광운대학교 KwangWoon University

Assignment 2

Open Source SW

Kwangwoon Univ.
Dept. of Computer Engineering
Ki-Hoon Lee

Requirements

- **Softcopy**
  - 보고서: **2018. 04. 27(금) 12:00**까지 U-캠퍼스로 제출**(개인별 제출)**
    - Due Delay 없음
    - 업로드 양식에 어긋날 경우 감점 처리
    - 소스 코드를 제외한 개인별 보고서만 제출
    - 보고서 파일 명은 **OSS_과제번호_학번**으로 작성
      - e.g. OSS_2_2012XXXXXX.**pdf** **보고서는 PDF로 변환하여 제출**
  - 소스코드: **2018. 04. 27(금) 12:00**까지 github상에 올라와 있는 코드로 확인
    - 마감 날짜 이후에 코드를 push하거나 Readme.md나 wiki등 수정한 내역이 있을 경우 감점
    - 소스코드는 주어진 스켈레톤 코드(main.c)를 수정해서 사용할 것
    - repository 이름은 **Assignment2**로 작성하고, 공개범위는 반드시 **public**으로 할 것

Assignment 2

- 파일 비교 프로그램 구현
  - linux **ubuntu**상에서 두 개의 파일을 비교하는 프로그램 구현
  - 다음과 같은 기능을 구현하되, milestone과 issue를 이용해서 개발을 진행
    1. 파일 정보 받아 오기
    2. 파일 시간 받아 오기
    3. 파일 사이즈 비교
    4. 파일 블락 수 비교
    5. 파일 수정 날짜(월/일) 비교
    6. 파일 수정 시간(시/분) 비교
  - 각각의 기능별로 milestone을 생성하고 해당 milestone내의 to-do를 issue로 관리
  - 코드 취합은 **반드시 git을 사용**할 것
    - commit하지 않은 팀원이 존재할 경우 감점
    - 코드를 통째로 올렸을 경우 감점

Assignment 2

- 파일 비교 프로그램
  - 스켈레톤 코드 내부의 함수를 작성하여 commit할 것

```c
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <time.h>

struct stat stat1, stat2;
struct tm *time1, *time2;

void filestat1(void);
void filestat2(void);
void filetime1(void);
void filetime2(void);
void sizecmp(void);
void blockcmp(void);
void datecmp(void);
void timecmp(void);

int main(void)
{
    filestat1();
    filestat2();
    filetime1();
    filetime2();
    sizecmp();
    blockcmp();
    datecmp();
    timecmp();
}
```

```c
//파일 1의 정보를 가져오는 함수 작성
void filestat1(void)
{
}

//파일 2의 정보를 가져오는 함수 작성
void filestat2(void)
{
}

//파일 1의 시간 정보를 가져오는 함수 작성
void filetime1(void)
{
}

//파일 2의 시간 정보를 가져오는 함수 작성
void filetime2(void)
{
}
```

```c
//두 개의 파일 크기를 비교하는 함수 작성
void sizecmp(void)
{
}

//두 개의 파일 블락 수를 비교하는 함수 작성
void blockcmp(void)
{
}

//두 개의 파일 수정 날짜를 비교하는 함수 작성
void datecmp(void)
{
}

//두 개의 파일 수정 시간를 비교하는 함수 작성
void timecmp(void)
{
}
```

각각의 함수를 팀원별로 알맞게 분배하여 구현할 것
e.g.) a는 1,2번 함수 구현, b는 3, 4번 함수 구현…

Assignment 2

- 파일 블락 수 비교

```
//두 개의 파일 블락 수를 비교하는 함수 작성
void blockcmp(void)
{
}
```

  - ※ block size?
    - 데이터 또는 프로그램의 한 블록 크기이며, 정보를 처리하는 논리적인 단위로 보통 레코드의 집합으로 구성됨

Assignment 2

- 두 개의 파일을 비교하는 함수에서는 두 개의 파일을 비교한 결과를 출력하는 것을 구현
  - e.g.) 파일 크기 비교 함수에서는 두 개의 파일을 비교하고 어느 파일이 더 큰지(or 작은지) 출력
    - 둘 중 하나가 크면 "OOO is bigger" 출력
    - 같으면 "sizes are equal" 출력
  - e.g.) 파일 수정 시간 비교 함수에서는 두 개의 파일을 비교하고 어느 파일이 더 최근에 수정했는지 출력
    - 둘 중 하나가 빠르면 "OOO is early" 출력
    - 같으면 "same time" 출력

