

- 3장 -

3번

- 코드(p3.c)

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

char ls[10]; /* 변환한 permission 문자열 저장 */

int lsoct(char perm[]){
    /* 비트 연산자를 이용하여 이진수로 변환 후 리턴 */
    int bi = 0b0;
    int cur = 0b1;
    for(int i = 8; i > -1; i--){
        /* '-'가 아닌 경우 1, 맞는 경우 0 */
        if(perm[i] != '-'){
            bi |= cur;
        }
        cur <<= 1;
    }
    return bi;
}
```

```

char *octls(char* perm){
    int oc = atoi(perm); /* permission 10진법 정수로 변경 */
    int i = 0; /* ls 인덱스 */

    /* oc를 백의 자리수부터 문자열로 변환 */
    int mod = 100;
    while (oc > 0){
        int cur = oc / mod;
        oc %= mod;
        mod /= 10;

        /* ls 각 자리에 해당하는 문자 저장 */
        if(cur >= 4){
            ls[i] = 'r';
            cur -= 4;
        } else{
            ls[i] = '-';
        }
        if(cur >= 2){
            ls[i + 1] = 'w';
            cur -= 2;
        } else{
            ls[i + 1] = '-';
        }
        if(cur >= 1){
            ls[i + 2] = 'x';
        } else{
            ls[i + 2] = '-';
        }
        i += 3;
    }

    return ls;
}

```

```

int main(int argc, char *argv[]) {
    /* 명령이 잘못된 경우 */
    if(argc != 3) {
        printf("Wrong input. Usage : %s <lsoct or octls> <permission>\n", argv[0]);
        return 0;
    }
    /* lsoct를 실행 */
    if(argv[1][0] == 'l'){
        printf("%o\n", lsoct(argv[2]));
    }
    /* octls를 실행 */
    else if(argv[1][0] == 'o'){
        printf("%s\n", octls(argv[2]));
    }
    /* 잘못된 입력 */
    else{
        printf("Wrong input\n");
    }

    return 0;
}

```

- 실행결과1: lsoct

```

boo@boo-VirtualBox:~/Downloads/hw2/ch03$ ./test lsoct rw-r--r--
644

```

- 실행결과2: octls

```

boo@boo-VirtualBox:~/Downloads/hw2/ch03$ ./test octls 0777
rwxrwxrwx

```

6번

- 코드(p6.c)

```

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

char *filename = "afile";

void whatable(){
    /* reading permission 여부 */
    if (access (filename, R_OK) == -1){
        perror("reading: ");
    } else{
        printf ("%s is readable\n", filename);
    }
    /* writing permission 여부 */
    if (access (filename, W_OK) == -1){
        perror("writing: ");
    } else{
        printf ("%s is writable\n", filename);
    }
    /* executing permission 여부 */
    if (access (filename, X_OK) == -1){
        perror("executing: ");
    } else{
        printf ("%s is executable\n", filename);
    }
}

int main(){
    whatable();
}

```

- 실행결과

```

boo@boo-VirtualBox:~/Downloads/hw2/ch03$ ./test
afile is readable
afile is writable
executing: : Permission denied

```

7번

- 코드(p7.c)

```
#include <unistd.h>
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <stdlib.h>
```

```
/* 문자열 permission을 정수 형태로 변경하는 함수 */
```

```
int lsocet(char perm[]){
    int bi = 0b0;
    int cur = 0b1;
    for(int i = 8; i > -1; i--){
        if(perm[i] != '-'){
            bi |= cur;
        }
        cur <= 1;
    }
    return bi;
}
```

```
/* permission 변경 함수 */
```

```
int setperm(char* filename, int perm){
    if(chmod(filename, perm) == -1){
        perror("Error: ");
        return 0;
    }
    printf("Permission of %s is changed\n", filename);
    return 0;
}
```

```
int main(int argc, char *argv[]){
    /* 명령이 잘못된 경우 */
    if(argc != 3) {
        printf("Wrong input. Usage : %s <filename> <permission>\n", argv[0]);
        return 0;
    }

    int perm = 00;
    /* permission이 문자열로 입력된 경우 */
    if(argv[2][0] == 'r' || argv[2][0] == '-'){
        perm = lsocet(argv[2]);
    }
    /* permission이 8진수로 입력된 경우 */
    else{
        perm = strtol(argv[2], NULL, 8);
    }
    /* permission이 변경 */
    if(setperm(argv[1], perm) == -1){
        printf("Error is occurred\n");
    }
}
```


- 실행 결과1: 문자열 permission 입력했을 때

