

HW번호. 제목

부산대학교 정보컴퓨터공학부

2020-55645

신세환

제출일: 2024-04-18

보고서에는 다음과 같은 내용을 포함하도록 한다. 양식은 자유롭게 수정하여 사용한다.

1. 문제정의

5.3절의 RecordFile 구현을 makefile이 하나인 하나의 폴더로 구현하지 말고 소스 코드를 폴더 별로 분리하여 makefile을 Recursive하게 적용해보자.

2. Makefile 설명

1) Buf/makefile

```
CC=g++ # 컴파일러 지정
LIBNAME=libmybuf # 라이브러리 이름 지정
LIB=$(LIBNAME).so.1.0 # 라이브러리 파일 이름 지정
CFLAGS= -Wall # 컴파일 옵션 : 모든 경고 표시
OBJS = buffile.o iobuffer.o # 컴파일할 소스 파일 명시

INCDIR=../include # 헤더 파일 위치 지정
LIBDIR=../lib # 라이브러리 저장 위치 지정
# 나중에 추가적으로 export LD_LIBRARY_PATH=/mnt/d/github/fs2024-code01-sinsehwon/ch05/ch5.3_RecordFile/lib:$LD_LIBRARY_PATH
# 이런식으로 환경변수값 변경해줘야 함
$(LIB): $(OBJS) # OBJS를 통해 라이브러리 구성
    $(CC) -fPIC -shared -Wl,-soname=$(LIBNAME).so.1 $(OBJS) -o $@ -lc
    cp $(LIB) $(LIBDIR)
    ln -s $(LIBDIR)/$(LIBNAME).so.1.0 $(LIBDIR)/$(LIBNAME).so
    ln -s $(LIBDIR)/$(LIBNAME).so.1.0 $(LIBDIR)/$(LIBNAME).so.1

# OBJS를 활용해서 공유 라이브러리를 위치 독립적으로 만든다. 이때 링커에게 추가 정보
```

```

를 제공한다.
# 생성된 공유 라이브러리를 $(LIBDIR)로 복사한다.
# .so.1.0을 가리키는 .so 심볼릭 링크를 만든다.
# .so.1.0을 가리키는 .so.1 심볼릭 링크를 만든다.

%.o: %.cpp # cpp파일을 object파일로 컴파일
    $(CC) -fPIC -c $(CFLAGS) -I$(INCDIR) -o $@ $<

clean: # object파일과 라이브러리 파일 삭제
    -rm -rf $(OBJS) $(LIB) $(LIBDIR)/$(LIBNAME).so*

```

2) makefile

```

CC=g++ # 컴파일러 지정
CFLAGS=-Wall # 컴파일 경고 옵션 지정
OBJS = makerec.o recfile.o # 사용할 object 파일 지정
INCDIR=./include # 헤더 파일 위치 지정
LIBDIR=./lib # 라이브러리 위치 지정
#export LD_LIBRARY_PATH=/mnt/d/github/fs2024-code01-
sinsehwon/ch05/ch5.3_RecordFile/lib:$LD_LIBRARY_PATH
#사용 필요
DIRS = buf record_file var # makefile수행할 하위 디렉토리 목록

.PHONY: all clean # all과 clean이 파일이 아니라 명령어임을 나타냄

all: RecordFileTest # RecordFileTest이 최종 타겟임을 나타냄
%.o: %.cpp # 헤더파일을 통해 cpp파일을 컴파일함
    $(CC) -c -I$(INCDIR) $(CFLAGS) -o $@ $<

RecordFileTest: $(OBJS)
    @for d in $(DIRS); \
    do \
        $(MAKE) -C $$d; \
    done
    $(CC) -o RecordFileTest $(OBJS) -L$(LIBDIR) -lmybuf -lmyrecord_file
-lmyvar
    export LD_LIBRARY_PATH=$(LIBDIR)
# 디렉토리를 순회하면서 반복문 수행
# 각 디렉토리에서 makefile을 수행한다.
# -L플래그로 라이브러리 경로를 지정해주고 -l로 링크할 라이브러리를 지정해준
상태에서 RecordFileTest를 만든다.