- 첫 번째 파일 이름은 text1로, 두 번째 파일 이름은 text2로 지정할 것
  - **text1.txt, text2.txt가 아닌 text1, text2로 지정할 것**

Assignment 2

- 팀장은 github repository를 생성
- 팀원은 해당 github repository를 fork하고 checkout하여 작업을 진행
- **코드 취합은 반드시 git의 fork & pull request를 이용하여 진행**
- 코드를 한번에 통째로 올리지 말 것
  - e.g.) 각각의 함수를 작성한 뒤 복사 및 붙여넣기를 사용해 텍스트 파일로 수동으로 합쳐서 한 파일로 올리지 말 것
- Pull request를 통하지 않고 master branch에 바로 push할 경우 감점
- pull request를 보냈을 경우 나머지 팀원들이 확인 댓글을 단 뒤 merge를 수행

Assignment 2

- 결과 화면
  - 프로그램을 실행한 결과와 ls –l의 결과를 모두 캡쳐할 것

```
root@linkaden:~/2014722064_OSS# ./main
size compare
text2 is bigger

block compare
text2 is bigger

date compare
same date

time compare
same time
```

프로그램 실행 결과

```
root@linkaden:~/2014722064_OSS# ls -l
total 40
-rwxr-xr-x 1 root root  9176  2월  12 16:15 main
-rw-r--r-- 1 root root  2111  2월  12 16:15 main.c
-rwxr-xr-x 1 root root  8824  2월  12 16:48 test
-rw-r--r-- 1 root root   476  2월  12 16:47 test.c
-rw-r--r-- 1 root root    38  2월  12 16:13 text1
-rw-r--r-- 1 root root  1977  2월  12 16:13 text2
root@linka
```

ls -l의 실행 결과

파일 사이즈

수정 날짜와 시간

Assignment 2

- Readme 작성
  - Readme 파일은 repository 폴더 최상단에 Read.md로 존재함
  - Repository 생성시 Readme.md를 생성하면서 초기화할 수도 있고, 임의로 생성할 수도 있음
  - Markdown 문법 사용
- Readme에는 과제명, 팀장 및 팀원 이름과 github id를 적을 것

Assignment2

팀장

2014722064 OOO(github id)

팀원

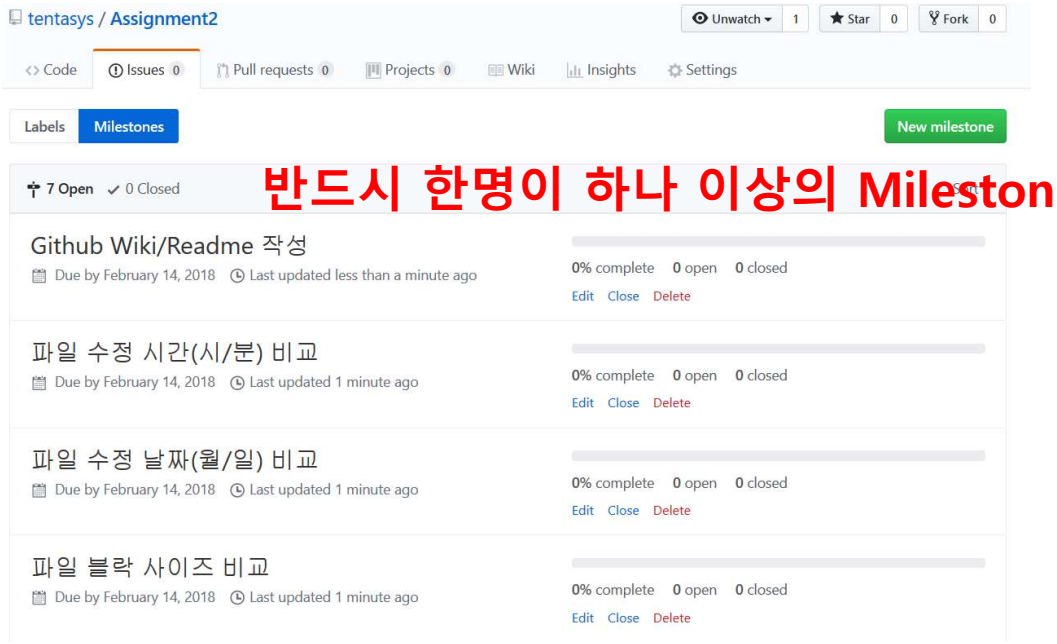
20147220XX OOO (github id)

