

2024.11.21	실습 과제	학번	22312072
과제 10	시스템 프로그래밍	이름	차민경
<p>• 참고사항</p> <p>모든 실습 과제는 각 문항에서 요구하는 문제의 명령어와 그 출력 결과를 동시에 기재하여야 합니다.</p> <p>예: 오늘 날짜를 출력하는 명령어를 쓰시오.</p> <p>답:</p> <p>date</p> <p>2024. 11. 21. (목) 15:00:00 KST</p>			

1. chap5 33p를 참고하여 "fchmod.c" 파일을 작성하고, 빈 텍스트 파일 "mod.txt"를 만들어 권한을 777로 설정하는 예시를 보이시오.

답:

```
[yu22312072@acslab-146:~/10$ touch mod.txt
[yu22312072@acslab-146:~/10$ ls -al mod.txt
-rw-rw-r-- 1 yu22312072 yu22312072 0 11월 21 15:17 mod.txt
[yu22312072@acslab-146:~/10$ ./fchmod 777 mod.txt
[yu22312072@acslab-146:~/10$ ls -al mod.txt
-rwxrwxrwx 1 yu22312072 yu22312072 0 11월 21 15:17 mod.txt
```

2. chap5 37p를 참고하여 "cptime.c" 파일을 작성하고, "mod2.txt"를 생성하여 [1]의 "mod.txt" 파일의 시간을 "mod2.txt"로 복사하시오.

답:

```
[yu22312072@acslab-146:~/10$ ls -al | grep .txt
-rw-rw-r-- 1 yu22312072 yu22312072 0 11월 21 15:24 mod2.txt
-rwxrwxrwx 1 yu22312072 yu22312072 0 11월 21 15:28 mod.txt
[yu22312072@acslab-146:~/10$ ./cptime mod.txt mod2.txt
[yu22312072@acslab-146:~/10$ ls -al | grep .txt
-rw-rw-r-- 1 yu22312072 yu22312072 0 11월 21 15:28 mod2.txt
-rwxrwxrwx 1 yu22312072 yu22312072 0 11월 21 15:28 mod.txt
```

3. chap5 45p를 참고하여 “list1.c” 파일을 작성하고, 실행하시오.

답:

```
[yu22312072@acslab-146:~/10$ ./list1
list1
pid
atexit.c
cptime.c
ftype
environ.c
args
ftype.c
cptime
example_fifo
mod.txt
mod2.txt
..
pid.c
example_link
list1.c
fchmod
atexit
printenv.c
.
args.c
fchmod.c
environ
```

4. 다음 조건을 만족하여 “list1.c”의 확장 기능을 추가한 “list2.c”를 작성하시오.

- 현재 디렉터리의 일반 파일에 대해서만 작업을 수행
- 파일의 권한을 777로 설정
- 파일의 시간 정보를 Unix Timestamp 0으로 설정
- 단, utimebuf 구조체를 사용하여 actime, modtime을 0으로 설정할 것

답:

```
yu22312072@acslab-146:~/10$ cat list2.c
#include <sys/types.h>
#include <sys/stat.h>
#include <dirent.h>
#include <stdio.h>
#include <stdlib.h>
#include <utime.h>

int main(int argc, char **argv){
    DIR *dp;
    char *dir;
    struct dirent *d;
    struct stat buf;
    struct utimbuf time;
    char path[BUFSIZ+1];

    if (argc == 1)
        dir = ".";
    else
        dir = argv[1];

    if ((dp = opendir(dir)) == NULL)
        perror(dir);

    while ((d = readdir(dp)) != NULL){
        sprintf(path, "%s/%s", dir, d->d_name);
        printf("%s\n", d->d_name);

        if(stat(path,&buf)<0)
            perror(path);

        if(!S_ISREG(buf.st_mode))
            continue;

        if(chmod(path, 0777) < 0){
            perror("chmode failed");
            continue;
        }

        time.actime = 0;
        time.modtime = 0;

        if(utime(path, &time) < 0)
            perror("utime failed");
    }
    closedir(dp);
    exit(0);
}
```

```
[yu22312072@acslab-146:~/10$ ls -l mod.txt
----- 1 yu22312072 yu22312072 0 11월 27 20:08 mod.txt
[yu22312072@acslab-146:~/10$ ./list2
list1
pid
atexit.c
list2.c
cptime.c
ftype
environ.c
args
ftype.c
cptime
example_fifo
mod.txt
mod2.txt
..
pid.c
example_link
list1.c
list2
fchmod
atexit
printenv.c
.
args.c
fchmod.c
environ
[yu22312072@acslab-146:~/10$ ls -l mod.txt
-rwxrwxrwx 1 yu22312072 yu22312072 0 1월 1 1970 mod.txt
```