```

```
# 동적 라이브러리 경로 지정
clean:
    @for d in $(DIRS); \W
    do \W
        $(MAKE) -C $$d clean; \W
    done
    -rm -rf RecordFileTest $(OBS)
# 디렉토리를 순회하면서 make clean연산을 수행한다.
# 이후 RecordFileTest파일과 obj파일들을 삭제한다.
```

3. 프로그램 실행

1) 프로그램 실행 환경

- WSL Ubuntu 환경에서 g++ (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0를 사용하였다.
- export문을 사용해서 LD_LIBRARY_PATH 값을 변경시킨다.

export

LD_LIBRARY_PATH=/home/sk3456/git_project/hw5_recursive_compile_practice/fs24hw5-sinsehwan/lib:\$LD_LIBRARY_PATH로 수정해주었다.

2) 프로그램 실행 방법

LD_LIBRARY_PATH 설정 후 make수행

```
sk3456@DESKTOP-PQL10RV:~/git_project/hw5_recursive_compile_practice/fs24hw5-sinsehwan$ export LD_LIBRARY_PATH=/home/sk3456/git_project/hw5_recursive_compile_practice/fs24hw5-sinsehwan/lib:$LD_LIBRARY_PATH
sk3456@DESKTOP-PQL10RV:~/git_project/hw5_recursive_compile_practice/fs24hw5-sinsehwan$ make
g++ -c -I./include -Wall -o makerec.o makerec.cpp
makerec.cpp: In function 'int main()':
makerec.cpp:18:31: warning: ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]
    18 |         RecordingFile Create ("record.dat", ios::out);
```

./RecordFileTest

```
sk3456@DESKTOP-PQL10RV:~/git_project/hw5_recursive_compile_practice/fs24hw5-sinsehwan$ ./RecordFileTest
Recording R[0] at recaddr 8
Recording2 R[0] at recaddr 8
Recording R[1] at recaddr 53
Recording2 R[1] at recaddr 53
Recording R[2] at recaddr 108
Recording2 R[2] at recaddr 108
Recording R[3] at recaddr 143
Recording2 R[3] at recaddr 143
Recording R[4] at recaddr 187
Recording2 R[4] at recaddr 187
Recording R[5] at recaddr 232
Recording2 R[5] at recaddr 232
Recording R[6] at recaddr 276
Recording2 R[6] at recaddr 276
Recording R[7] at recaddr 329
Recording2 R[7] at recaddr 329
Recording R[8] at recaddr 373
Recording2 R[8] at recaddr 373
Recording R[9] at recaddr 418
Recording2 R[9] at recaddr 418
```

4. Github 화면

Git add, git commit

```
sk3456@DESKTOP-PQL1ORV:~/git_project/hw5_recursive_compile_practice/fs24hw5-sinsehwan$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Makefile
    RecordFileTest
    buf/
    include/
    lib/
    makerec.cpp
    makerec.o
    parts.cpp
    recfile.cpp
    recfile.o
    record.dat
    record2.dat
    record_file/
    var/

nothing added to commit but untracked files present (use "git add" to track)
sk3456@DESKTOP-PQL1ORV:~/git_project/hw5_recursive_compile_practice/fs24hw5-sinsehwan$ git add .
sk3456@DESKTOP-PQL1ORV:~/git_project/hw5_recursive_compile_practice/fs24hw5-sinsehwan$ git commit -m "recursive makefile source file [Add]"
[main 5f21705] recursive makefile source file [Add]
 41 files changed, 896 insertions(+)
 create mode 100644 Makefile
 create mode 100755 RecordFileTest
 create mode 100644 buf/Makefile
 create mode 100644 buf/Makefile.txt
 create mode 100644 buf/buffile.cpp
 create mode 100644 buf/buffile.o
```

Git push

```
sk3456@DESKTOP-PQL1ORV:~/git_project/hw5_recursive_compile_practice/fs24hw5-sinsehwan$ git push
Enumerating objects: 42, done.
Counting objects: 100% (42/42), done.
Delta compression using up to 16 threads
Compressing objects: 100% (38/38), done.
Writing objects: 100% (41/41), 41.78 KiB | 5.22 MiB/s, done.
Total 41 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), done.
To https://github.com/datalab-pnu/fs24hw5-sinsehwan.git
 6f84d83..5f21705  main -> main
sk3456@DESKTOP-PQL1ORV:~/git_project/hw5_recursive_compile_practice/fs24hw5-sinsehwan$
```

5. 논의 사항

template관련 에러 미발생