

CSED101. Programming & Problem solving

Spring, 2019

Programming Assignment #4 (70 points)

고병현(briango28@postech.ac.kr)

■ **Due:** 2019.05.18 23:59

■ **Development Environment:** Windows Visual Studio 2017

■ 제출물

- **C Code 압축 파일** (assn4.zip)
 - 파일 목록: `mystring.h`, `mystring.c`, `assn4.c`
 - 프로그램의 소스 코드를 이해하기 쉽도록 반드시 주석을 붙일 것
- **보고서 파일** (.docx or .hwp) 예) `assn4.docx` 또는 `assn4.hwp`
 - `AssnReadMe.pdf` 를 참조하여 작성할 것.
 - 보고서는 Problem2에 대해서만 작성 할 것.
 - **명예 서약(Honor code):** 표지에 다음의 내용을 포함한다. “나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.” 보고서 표지에 명예 서약이 없는 경우는 과제를 제출하지 않은 것으로 처리한다.
 - 보고서에는 개발 환경 (운영체제 및 컴파일러)를 명시하도록 한다.
- 소스코드와 보고서 파일을 LMS를 이용하여 제출한다.

■ 주의사항

- 각 문제에 해당하는 요구사항을 반드시 지킬 것.
- 모든 문제의 출력 형식은 아래의 예시들과 동일해야 하며, 같지 않을 시는 감점이 된다.
- 컴파일 혹은 실행이 안되면 무조건 0점 처리된다.
- 하루 late시 20%가 감점되며, 3일 이상 지나면 받지 않는다. (0점 처리)
- 부정행위에 관한 규정은 POSTECH 전자컴퓨터공학부 학부위원회의 ‘POSTECH 전자컴퓨터공학부 부정행위 정의’를 따른다. (LMS의 과목 공지사항의 제목 [document about cheating]의 첨부파일인 `disciplinary.pdf`를 참조할 것.)
- 이번 과제에서는 추가 기능 구현에 대한 추가 점수는 없습니다.
- 과제 작성시 전역 변수, 구조체 및 연결 리스트(linked list)를 사용할 수 없습니다.
- 요구되는 기능을 사용자 정의 함수로 적절히 분리하여 코드를 작성하여야 합니다. (main() 함수만 사용하는 경우 감점이 됩니다.)

■ Problem 1: 문자열 처리 함수 (5점)

(문제)

C 언어에서 제공되는 문자열 함수는 `<string.h>` 라이브러리에서 제공된다. 문자열 라이브러리에 포함되어 있는 함수 중 일부를 직접 작성해 보자.

(설명)

제공된 assn4_p1.zip 의 압축을 풀면 mystring.h와 mystring.c가 주어진다.

mystring.h 는 아래와 같은 문자열 처리 함수의 선언을 포함하고 있다. (그대로 사용하고 변경하지 말 것)

```
#pragma once

int mystrlen(char *str);
char *mystrcpy(char *toStr, char *fromStr);
int mystrcmp(char *str1, char *str2);
char *mystrcat(char *dest, char *src);
char *mystrchr(char *str, char c);
```

위 함수의 구현부를 포함한 mystring.c 를 작성하라. 각 함수의 정의는 다음과 같다.

(1) int mystrlen(char *str)

NULL 문자를 제외한 문자열의 길이를 반환한다. 빈 문자열의 경우 0 을 반환한다.

예제)

```
printf("%d\n", mystrlen("cs101")); // 결과: 5
```

(2) char *mystrcpy(char *toStr, char *fromStr)

문자열 복사 함수로 NULL 문자를 포함한 문자열 fromStr 를 문자열 toStr 에 복사한 후, 문자열 toStr 의 시작 주소를 반환한다.

예제)

```
char str[256];
printf("%s\n", mystrcpy(str, "Good Day")); // 결과: Good Day
printf("%s\n", mystrcpy(str, "Hello"));    // 결과: Hello
```

(3) int mystrcmp(char *str1, char *str2)

문자열 str1 과 str2 의 대소를 비교한다(대소문자 구분). 비교기준은 아스키코드표의 값을 기준으로 한다.

각 문자열의 첫 번째 문자부터 비교를 시작한다. 만일 같다면 두 문자가 다를 때까지나 NULL 에 도달할 때까지 계속 비교를 수행한다.

- 비교 중 str1 의 문자가 작을 경우 -1, 클 경우 1 을 반환한다.
- 문자열이 길이가 같고 모든 문자가 같을 경우, 0 을 반환한다.
- 비교 중 하나의 문자열이 먼저 끝에 도달할 경우, 끝난 문자열을 작다고 판단한다.