```
boo@boo-VirtualBox:~/Downloads/hw2/ch03$ ./test afile rwxrwxrwx
Permission of afile is changed
boo@boo-VirtualBox:~/Downloads/hw2/ch03$ ls -l
total 40
-rwxrwxrwx 1 boo boo    0 10월  8 20:40 afile
```

- 실행결과2: 정수형 permission 입력했을 때

```
boo@boo-VirtualBox:~/Downloads/hw2/ch03$ ./test afile 0644
Permission of afile is changed
boo@boo-VirtualBox:~/Downloads/hw2/ch03$ ls -l
total 40
-rw-r--r-- 1 boo boo    0 10월  8 20:40 afile
```

8번

- 코드(p8.c)

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

int main(int argc, char *argv[]){
    /* 명령이 잘못된 경우 */
    if(argc != 2) {
        printf("Wrong input. Usage : %s <filename>\n", argv[0]);
        return 0;
    }
    /* 해당 파일에 writing 권한이 없는 경우 */
    if (access (argv[1], W_OK) == -1){
        perror("Warning: ");
        printf("Do you want to continue?(y/n)\n");
        char x = getchar();
        /* 'n' 입력한 경우 종료 */
        if (x == 'n')
            return 0;
    }
    /* unlink */
    if(unlink(argv[1]) == -1){
        perror("Error: ");
        return 0;
    }

    printf("unlink %s is successful\n", argv[1]);
}
```

- 프로그램 실행 전: 'afile' 존재

```
boo@boo-VirtualBox:~/Downloads/hw2/ch03$ ls
afile p10.c p3.c p6.c p7.c p8.c test
```

- 실행 결과1: writing 권한이 있는 경우

```
boo@boo-VirtualBox:~/Downloads/hw2/ch03$ ./test afile
unlink afile is successful
boo@boo-VirtualBox:~/Downloads/hw2/ch03$ ls
p10.c p3.c p6.c p7.c p8.c test
```

- 실행결과2: writing 권한이 없는 경우

```
boo@boo-VirtualBox:~/Downloads/hw2/ch03$ ./test afile
Warning: : Permission denied
Do you want to continue?(y/n)
y
unlink afile is successful
```

10번

- 코드(p10.c)

```
#include <unistd.h>
#include <sys/stat.h>
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>

#define MFILE 10 /* 입력 파일 최대 개수 */
#define BUFSIZE 512
#define INTERVAL 3 /* 모니터링 시간 간격 */

void cmp(const char *, time_t, int j);
void slowwatch(int argc, char* argv[]);
struct stat sb;
time_t last_time[MFILE + 1];
void fileread(const char *);
```

```

int main(int argc, char *argv[]){
    /* 명령이 잘못된 경우 */
    if(argc < 2){
        fprintf(stderr, "usage: lookout filename ...\n");
        exit(1);
    }
    /* 입력 파일이 너무 많은 경우 */
    if(argc > MFILE){
        fprintf(stderr, "lookout: too many filenames\n");
        exit(1);
    }

    slowwatch(argc, argv);
}

void slowwatch(int argc, char* argv[]){
    int j;
    /* 초기화 */
    for(j = 1; j < argc; j++){
        /* if the file is not exist */
        if(access(argv[j], F_OK) == -1){
            last_time[j] = 0;
            continue;
        }
        /* get st_mtime */
        if(stat(argv[j], &sb) == -1){
            fprintf(stderr, "lookout: couldn't stat %s\n", argv[j]);
        }
        last_time[j] = sb.st_mtime;
    }

    /* 파일 변경될 때까지 루프 */
    for(;;){
        for(j = 1; j < argc; j++)
            cmp(argv[j], last_time[j], j);
        sleep(INTERVAL);
    }
}

