

13주차 과제					
학년	2	학번	2016707079	이름	하상천

## 1. 과제 설명

파일 입출력을 활용하여 메모장에 현재 날짜와 시간을 받아 txt 파일로 저장하게 하고, 저장된 시간과 현재시간이 업데이트 된 시간과 날짜를 출력하는 프로그램을 만드세요.

## 2. 과제 이론

### If문

조건을 주어 그 조건을 만족할 때에 실행하도록 한다.

Ex) if(i==1) 이면 i의 값이 1일 때만 실행하도록 한다.

그 외에는 다른 실행을 하고 싶다면 똑같은 형식으로 else if 또는 else로 조건을 주면 된다.

### feof함수

파일에서 입출력이 일어나는 위치가 파일의 끝인지를 검사하는 함수이다.

끝이면 0이 아닌 다른 값, 끝이 아니면 0을 출력한다.

Char \*fgets(char \*s, int n, FILE \*fp)

fgets()는 fp에서 문자들을 읽어서 s가 가리키는 위치에 저장한다. 하나의 라인이 완전히

읽히거나 n-1개의 문자들이 읽히면 종료된다. s는 입력 문자열이 저장되는 버퍼이고

n은 입력 가능한 최대 문자 개수이다. fgets()에서 자동적으로 문자열의 끝에 널문자가

삽입되기 때문에 널문자를 붙일 필요가 없다.

## 파일모드

### 파일의 접근 모드 (r, w, a, r+, w+, a+)

모드	설명
<b>r</b>	<ul style="list-style-type: none"> <li>읽기(read) 전용으로 파일을 엽니다.</li> <li>파일이 없거나 찾을 수 없는 경우에 호출 실패</li> </ul>
<b>w</b>	<ul style="list-style-type: none"> <li>쓰기(write) 전용으로 파일을 엽니다.</li> <li>지정한 파일명이 있는 경우: 파일 내용을 모두 지우고 새로 만듭니다.</li> <li>지정한 파일명이 없는 경우: 새로운 파일을 생성 합니다.</li> </ul>
<b>a</b>	<ul style="list-style-type: none"> <li>추가(append) 쓰기 전용으로 파일을 엽니다.</li> <li>지정한 파일이 있으면 파일의 끝에서부터 내용을 추가합니다.</li> </ul>
<b>r+</b>	<ul style="list-style-type: none"> <li>파일을 읽고 쓰기 위해 엽니다.</li> <li>지정한 파일이 있는 경우: 기존의 내용을 덮어씁니다.</li> <li>지정한 파일이 없는 경우: 새로운 파일을 생성해서 데이터를 씁니다(저장).</li> </ul>
<b>w+</b>	<ul style="list-style-type: none"> <li>파일을 읽고 쓰기 위해 엽니다.</li> <li>지정한 파일이 있는 경우: 파일의 내용을 모두 지우고 새 파일을 만듭니다.</li> <li>지정한 파일이 없는 경우: 새로운 파일을 생성 한다.</li> </ul>
<b>a+</b>	<ul style="list-style-type: none"> <li>파일을 읽고 추가 쓰기 위해 엽니다.</li> <li>지정한 파일이 있으면 파일의 끝에서부터 내용을 추가합니다.</li> <li>나머지 기능은 r+와 같습니다.</li> </ul>

### 파일 입출력 모드

모드	설명
<b>t</b>	텍스트 파일 모드 (text file mode)
<b>b</b>	바이너리 파일 모드 (binary file mode)

파일 접근 모드		입출력 모드		파일 오픈 모드	
r				rt	rb
w				wt	wb
a				at	ab
r+	+	t	=	r+t	r+b
w+		b		w+t	w+b
a+				a+t	a+b

### 3. 주요 소스 설명

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

```
int main()
```

```
{
```

```
    char str1[30]; //배열 선언
```

```
    FILE * rp = fopen("Hasangcheon.txt", "rt"); //파일을 읽기 모드로 연다.
```

```
    if (rp == NULL) //파일 열기를 실패한 경우
```

```
    {
```

```

        puts("Fail to Open the file");

        return -1; //파일의 끝을 표현하기 위한 상수(EOF)
    }

    else {

        puts("Success to open the file");

    }

    while (!feof(rp)) // 마지막에 있는 문자열만 str1에 입력. 파일에서 입출력이 일어나는 위치가 파일의 끝이면 1을 출력하는 feof함수
    {

        fgets(str1, sizeof(str1), rp);

    }

    printf("%s", str1); //str1 출력

    fclose(rp); //열린 파일을 닫는 함수 fclose
}