예제)

```
printf("%d\n", mystrcmp("csed101", "csed103")); // 결과: -1
printf("%d\n", mystrcmp("csed", "Csed"));        // 결과: 1
printf("%d\n", mystrcmp("csed", "cse"));         // 결과: 1
printf("%d\n", mystrcmp("csed", "csed103"));     // 결과: -1
```

(4) char *mystrcat(char *dest, char *src)

문자열 연결 함수로 dest의 끝에 src를 이어 붙인다. 즉, 문자열 dest 뒤의 NULL 문자는 src의 첫 번째 문자로 덮어 씌워지고 src의 NULL 문자는 남는다. 문자열 연결 후, 문자열 dest의 시작 주소를 반환한다.

예제)

```
char str[256] = "Hello";
mystrcat(str, ", World");
printf("%s\n", str); // 결과: Hello, World
```

(5) char* mystrrchr(char *str, char c)

문자열 str 을 검색하여 마지막으로 발견된 문자 c 의 위치 주소를 반환한다. 즉, 문자열 str 의 뒤에서부터 검색하여 처음으로 c와 일치하는 문자의 위치 주소를 반환한다. 만약 문자열 str 에서 일치하는 것이 없으면 NULL 을 반환한다.

예제)

```
printf("%s\n", mystrrchr("banana", 'n'));        //결과: na
printf("%s\n", mystrrchr ("hello world", 'l'));  //결과: ld
```

■ Problem 2: Text File Viewer (65점)

(문제)

영문 텍스트 파일을 읽고 보여주는 간단한 텍스트 파일 뷰어를 작성하자.

(목적)

- 포인터와 동적 할당, 텍스트 파일의 사용법을 익힌다.

(설명 및 요구사항)

- 파일 이름은 확장자를 포함하여 "assn4.c"로 한다.
- 표준 헤더 파일 `<string.h>`를 `#include` 하여 사용할 수 있다.
- 이번 숙제에서 구현해야 할 텍스트 파일 뷰어 프로그램의 기능은 다음과 같다.
 - 페이지 단위로 파일을 출력한다.
 - 실행 중에 텍스트 파일을 닫고 다른 텍스트 파일을 읽을 수 있다.
 - 긴 줄은 60자 단위로 끊어서 출력한다.
 - 한 단어가 여러 줄에 걸쳐서 출력되지 않도록 하는 워드 랩 (word wrap) 기능을 구현한다.

(1) 초기 화면 및 파일 읽기

- 프로그램을 실행시키면 아래와 같이 파일 이름을 입력 받기 위한 화면이 출력된 후, 사용자 입력을 기다린다. (노란색으로 표시된 글자는 사용자 입력에 해당한다.)

```
* Text File Viewer *
filename: sample1.txt
```

- 이 때, 뷰어를 통하여 출력할 파일을 (e.g. "`sample1.txt`";) 입력 받고, 입력 받은 파일을 `fopen()` 함수의 "`r`" 모드 (읽기 모드)를 사용하여 연다. 파일 이름은 공백을 포함하지 않으며, 30자를 넘지 않는다고 간주한다.
- 만약 파일을 여는 데에 실패하면 다음과 같은 오류 메시지를 출력한 뒤, 사용자가 Enter 키를 입력하면 화면을 지우고 초기 화면을 다시 출력한다. (`getchar()` 함수 사용)

```
* Text File Viewer *
filename: sample1.txt
Unable to open sample1.txt
Press Enter to continue...
```

- 파일을 여는 데에 성공하는 경우, 파일의 내용을 모두 읽어서 아래와 같이 첫 **10줄**을 (한 줄에 최대 **60글자**) 한 페이지에 출력한다.
 - 아래의 내용 출력하기 전에, 출력 화면을 모두 지운다.
 - 텍스트 데이터는 char* 형의 배열을 이용하여 관리하자. 9쪽의 메모리 할당에 설명되어 있는 방법대로 구현한다.
 - 내용물을 모두 읽은 파일은 fclose() 함수로 닫는다.

```
Avengers: Endgame is a 2019 American superhero film based on
the Marvel Comics superhero team the Avengers, produced by
Marvel Studios and distributed by Walt Disney Studios Motion
Pictures. It is the sequel to 2012's The Avengers, 2015's
Avengers: Age of Ultron and 2018's Avengers: Infinity War,
and the 22nd film in the Marvel Cinematic Universe (MCU),
serving as the conclusion to the Infinity Saga. The film is
directed by Anthony and Joe Russo, written by Christopher
Markus and Stephen McFeely and features an ensemble cast
including Robert Downey Jr., Chris Evans, Mark Ruffalo,
=====[1, 10]
f, b, o, q> f
```

- 그 아래에는 출력의 끝을 표시하는 '='문자 50개와 함께 현재 화면에 출력된 페이지의 줄 번호 정보를 출력한다. [1, 10]은 1번 줄에서 10번 줄까지 출력되었음을 의미한다.
- 그 아래 "f, b, o, q>"가 출력되고 사용자 명령어를 기다린다.
- **f**는 forward, **b**는 backward, **o**는 open file, **q**는 quit 을 의미하며 각 기능에 대한 설명은 아래와 같다.
- 위의 네 가지 명령 이외의 입력을 받으면, 즉시 화면을 지우고 현재 화면을 다시 출력한다.