```



```

void cmp(const char *name, time_t last, int j) {
    /* 변경시간 검사 */
    if (access(name, F_OK) != -1 && (stat(name, &sb) == -1 || sb.st_mtime != last)){
        /* 파일 변경이 감지되면 텍스트 출력 후 마지막 수정 시간 저장 */
        fprintf(stderr, "lookout: %s changed\n", name);
        last_time[j] = sb.st_mtime;
        fileread(name);
    }
}

void fileread(const char *filename){
    char buffer[BUFSIZE]; /* 문자열 저장 */
    int filesdes; /* 파일 디스크립터 */

    ssize_t nread; /* 읽은 크기 */

    /* 파일 오픈 */
    if ((filesdes = open(filename, O_RDONLY)) == -1){
        printf("error in opening %s\n", filename);
        exit(1);
    }
    /* 파일 읽기 */
    while( (nread = read(filesdes, buffer, BUFSIZE - 1)) > 0){
        buffer[nread] = '\0';
        fprintf(stderr, "contents:\n%s", buffer);
    }
    printf("\n");
    /* 파일 클로즈 */
    if (close(filesdes) == -1){
        printf("error in closing %s\n", filename);
        exit(1);
    }
}

```

- 프로그램 실행 후 afile 파일을 생성/수정 했을 때 결과

```

boo@boo-VirtualBox:~/Downloads/hw2/ch03$ ./test afile
lookout: afile changed
contents:
This is afile for exercise10

```

- 프로그램 실행 결과2: background 프로세스로 실행 후 file1과 file2 파일을 생성했을 때

```

boo@boo-VirtualBox:~/Downloads/hw2/ch03$ ./test file1 file2 & 모니터링 프로그램 실행
[1] 2660
boo@boo-VirtualBox:~/Downloads/hw2/ch03$ cat > file1 file1 생성
lookout: file1 changed 모니터링 프로그램

hello. This is file1 file1 내용 입력
lookout: file1 changed
contents:
hello. This is file1 ] 모니터링 프로그램

boo@boo-VirtualBox:~/Downloads/hw2/ch03$ cat > file2 file2 생성
lookout: file2 changed 모니터링 프로그램

hi. I created file2 file2 내용 입력
lookout: file2 changed
contents:
hi. I created file2 ] 모니터링 프로그램

```

12번

- 코드(p12.c)

```

#include <unistd.h>
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <stdlib.h>
#include <unistd.h>

struct stat sb;

/* 문자열 permission을 정수 형태로 변경하는 함수 */
int lsoct(char perm[]){
    int bi = 0b0;
    int cur = 0b1;
    /* 문자열 맨 앞에 파일 종류 정보가 있는 경우 */
    if(perm[9] != '\0'){
        for(int i = 9; i > 0; i--){
            if(perm[i] != '-'){
                bi |= cur;
            }
            cur <=< 1;
        }
    }
    /* 문자열 맨 앞에 파일 종류 정보가 없는 경우 */
    else{
        for(int i = 8; i > -1; i--){
            if(perm[i] != '-'){
                bi |= cur;
            }
            cur <=< 1;
        }
    }

    return bi;
}

```

```

int main(int argc, char *argv[]){
    /* 명령이 잘못된 경우 */
    if(argc != 3) {
        printf("Wrong input. Usage : %s <mode> <filename>\n", argv[0]);
        return 1;
    }
    /* Absolute mode인 경우 */
    int mode = 0b0;
    /* 문자열 permission이 입력된 경우 */
    if(argv[1][0] == '-' || argv[1][0] == 'r'){
        mode = lsoct(argv[1]);
        sleep(0.1);
        if (chmod(argv[2], mode)){
            return 2;
        }
        return 0;
    }
    /* 8진수 permission이 입력된 경우 */
    else if(argv[1][0] >= '0' && argv[1][0] < '8'){
        mode = strtol(argv[1], NULL, 8);
        if (chmod(argv[2], mode)){
            return 2;
        }
        return 0;
    }

    /* Symbolic mode인 경우 */

    /* file stat */
    if(stat(argv[2], &sb) == -1){
        perror("Error: ");
        return 3;
    }
    /* cur: 현재 파일의 권한 */
    int cur = sb.st_mode & (S_IRWXU | S_IRWXG | S_IRWXO);
    int target = 0b0; /* 권한 변경할 영역(user, group, other 관점) */
    int permission = 0b0; /* 권한 변경할 영역(read, write, exec 관점) */
    int op = 0; /* 연산자 종류 저장(1: '+', 2: '-', 3: '=') */
    int res = 0b0; /* Absolute mode */
    int i = 0; /* 현재 while문의 인덱스 */

