedgerWrite(Lbuffer,"student.dat");

    LengthFieldBuffer Jbuffer;
    JournalWrite(Jbuffer, "cr.dat");

    LedgerRead(Lbuffer,"student.dat");
    JournalRead(Jbuffer, "cr.dat");
}

```

main함수가 존재하는 testLedgerMain.cpp에 테스트할 객체를 저장할 파일을 미리 만들어주는 setting()함수를 정의해두었다.

4. 프로그램 실행

1) 프로그램 실행 환경

- wsl Ubuntu에서 g++ (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
을 사용하였다.

2) 프로그램 실행 방법

컴파일 및 실행은 정상적으로 되는데 환경설정 할 때 사용하는 recfile 클래스가 바이너리로 입
출력 하는 형태라

소스로 사용하는 파일인 student.dat과 cr.dat이 txt파일과 호환되지 않아서 결과를 확인하지는
못했습니다.

파일 형식이 달라서 에러 발생하는 상황입니다.

```
make[1]: Leaving directory '/home/sk3456/git_project/hw7_cosequential_processing/fs24hw7-sinsehwan/student_rec'
g++ -o testLedgerMain testLedgerMain.o -L./lib -lmyledger -lmybuffer -lmycoseq -lmystring -lmystrlistp -lmyvar -Wl,-rpath=./lib
#g++ -o testLedgerMain testLedgerMain.o -L./lib -lmybuffer -lmycoseq -lmystring -lmystrlistp -lmyvar -lmyledger -lmyledgepost -
b
#g++ -o testIdxFile testLedgerMain.o -L./lib -lmybuffer -lmyfixed -lmyindex -lmyrecording -lmyvar -Wl,-rpath=./lib
#export LD_LIBRARY_PATH=./lib
sk3456@DESKTOP-PQL1ORV:~/git_project/hw7_cosequential_processing/fs24hw7-sinsehwan$ ./testLedgerMain
Student Samples=====
terminate called after throwing an instance of 'std::invalid_argument'
  what():  stoi
Aborted
sk3456@DESKTOP-PQL1ORV:~/git_project/hw7_cosequential_processing/fs24hw7-sinsehwan$
```

5. Github 화면

(1) cloning, adding, committing, push을 위한 github 명령들을 포함

```

rm -f testLedgerMain testLedgerMain.o ./lib
sk3456@DESKTOP-PQL1ORV:~/git_project/hw7_cosequential_processing/fs24hw7-sinsehwan$ git add .
sk3456@DESKTOP-PQL1ORV:~/git_project/hw7_cosequential_processing/fs24hw7-sinsehwan$ git commit -m "src code v1 [add]"
[main c1173f5] src code v1 [add]
42 files changed, 2215 insertions(+)
create mode 100644 .vscode/settings.json
create mode 100644 Makefile
create mode 100644 buf/Makefile
create mode 100644 buf/bufferfile.cpp
create mode 100644 buf/ioBuffer.cpp
create mode 100644 buf/recfile.cpp
create mode 100644 coseq/Makefile
create mode 100644 coseq/coseq.cpp
create mode 100644 coseq/mastrans.cpp
create mode 100644 cr.dat
create mode 100644 include/bufferfile.h
create mode 100644 include/coseq.h
create mode 100644 include/courseRegistration.h
create mode 100644 include/delim.h
create mode 100644 include/ioBuffer.h
create mode 100644 include/ledgerpost.h
create mode 100644 include/ledger.h
create mode 100644 include/length.h
create mode 100644 include/mastrans.h
create mode 100644 include/recfile.h
create mode 100644 include/strclass.h
create mode 100644 include/strlist.h
create mode 100644 include/student.h
create mode 100644 include/varlen.h
create mode 100644 ledgerpost/Makefile
create mode 100644 ledgerpost/ledgerpost.cpp
create mode 100644 report.txt
create mode 100644 str/Makefile
create mode 100644 str/strclass.cpp
create mode 100644 strlistp/Makefile
create mode 100644 strlistp/strlist.cpp
create mode 100644 student.dat
create mode 100644 student_reg/Makefile
create mode 100644 student_reg/courseRegistration.cpp
create mode 100644 student_reg/ledgerpost.cpp
create mode 100644 student_reg/student.cpp
create mode 100644 testLedgerMain.cpp
create mode 100644 testSetting.cpp
create mode 100644 var/Makefile
create mode 100644 var/delim.cpp
create mode 100644 var/length.cpp