(2) f: forward

- 이전 출력 예시 화면에서 'f' 입력 시, 화면을 지운 뒤 다음 페이지 (e.g. 11 – 20번 줄)를 아래 화면과 같이 출력한다.
- 이 때, 한 단어가 두 줄에 걸쳐서 출력되는 일이 없도록 각 줄의 끝을 잘 조절하여 출력한다.
 - 단어는 문장부호를 무시하고 무조건 띄어쓰기를 기준으로 구분한다.
 - 워드 랩 기능 구현은 문서의 마지막 쪽을 참고하여 구현한다.

```
Chris Hemsworth, Scarlett Johansson, Jeremy Renner, Don
Cheadle, Paul Rudd, Brie Larson, Karen Gillan, Danai Gurira,
Bradley Cooper, and Josh Brolin. In the film, the surviving
members of the Avengers and their allies travel through the
quantum realm to recover the Infinity Stones and reverse the
damage caused by Thanos in Infinity War.
```

```
The film was announced in October 2014 as Avengers: Infinity
War - Part 2. The Russo brothers came on board to direct in
April 2015, and by May, Markus and McFeely signed on to
=====[11, 20]
f, b, o, q> b
```

(3) b: backward

- 이전 출력 예시에서 'b' 입력 시, 화면을 지운 후 이전 페이지를 아래 화면과 같이 출력한다.

```
Avengers: Endgame is a 2019 American superhero film based on
the Marvel Comics superhero team the Avengers, produced by
Marvel Studios and distributed by Walt Disney Studios Motion
Pictures. It is the sequel to 2012's The Avengers, 2015's
Avengers: Age of Ultron and 2018's Avengers: Infinity War,
and the 22nd film in the Marvel Cinematic Universe (MCU),
serving as the conclusion to the Infinity Saga. The film is
directed by Anthony and Joe Russo, written by Christopher
Markus and Stephen McFeely and features an ensemble cast
including Robert Downey Jr., Chris Evans, Mark Ruffalo,
=====[1, 10]
f, b, o, q>
```

(4) o: open file

- 'o' 입력 시, 새로운 파일을 읽기 위해 아래와 같이 "filename: "이 출력되고 사용자 입력을 기다린다. 파일을 여는 데에 성공하면 기존의 동적 할당된 메모리를 전부 할당 해제한다.

```
Avengers: Endgame is a 2019 American superhero film based on
the Marvel Comics superhero team the Avengers, produced by
Marvel Studios and distributed by Walt Disney Studios Motion
Pictures. It is the sequel to 2012's The Avengers, 2015's
Avengers: Age of Ultron and 2018's Avengers: Infinity War,
and the 22nd film in the Marvel Cinematic Universe (MCU),
serving as the conclusion to the Infinity Saga. The film is
directed by Anthony and Joe Russo, written by Christopher
Markus and Stephen McFeely and features an ensemble cast
including Robert Downey Jr., Chris Evans, Mark Ruffalo,
=====[1, 10]
f, b, o, q> o
filename: sample2.txt
```

- 파일을 여는 데에 실패하면 기존 할당 메모리를 할당 해제하지 않고, 아래 화면과 같은 에러 메시지를 출력한다. 사용자가 Enter 를 입력 하면 화면을 지우고 기존에 열람하던 텍스트를 다시 출력한다.

```
Avengers: Endgame is a 2019 American superhero film based on
the Marvel Comics superhero team the Avengers, produced by
Marvel Studios and distributed by Walt Disney Studios Motion
Pictures. It is the sequel to 2012's The Avengers, 2015's
Avengers: Age of Ultron and 2018's Avengers: Infinity War,
and the 22nd film in the Marvel Cinematic Universe (MCU),
serving as the conclusion to the Infinity Saga. The film is
directed by Anthony and Joe Russo, written by Christopher
Markus and Stephen McFeely and features an ensemble cast
including Robert Downey Jr., Chris Evans, Mark Ruffalo,
=====[1, 10]
f, b, o, q> o
```

```
filename: sample2.txt
Unable to open sample2.txt
Press Enter to continue...
```

- "sample2.txt" 파일을 여는 데에 성공하면 아래와 같이 텍스트 파일의 내용이 출력된다.

```
The Last Question by Isaac Asimov 1956

The last question was asked for the first time, half in
jest, on May 21, 2061, at a time when humanity first stepped
into the light. The question came about as a result of a
five dollar bet over highballs, and it happened this way:

Alexander Adell and Bertram Lupov were two of the faithful
attendants of Multivac. As well as any human beings could,
they knew what lay behind the cold, clicking, flashing face
===== [1, 10]
f, b, o, q>
```