```

```

/* 두 번째 인자 문자열의 각 문자 탐색 */
/* 두 번째 인자: [who][+|-|=][mode...] 구조 */
while(1){
    /* who 알아내기 */
    if(argv[1][i] == 'u')
        target += 0b111000000;
    else if(argv[1][i] == 'g')
        target += 0b000111000;
    else if(argv[1][i] == 'o')
        target += 0b000000111;
    else if(argv[1][i] == 'a')
        target += 0b111111111;
    /* 연산자 알아내기 */
    else if(argv[1][i] == '+')
        op = 1;
    else if(argv[1][i] == '-' && i > 0 && (argv[1][i-1] == 'u' || argv[1][i-1] ==
'g' || argv[1][i-1] == 'o' || argv[1][i-1] == 'a'))
        op = 2;
    else if(argv[1][i] == '=')
        op = 3;
    /* mode 알아내기 */
    else if(argv[1][i] == 'r')
        permission += 0b100100100;
    else if(argv[1][i] == 'w')
        permission += 0b010010010;
    else if(argv[1][i] == 'x')
        permission += 0b001001001;
    else if(argv[1][i] == '-' && op == 3)
        permission += 0b0;
}

```

:


```

/* 다른 옵션이 있거나 문자열이 끝난 경우 계산 후 권한 변경*/
else if(argv[1][i] == ',' || argv[1][i] == '\\0'){
    res = target & permission;
    /* +연산자인 경우 */
    if(op == 1){
        res |= cur;
    }
    /* -연산자인 경우 */
    else if(op == 2){
        res = cur - (res & cur);
    }
    /* =연산자인 경우 */
    else if(op == 3){
        res = res | (cur - (target & cur));
    }
    /* 권한 변경 */
    if (chmod(argv[2], res)){
        return 2;
    }

    /* 변수 초기화 */
    target = 0b0;
    permission = 0b0;
    op = 0;
    res = 0b0;
    /* file stat */
    if(stat(argv[2], &sb) == -1){
        perror("Error: ");
        return 3;
    }
    cur = sb.st_mode & (S_IRWXU | S_IRWXG | S_IRWXO);
    sleep(0.1);
}
/* 잘못된 입력이 있는 경우 */
else{
    printf("Error: Wrong input\n");
    return 4;
}
/* 문자열이 끝난 경우 루프 탈출 */
if(argv[1][i] == '\\0')
    break;
/* while 루프 인덱스 증가 */
i++;
}
/* success */
return 0;
}

```

- 예시파일(afile) 초기 권한

```

boo@boo-VirtualBox:~/Downloads$ ls -l
total 144
-rw-rw-r-- 1 boo boo    0 10월 12 17:41 afile

```

- 실행결과1: 8진수 absolute mode로 입력했을 때

```
boo@boo-VirtualBox:~/Downloads$ ./myChmod 0701 afile
boo@boo-VirtualBox:~/Downloads$ ls -l
total 144
-rwx-----x 1 boo boo    0 10월 12 17:41 afile
```

- 실행결과2: 문자열 absolute mode로 입력했을 때

```
boo@boo-VirtualBox:~/Downloads$ ./myChmod -rwxrwx--- afile
boo@boo-VirtualBox:~/Downloads$ ls -l
total 144
-rwxrwx--- 1 boo boo    0 10월 12 17:41 afile
```

- 실행결과3: symbolic mode로 입력했을 때1

```
boo@boo-VirtualBox:~/Downloads$ ./myChmod u=rw-,g-rx,o+x afile
boo@boo-VirtualBox:~/Downloads$ ls -l
total 144
-rw--w---x 1 boo boo    0 10월 12 17:41 afile
```

- 실행결과4: symbolic mode로 입력했을 때2

```
boo@boo-VirtualBox:~/Downloads$ ./myChmod a-w afile
boo@boo-VirtualBox:~/Downloads$ ls -l
total 144
-r-----x 1 boo boo    0 10월 12 17:41 afile
```