
Server Programing

Week3



학 과: 소프트웨어학과

학 번: 2020039082

이 름: 김도현

Practice01

```
user2020039082@swist2: ~/w × + ▾  
user2020039082@swist2:~/week3-1$ ./pr03_01.out  
user2020039082@swist2:~/week3-1$ |
```

- fp=fopen("test.txt", "w"); 문제조건에 맞게 w로 접근하도록 변경했고, fp=null인 경우 if문 안으로 들어가 errno를 출력하게 된다.
- 이후 if문 안에서는 exit(1)로 탈출, 파일포인터를 fclose로 닫는다.

Practice02

- 결과화면

```
user2020039082@swist2: ~/w × + ▾  
user2020039082@swist2:~/week3-1$ ./pr03_02.out  
missing.txt: Permission denied  
user2020039082@swist2:~/week3-1$ |
```

- 추가 설명 화면

```
user2020039082@swist2:~/week3-1$ vim pr03_02.c  
user2020039082@swist2:~/week3-1$ touch missing.txt  
user2020039082@swist2:~/week3-1$ chmod -x missing.txt  
user2020039082@swist2:~/week3-1$ ls -al  
total 48  
drwxrwxr-x 2 user2020039082 user2020039082 4096 Sep 19 16:15 .  
drwxr-x--- 6 user2020039082 user2020039082 4096 Sep 19 16:15 ..  
-rwxrwxr-x 1 user2020039082 user2020039082 16040 Sep 19 16:11 a.out  
-rw-rw-r-- 1 user2020039082 user2020039082 0 Sep 19 16:15 missing.txt  
-rw-rw-r-- 1 user2020039082 user2020039082 226 Sep 19 16:10 pr03_01.c  
-rw-rw-r-- 1 user2020039082 user2020039082 192 Sep 19 16:15 pr03_02.c  
-rwxrwxr-x 1 user2020039082 user2020039082 16040 Sep 19 16:13 pr03_02.out  
user2020039082@swist2:~/week3-1$ |
```

- touch로 missing.txt를 만들어서 실행권한을 지웠다.
- 이후 pr03_02.out을 실행하게 되면, access("missing.txt", X_OK)코드를 통해 missing.txt파일에 접근하게 된다. 하지만 실행권한이 없기 때문에 Permission denied가 출력되게 된다.
-

Practice03

```
user2020039082@swist2: ~/w × + v
user2020039082@swist2:~/week3-1$ chmod 444 missing.txt
user2020039082@swist2:~/week3-1$ ./pr03_03.out
error: Permission denied(missing.txt)
user2020039082@swist2:~/week3-1$ echo $?
13
user2020039082@swist2:~/week3-1$ |
```

- Chmod 444는 읽기 권한을 부여하는 것이므로 쓰기권한으로 접근하기 때문에 permission denied가 발생하게 된다.
- (fp=fopen("missing.txt", "w")) == NULL를 조건으로 if문에 진입하게 되고, 쓰기 권한이 없기 때문에 printf("error: %s(missing.txt)\n", err);를 출력하게 된다.

Practice04

```
user2020039082@swist2: ~/w × + v
user2020039082@swist2:~/week3-1$ gcc -o pr03_04.out pr03_04.c
user2020039082@swist2:~/week3-1$ ./pr03_04.out -f
./pr03_04.out: invalid option -- 'f'
./pr03_04.out [option] int m int n
user2020039082@swist2:~/week3-1$ ./pr03_04.out -g
./pr03_04.out: option requires an argument -- 'g'
./pr03_04.out [option] int m int n
user2020039082@swist2:~/week3-1$ ./pr03_04.out -g 7 5
1
user2020039082@swist2:~/week3-1$ ./pr03_04.out -n 7 5
35
user2020039082@swist2:~/week3-1$ ./pr03_04.out -t 7 5
35
user2020039082@swist2:~/week3-1$ ./pr03_04.out -p 7 5
12
user2020039082@swist2:~/week3-1$ |
```

- gcd함수는 유클리드 호제법을 이용하여 m자리에는 n, n자리에는 m%n을 넣어 구성하였고,
- lcm의 경우 최소 공배수로 최대 공약수 * m * n 을 하면 되기에 return (m * n) / gcd(m, n);

로 구성하였다.

- (n = getopt(argc, argv, "g:n:t:p:")) != -1 if문 조건은 왼쪽과 같이 작성하였고, 옵션은 g n t p로 설정하여, gcd, lcm, 곱셈, 덧셈을 수행할 수 있도록 해주었다.

Practice05

```
user2020039082@swist2: ~/w  X + v
user2020039082@swist2:~/week3-1$ gcc -o pr03_05.out pr03_05.c
user2020039082@swist2:~/week3-1$ ./pr03_05.out
>> 3
dobby
is
free
>> 4
user2020039082@swist2:~/week3-1$ |
```

- Count_vowel함수에서 for문으로 한 글자씩 ch로 두어 'a'인지, 'e'인지, 'i'인지, 'o'인지, 'u'인지 검사하여 한 문장에 모음이 몇 개 있는지 result에 저장해두어 return한다.
- main코드에서는 for문으로 n값만큼 calloc을 선언하여 메모리 동적할당을 해주고, 각 문장에서 모음이 몇 개이었는지 count_vowel함수를 호출하여 main문 안에 있는 result변수에 값을 저장한다. 이후 n번만큼 for문으로 동적할당을 해제해준다.

Practice06

```
user2020039082@swist2: ~/w  X + v
user2020039082@swist2:~/week3-1$ gcc -o pr03_06.out pr03_06.c
user2020039082@swist2:~/week3-1$ ./pr03_06.out
110
user2020039082@swist2:~/week3-1$ |
```

- 동적할당한 변수를 해제한 이후에 사용하면 안되기 때문에, free문을 맨 아래로 내려주었다.

Practice07

```
user2020039082@swist2: ~/w X + v
user2020039082@swist2:~/week3-1$ gcc -o pr03_07.out pr03_07.c
user2020039082@swist2:~/week3-1$ ./pr03_07.out
3
6
7
5
3
The first value: 3
user2020039082@swist2:~/week3-1$ |
```

- Printf와 마지막 free에서 array처음을 가리킬 수 있도록 (array-size)로 변경함으로써 세그먼트 오류를 피할 수 있다.

Practice08

```
user2020039082@swist2: ~/w X + v
user2020039082@swist2:~/week3-1$ gcc -o pr03_08.out pr03_08.c
user2020039082@swist2:~/week3-1$ ./pr03_08.out
SilverProgramming
user2020039082@swist2:~/week3-1$ |
```

- Char str*;로 선언하면, 문자열 리터럴을 가리키기 때문에 값을 수정하면 오류가 발생할 수 있다. (읽기 전용 메모리에서 시도하기 때문에)
- 리터럴 배열로 바꾸면 읽기/쓰기가 가능한 메모리에 할당이 되기 때문에, 코드를 char str[] = "ServerProgramming";로 바꾸어 주었다.