20147220XX OOO (github id)

20147220XX OOO (github id)

Assignment 2

- Milestone 생성
  - 각각의 모듈(기능 1~6)별 milestone과 github wiki/readme 작성 milestone은 반드시 생성해야 함(총 milestone은 3개 이상)
  - 그 이외에 필요하다고 생각하는 milestone을 작성해도 무방함



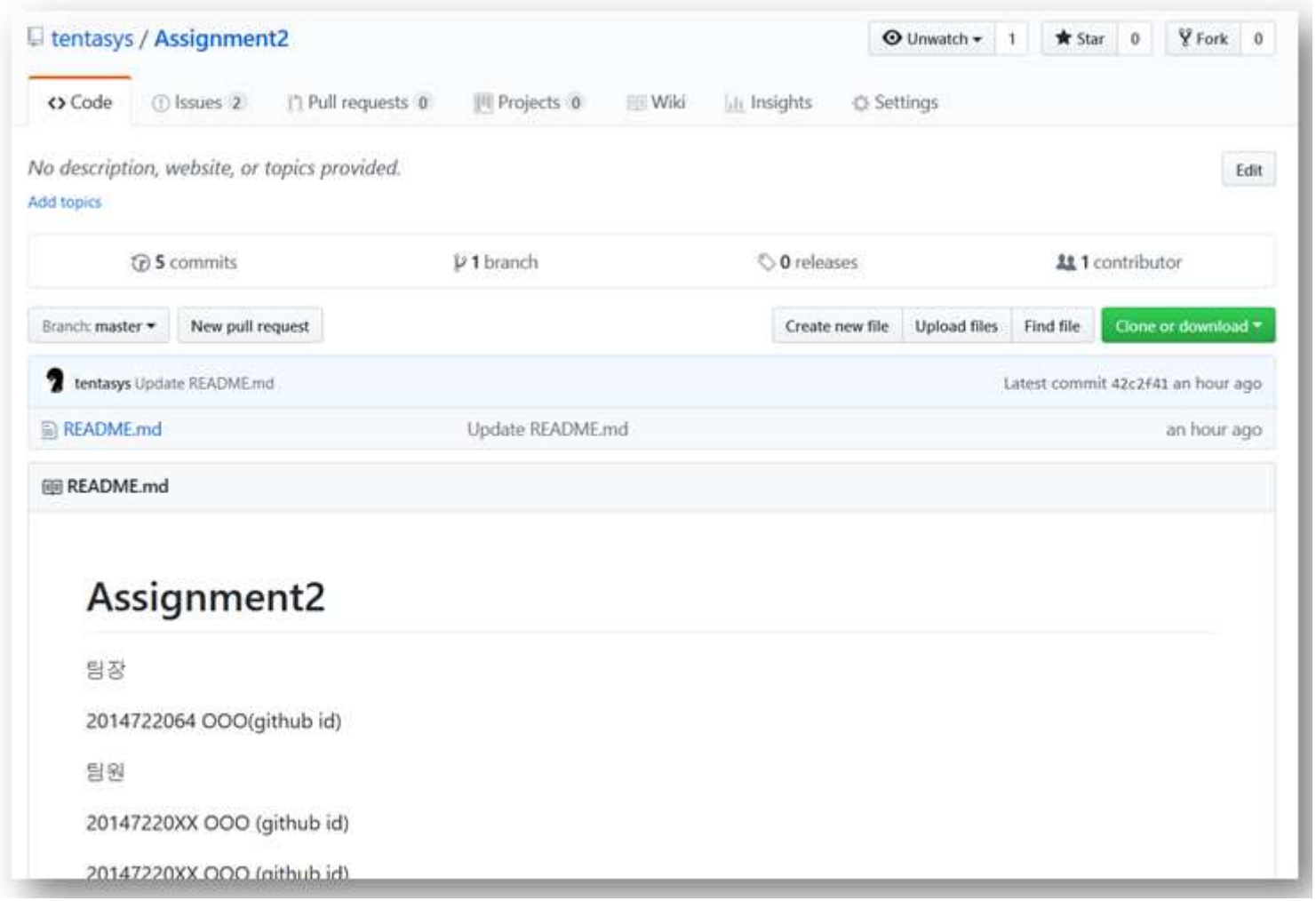
반드시 한명이 하나 이상의 Milestone을 담당할 것

Assignment 2

- Issue 작성
  - Milestone을 달성하는데 필요하다고 생각하는 step을 issue로 생성하여 관리
  - Milestone에 맞게 issue와 assignee가 잘 할당되어 있어야 함
  - 과제 마감 시 closed상태가 아닌 issue가 있으면 감점
  - Issue의 업데이트를 활발히 하여 현재 자신이 어느 작업을 하고 있는지 팀원이 알 수 있도록 해야 함
  - 한 milestone에 최소한 하나 이상의 issue가 존재해야 함

Assignment 2

- github 메인 캡처는 다음과 같은 화면을 캡처할 것



Assignment 2 Scenario

- A, B, C, D 4명으로 구성된 팀의 경우, A를 팀장으로 선정
- A는 github repository를 생성하고 그 주소를 팀원들에게 공유
- B, C, D는 해당 repository를 fork하거나 A가 B, C, D를 collaborator로 등록시킨 다음에 branch를 checkout
- A 또한 master branch가 아닌 새로운 branch를 생성하고 checkout
- A는 함수 1, 2, 3을 구현, B는 함수 4, 5, 6을 구현, C는 함수 7, 8을 구현, D는 readme 작성 및 wiki page를 작성하도록 역할을 분배