5. chap5 64p를 참조하여 심볼릭 링크를 생성하는 “link.c”를 작성하고, 이를 이용하여 새 파일 “link.txt”의 심볼릭 링크 “sym.txt”를 만드시오. 이후, chap5 66p의 “rlink.c”의 결과를 보이시오.

답:

```
[yu22312072@acslab-146:~/10$ touch link.txt
[yu22312072@acslab-146:~/10$ ls -l link.txt
-rw-rw-r-- 1 yu22312072 yu22312072 0 11월 27 20:48 link.txt
[yu22312072@acslab-146:~/10$ ./link link.txt link_sym.txt
[yu22312072@acslab-146:~/10$ ls -l link.txt link_sym.txt
lrwxrwxrwx 1 yu22312072 yu22312072 8 11월 27 20:48 link_sym.txt -> link.txt
-rw-rw-r-- 1 yu22312072 yu22312072 0 11월 27 20:48 link.txt
[yu22312072@acslab-146:~/10$ ./rlink link_sym.txt
link.txtyu22312072@acslab-146:~/10$ █
```

6. “setenv” 함수를 사용하여 “HELLO”라는 이름의 환경 변수에 “LINUX” 값을 저장하고, “getenv” 함수를 이용해 “HELLO” 환경 변수의 값을 출력하는 C 프로그램을 작성 후, 실행하시오.

답:

```
[yu22312072@acslab-146:~/10$ cat setenv.c
#include <stdio.h>
#include <stdlib.h>

int main(void){
    char *ptr;

    setenv("HELLO", "LINUX", 0);
    ptr = getenv("HELLO");
    printf("HELLO = %s\n", ptr);
}
[yu22312072@acslab-146:~/10$ ./setenv
HELLO = LINUX
```

7. chap8 19p를 참고하여 “atexit.c”를 컴파일하고, 결과를 출력하시오.

답:

```
[yu22312072@acslab-146:~/10$ ./atexit
main 끝
두 번째 exit처리 기
첫 번째 exit처리 기
```

8. chap8 24p를 참고하여 “pid.c”를 컴파일하고, 실행하여 “bash” 프로세스 번호를 보이시오.

답:

```
[yu22312072@acslab-146:~/10$ bash
[yu22312072@acslab-146:~/10$ ./pid
나의 프로세스 번호 : [78013]
내 부모 프로세스 번호 : [77941]
```

9. chap8 28p를 참고하여, 다음과 같이 uid, euid를 출력하는 C 프로그램을 작성 후, 실행 하시오.

```
$ ./uid
Real UID: 1001
Effective UID: 1001
```

답:

```
[yu22312072@acslab-146:~/10$ ./uid
Real UID:1098
Effective UID:1098
```

10. chap8 39p를 참고하여, 스택 영역에 저장되는 변수는 "stack"으로, 힙 영역에 저장되는 변수는 "heap"으로 선언한 C 소스 코드를 만드시오.

답:

```
[yu22312072@acslab-146:~/10$ cat heap.c
#include <stdio.h>
#include <stdlib.h>

int main(){
    int stack = 10;

    int *heap = (int *)malloc (sizeof(int));
    *heap = 20;

    printf("heap : %d\n", stack);
    printf("stack : %d\n", *heap);

    free(heap);
}

[yu22312072@acslab-146:~/10$ ./heap
heap : 10
stack : 20
```