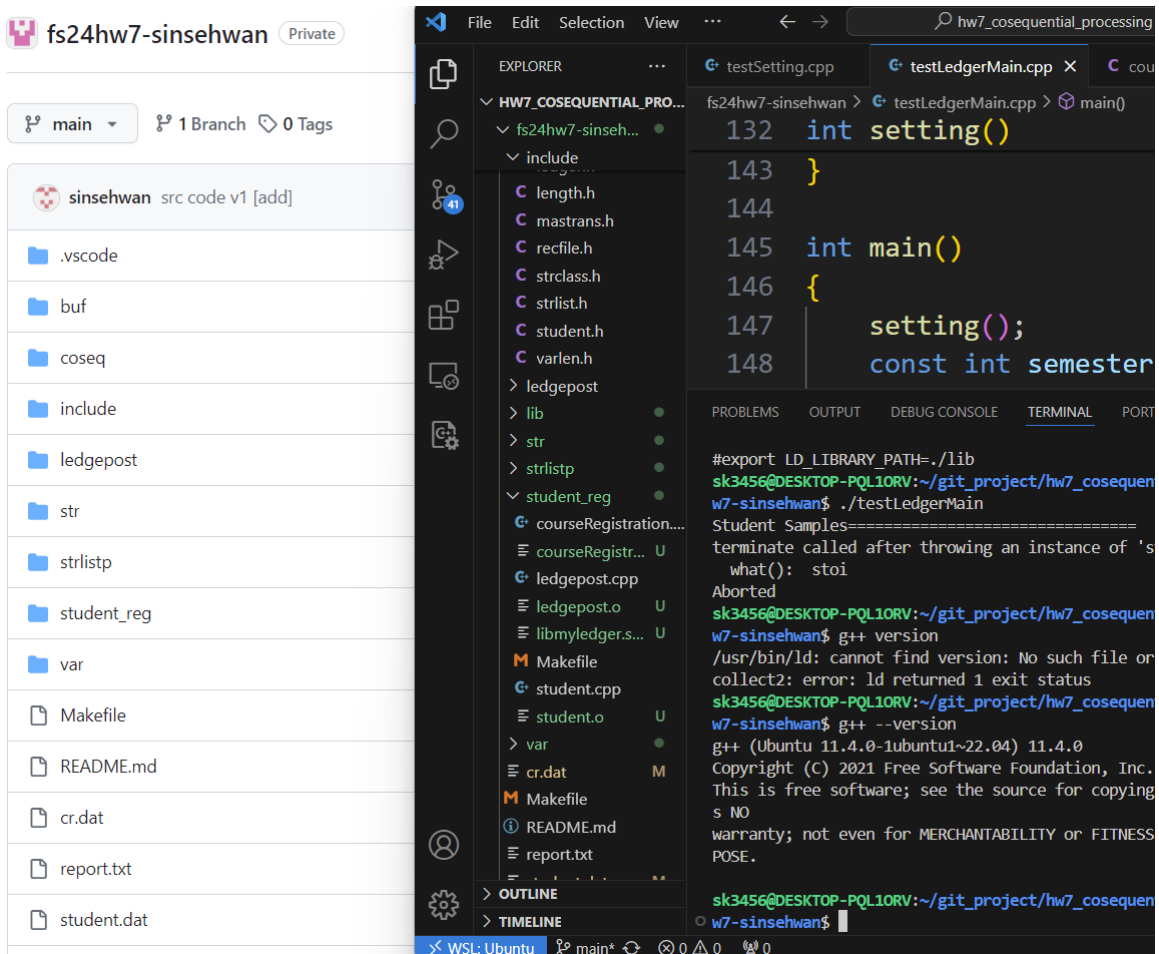
```

```

create mode 100644 testLedgerMain.cpp
create mode 100644 testSetting.cpp
create mode 100644 var/Makefile
create mode 100644 var/delim.cpp
create mode 100644 var/length.cpp
create mode 100644 var/varlen.cpp
sk3456@DESKTOP-PQL1ORV:~/git_project/hw7_cosequential_processing/fs24hw7-sinsehwan$ git push
Enumerating objects: 54, done.
Counting objects: 100% (54/54), done.
Delta compression using up to 16 threads
Compressing objects: 100% (51/51), done.
Writing objects: 100% (53/53), 20.29 KiB | 5.07 MiB/s, done.
Total 53 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), done.
To https://github.com/datalab-pnu/fs24hw7-sinsehwan.git
   f58dd00..c1173f5  main -> main
sk3456@DESKTOP-PQL1ORV:~/git_project/hw7_cosequential_processing/fs24hw7-sinsehwan$ make
mkdir -p ./lib
g++ -c -I./include -Wall -o testLedgerMain.o testLedgerMain.cpp
testLedgerMain.cpp: In function 'int setting()':
testLedgerMain.cpp:135:29: warning: ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]
  135 |         LedgerWrite(Lbuffer,"student.dat");
      |                               ^~~~~~
testLedgerMain.cpp:138:31: warning: ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]
  138 |         JournalWrite(Jbuffer, "cr.dat");
      |                               ^~~~~~
testLedgerMain.cpp:140:28: warning: ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]
  140 |         LedgerRead(Lbuffer,"student.dat");

```

(2) 소스 코드와 makefile을 push한 후, 본인의 Github repository를 스크린 캡처하여 포함



6. 논의 사항

- 테스트 프로그램에서 쓸 객체 내용 저장 파일을 txt로 만들어서 전체 프로세스를 테스트 해 보는 게 필요하다.
- CosequentialProcess::<int>Item(int)가 ledgerpost.cpp파일에 구현되어 있는데 구현이 안 되었고 나오는 템플릿 오류가 있었다. 예제 소스를 참고해서 최대한 에러가 안 나도록 구성해서 해결했지만 템플릿 오류가 뜬 근본적인 이유를 알지 못하겠다. 시간이 된다면 전체 코드를 다시 봐야 할 것 같다.