- A는 함수1 구현, 함수2 구현, 함수3 구현이라는 1개 이상의 milestone을 생성(다른 팀원들도 동일하게 milestone을 생성)
- A는 함수1 구현에 기반 기술 조사, 코드 작성, 코드 리뷰라는 3개의 issue를 생성(다른 팀원 및 다른 milestone들도 동일)
- 작업 진행 중, issue가 완료되면 issue를 close

Assignment 2 Scenario

- A가 함수1 구현을 완료한 경우, git push를 진행하고 pull request를 보냄
- B, C, D는 pull request를 확인하고 확인했다는 내용의 댓글을 작성
- B, C, D의 확인 댓글이 달리면 pull request를 merge하고 pull request를 close함
- 충돌이 발생했을 경우 충돌이 발생했다는 issue를 생성하거나 pull request에 충돌이 발생한 사람과 댓글을 주고받으면서 충돌을 해결

평가 기준

- 과제의 목적은 git의 기능을 활용하는 것이기 때문에 코드 공유 시 단순공유(카톡으로 올리기)를 하지 않고 git의 기능을 활용하여 공유할 것
- Github의 repository에서 fork와 merge를 잘 사용했고 branch를 잘 관리하였는가
- Github의 issue와 milestone을 잘 활용하여 의사소통하였는가

평가 기준

- 다음과 같은 요구사항을 충족해야 함

소스코드
파일 사이즈 비교가 제대로 되는가?
파일 블락 사이즈 비교가 제대로 되는가?
파일 수정 날짜 비교가 제대로 되는가?
파일 수정 시간 비교가 제대로 되는가?
결과화면을 제대로 출력하는가?
Runtime error가 일어나지 않고 제대로 동작하는가?
정해진 코드 양식에 따라 작성했는가?
코드 취합 과정에서 git을 사용하였는가?