```

#### 4. 실행화면

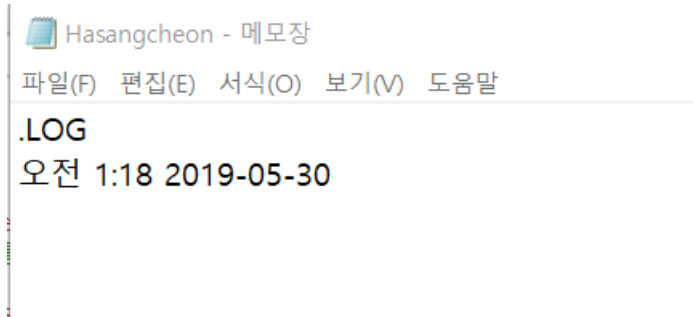
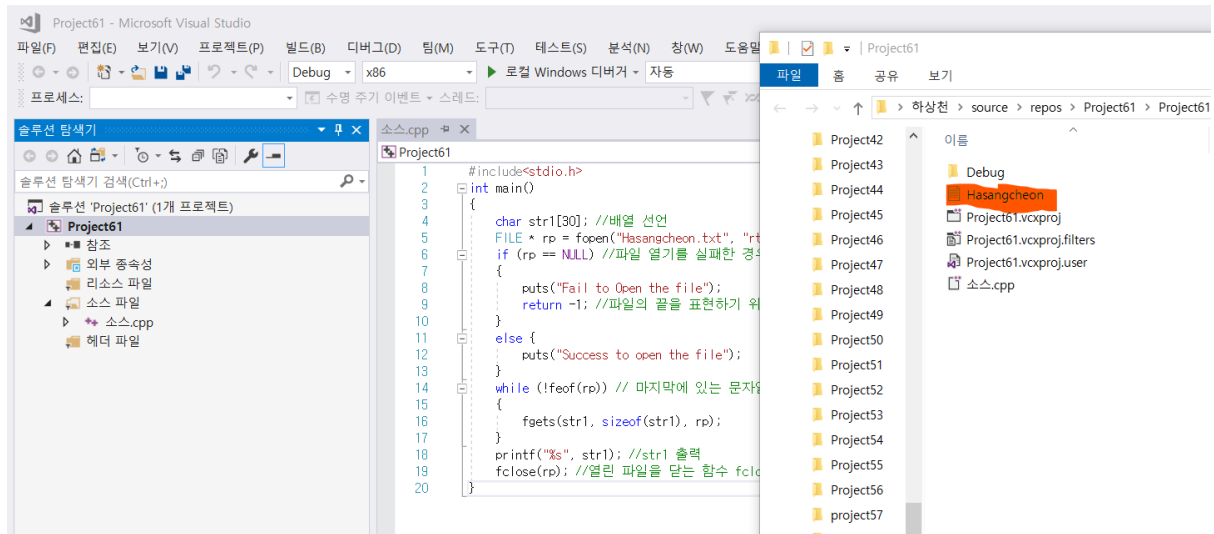
 Microsoft Visual Studio 디버그 콘솔

```

Success to open the file
오전 1:18 2019-05-30

C:\Users\seomk\source\repos\Project61\
이 창을 닫으려면 아무 키나 누르세요.

```



## 5. 고찰

이번주 과제는 파일 입출력에 대한 과제였다. 오랜만에 보는 내용이라 많이 까먹어서 책을 펴고 공부를 하고 과제를 풀었다. fopen함수를 이용해서 텍스트 파일 읽기 전용으로 파일을 엽니다. fopen함수는 오류발생시 NULL을 반환한다. 그래서 if문을 통해 오류가 발생하면 프로그램을 종료하도록 만들었다. 또한 파일의 끝에 도달했는지 여부를 확인하기 위해 feof함수를 사용했다. 파일의 끝에 도달하게 되면 0이 아닌 값을 반환하고, 파일의 끝에 도달하지 못했을 경우에 0을 반환한다. 문자열을 입력하기 위해서 fgets()를 사용했다. 처음에 fgets()를 잘 알지 못해서 인터넷을 찾아보았더니 이러한 특징이 있다는 것을 알게 되었다. char \*fgets(char \*string, int n, FILE \*stream)에서 어느 것이 먼저 오건 첫 번째 줄 바꾸기 문자(wn)가 나올 때까지, 스트림의 끝까지 또는 읽은 문자 수가 n-1과 같을 때까지 문자를 읽는다. 그리고 fgets() 함수는 결과를 string에 저장하고 스트림 끝에 널(null) 문자(w0)를 추가한다. fclose 함수를 이용하여 열린 파일을 닫았다. 엄청 자세하게 배우지는 않았지만 그래도 C언어를 처음부터 끝까지 배운 것 같아서 뿌듯하다. 복습도 꾸준히 해서 까먹지 않도록 노력해야겠다.