

Spring 2019 Operating Systems

Lab 06: Mini Project

0. 과제 수행 전 유의사항

프로그래밍은 자기 손으로 작성해야 한다. 아이디어를 서로 공유하고 도움을 받는 것은 좋으나, 다른 사람의 코드를 그대로 가져다 쓰는 것은 자신에게도 손해이고 다른 사람에게 피해를 끼치는 결과를 낳게되므로 허용하지 않는다. 과제 체크 시 두 사람의 코드가 유사하면, 두 사람 모두에게 과제 점수를 부여할 수 없으니 유의바란다.

1. 준비사항

- 자신의 홈 디렉토리에 프로젝트용 디렉토리를 하나 생성하고, 그 디렉토리로 이동한다.
- 다음 명령을 수행하여 프로젝트에 필요한 파일들을 자신의 디렉토리로 복사한다.

```
$ cp -r /opt/* .
```

복사가 정상적으로 완료되었다면, 아래와 같은 구조를 갖게될 것이다.

```
/home/s_201xxxxx
├── project용 폴더
│   └── data 폴더
│       ├── diamond.txt
│       ├── ruby.txt
│       ├── stone.txt
│       └── emerald.txt
├── sha1.c
├── sha1.h
└── 자신이 추가하는 main.c, Makefile 등의 파일들
```

- SHA1 해시값을 구하는 코드는 아래와 같다. 이 코드를 컴파일해서 SHA1 해시값이 제대로 출력되는지 확인한다.

```
#include <stdio.h>
#include "sha1.h"

main()
{
    int i, n;
    FILE *fp;
    char buffer[1024];
    SHA1Context sha1;
    static unsigned char hash[20] = {0};

    fp = fopen("./data/stone.txt", "rb");

    SHA1Reset(&sha1);
    while ((n = fread(buffer, 1, sizeof(buffer), fp)) > 0) {
        SHA1Input(&sha1, buffer, n);
    }
    SHA1Result(&sha1, hash);

    printf("SHA1 hash: ");
    for (i = 0; i < 20; i++) {
        printf("%02x", hash[i]);
    }
    printf("\n");

    fclose(fp);
}
```

2. 목표

data 디렉토리를 방문하여 그 안의 파일목록을 읽어온 뒤, SHA1 해시를 수행하여 같은 해시값을 보이는 파일을 찾아 그 파일들이 이름은 다르지만 내용이 같은 파일임을 사용자에게 알린다.

목표사항

- 입력받은 디렉토리를 방문한다.
- 입력받은 디렉토리의 파일 목록들을 출력한다.
- 목록 파일들의 해시값을 각각 계산한다.
- 해시값이 같은 파일을 찾아내어 사용자에게 해당 파일들의 파일정보를 출력한다.

심화목표

- stat 구조체를 사용하여 파일 정보를 출력
- fopen이 아닌 open 함수를 사용
- Makefile 작성 (폴더를 확인할 것이므로 make 명령으로 최종 컴파일이 가능해야 한다)

가점 조건 (3점) - 등급과 관계없이 추가점 부여

- lseek 함수를 사용하여 해시값이 다른 파일들은, 해시값이 같은 파일들과 비교했을 때 어느 부분이 다른지 출력해준다.

평가등급

등급	배점	조건
A+	20	목표사항 4가지를 모두 구현하고, 심화목표 3가지를 모두 구현
A	17	목표사항 4가지를 모두 구현하고, 심화목표 2가지를 구현
B+	14	목표사항 4가지를 모두 구현하고, 심화목표 1가지를 구현
B	11	목표사항 4가지를 모두 구현
C+	8	목표사항 3가지를 구현
C	5	목표사항 2가지를 구현
D	2	목표사항 1가지를 구현
F	0	미제출 및 부정행위

3. 제출

제출은 자신이 생성한 프로젝트 폴더를 압축해서 제출한다. 예를 들어 "abc" 라는 폴더의 압축방법은 다음과 같다.

\$ tar -cf abc.tar abc

서버로 부터 압축파일을 다운받아 스마트캠퍼스에 업로드 한다. 다운 방법은 Xftp, Filezilla 등의 유틸리티를 사용하면 된다.

Appendix. 참고자료

: argv 로 2개의 인자를 받아 fd1 의 파일내용을 fd2 로 복사하는 프로그램

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>

#define MAX 1024

main(int argc, char *argv[])
{
    int fd1, fd2, count;
    char buffer[MAX];

    if (argc != 3) {
        perror("argc is not 3");
        exit(0);
    }

    if ((fd1 = open(argv[1], O_RDONLY)) == -1) {
        perror("open failed");
        exit(1);
    }

    if ((fd2 = open(argv[2], O_WRONLY | O_CREAT, 0644)) == -1) {
        perror("open failed");
        exit(1);
    }

    while ((count = read(fd1, buffer, MAX)) > 0) {
        if (write(fd2, buffer, count) != count) {
            perror("write failed");
            exit(1);
        }
    }

    close(fd1);
    close(fd2);
    exit(0);
}
```

: 이하 코드는 현재 경로를 통해 파일의 정보를 출력하는 ls와 비슷한 동작을 하는 코드이다.

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <dirent.h>
#include <pwd.h>
#include <grp.h>
#include <time.h>

#define MAX 512

main(int argc, char *argv[])
{
    int i, index;
    struct stat st;
    DIR *dp;
    struct dirent *dirp;
    char *dir;
    char filename[MAX], list[MAX], buffer[MAX];

    dir = ".";

    dp = opendir(dir);

    while (dirp = readdir(dp)) {
        index = 0;
        if (!strcmp(dirp->d_name, ".", 1))
            continue;
        sprintf(filename, "%s/%s", dir, dirp->d_name);
        lstat(filename, &st);

        switch (st.st_mode & S_IFMT) {
            case S_IFREG:
                list[index++] = '-';
                break;
            case S_IFDIR:
                list[index++] = 'd';
                break;
            case S_IFIFO:
                list[index++] = 'p';
                break;
            case S_IFLNK:
                list[index++] = 'l';
        }

        for (i = 0; i < 3; i++) {
            if (st.st_mode & (S_IREAD >> i*3)) {
                list[index++] = 'r';
            } else {
                list[index++] = '-';
            }
            if (st.st_mode & (S_IWRITE >> i*3)) {
                list[index++] = 'w';
            } else {
                list[index++] = '-';
            }
            if (st.st_mode & (S_IEXEC >> i*3)) {
                list[index++] = 'x';
            } else {
                list[index++] = '-';
            }
        }
        list[index] = '\0';

        printf("%s%5d %s ", list, st.st_nlink, (getpwuid(st.st_uid))->pw_name);
        printf("%s%9d ", (getgrgid(st.st_gid))->gr_name, st.st_size);
        sprintf(buffer, "%s", ctime(&st.st_mtime));
        buffer[strlen(buffer)-1] = '\0';
        printf("%s %s\n", buffer, dirp->d_name);
    }

    closedir(dp);
    exit(0);
}
```