평가 기준

- 다음과 같은 요구사항을 충족해야 함

Github
Milestone이 task별로 잘 나뉘어져 있는가?
Issue가 Milestone에 제대로 할당되어 있는가?
Issue에 Assignee가 할당이 되어 있는가?
Task를 팀원에게 균등하게 배분하였는가?
Issue와 comment를 이용하여 팀원 간의 소통이 원활이 이루어졌는가?
과제 마감 시 closed되지 않은 issue가 존재하는가?
Readme.md를 제대로 작성하였는가? (가독성보다는 내용 중시)
Wiki page를 제대로 작성하였는가? (가독성보다는 내용 중시)
팀원 전부가 commit에 참여하였는가?
마감 기한에 맞춰 commit이 이루어졌는가?

평가 기준

- 다음과 같은 요구사항을 충족해야 함

Report
Github repository 주소가 제대로 적혀 있는가?
결과 화면 캡쳐가 존재하는가?
Github repository 메인 화면 캡쳐가 존재하는가?
고찰을 잘 작성하였는가?

- 이 외에도 본 문서에 나타난 요구사항을 지키지 않았을 시 감점이 될 수 있음

Appendix

- stat() : 파일의 정보를 받아오는 함수

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>

int stat(const char* path, struct stat* buf);
```

- const char* path : 파일의 경로
- struct stat* buf : 파일의 정보가 들어가는 구조체 포인터

```
struct stat {
dev_t st_dev; /* ID of device containing file */
ino_t st_ino; /* inode number */
mode_t st_mode; /* protection */
nlink_t st_nlink; /* number of hard links */
uid_t st_uid; /* user ID of owner */
gid_t st_gid; /* group ID of owner */
dev_t st_rdev; /* device ID (if special file)
```

```
off_t st_size; /* total size, in bytes */
blksize_t st_blksize; /* blocksize for file system I/O */
blkcnt_t st_blocks; /* number of 512B blocks allocated */
time_t st_atime; /* time of last access */
time_t st_mtime; /* time of last modification */
time_t st_ctime; /* time of last status change */
};
```

Appendix

- localtime(): time_t값을 활용할 수 있게 가공시켜주는 함수

```
#include <time.h>

struct tm *localtime(const time_t* timep);
```

- const time_t* timep : time_t형 변수를 가리키는 포인터
- struct tm* localtime : 시간 정보가 들어가는 구조체 포인터

```
struct tm {
int tm_sec; /* seconds */
int tm_min; /* minutes */
int tm_hour; /* hours */
int tm_mday; /* day of the month */
int tm_mon; /* month */
int tm_year; /* year */
int tm_wday; /* day of the week */
int tm_yday; /* day in the year */
int tm_isdst; /* daylight saving time */
};
```

- month의 경우 0부터 시작함(1월이 0으로 나타남)

Appendix

- stat과 localtime을 이용하여 파일의 정보를 가져오는 예제

main.c | test.c

```c
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <time.h>

int main(void)
{
    struct stat buf;
    struct tm* time;

    stat("text1", &buf);
    time = localtime(&buf.st_mtime);
    printf("size : %d \n", (int)buf.st_size);
    printf("blocks : %d \n", (int)buf.st_blocks);
    printf("month : %d \n", time->tm_mon+1);
    printf("date : %d \n", time->tm_mday);
    printf("hour : %d \n", time->tm_hour);
    printf("min : %d \n", time->tm_min);

    return 0;
}
```

```
root@linkaden:~/2014722064_OSS# ls -al
total 48
drwxr-xr-x  2 root root 4096  2월 12 16:48 .
drwx------ 29 root root 4096  3월 27 16:41 ..
-rwxr-xr-x  1 root root 9176  2월 12 16:15 main
-rw-r--r--  1 root root 2111  2월 12 16:15 main.c
-rwxr-xr-x  1 root root 8824  2월 12 16:48 test
-rw-r--r--  1 root root  476  2월 12 16:47 test.c
-rw-r--r--  1 root root   38  2월 12 16:13 text1
-rw-r--r--  1 root root 1977  2월 12 16:13 text2
root@linkaden:~/2014722064_OSS# ./test
size : 38
blocks : 8
month : 2
date : 12
hour : 16
min : 13
```

Appendix

- Github 예시 Repository
  https://github.com/tentasys/Assignment2