- 아래는 위 화면에서 'f'를 여러 번 반복해서 입력한 화면으로 sample2.txt 의 마지막 페이지 (701 - 709번 줄)를 보여준다.

```
this. Carefully, AC organized the program.

The consciousness of AC encompassed all of what had once
been a Universe and brooded over what was now Chaos. Step by
step, it must be done.

And AC said, "LET THERE BE LIGHT!"

And there was light----

===== [701, 709]
f, b, o, q>
```

(5) q: quit

- 'q' 입력 시, 동적 할당된 메모리를 전부 할당 해제 하고 프로그램을 종료한다.

```
this. Carefully, AC organized the program.

The consciousness of AC encompassed all of what had once
been a Universe and brooded over what was now Chaos. Step by
step, it must be done.

And AC said, "LET THERE BE LIGHT!"

And there was light----

===== [701, 709]
```

```
f, b, o, q> q
```

(6) 예외 처리

- 파일 길이는 1,000줄을 넘어가지 않는다고 가정한다.
- 모든 경우에, 해당 페이지에 출력할 줄이 10줄이 안 된다면, 출력할 수 있는 만큼 출력하고 나머지 줄은 비워둔다. 이 때, 화면 우측 하단의 줄 번호는 실제로 출력한 줄에 대해서 표기한다.
- 'f' 혹은 'b'를 입력하였지만 더 이상 이전 줄 혹은 다음 줄이 없는 경우, 에러 메시지 없이 즉시 화면을 지우고 현재 화면을 다시 출력한다.

(주의 사항)

<string.h> 헤더의 문자열 관련 함수를 사용할 때에는 인수로 주어지는 문자열의 범위가 겹치지 않도록 주의하자. 겹치는 경우에는 정상 동작이 보장되지 않는다.

부록

■ 공백을 포함한 문자열 입력 받기

파일로부터 공백을 포함한 임의의 길이의 문자열을 입력 받기 위해서는 `<stdio.h>` 헤더의 `fgets()` 사용이 권장된다. 함수의 원형은 다음과 같다.

```
char* fgets(char* str, int count, FILE* stream);
```

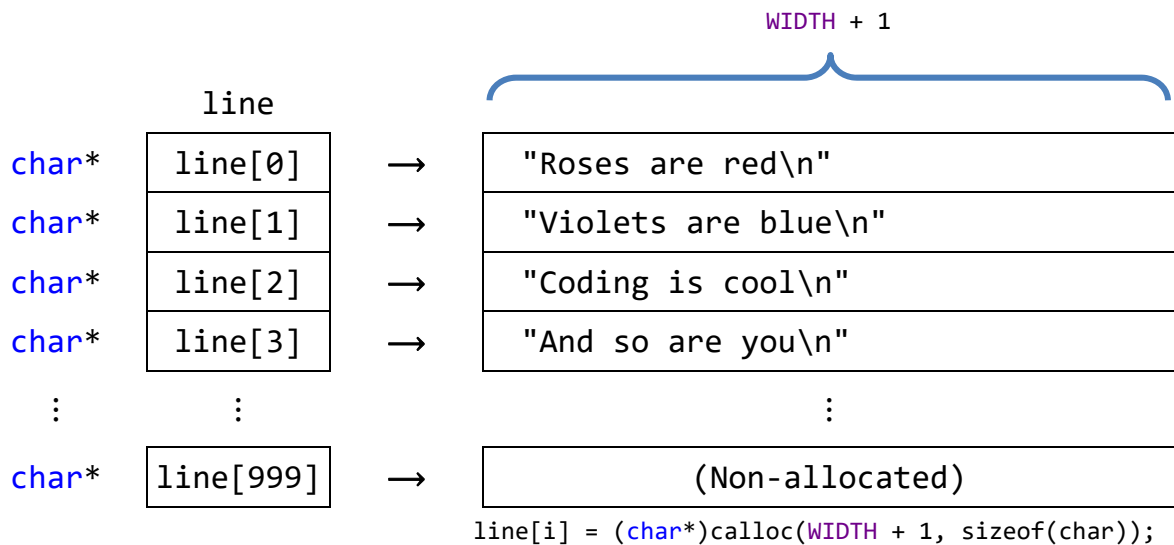
`fgets()`는 `stream`으로부터 문자를 최대 `count - 1` 개까지 읽어 `str`에 null 문자 `'\0'`를 첨가하여 저장한다. 만약 문자를 `count - 1` 개 읽기 전에 줄바꿈 문자 `'\n'`가 등장하면, 즉시 읽기를 중단하고 지금까지의 결과에 null 문자 `'\0'`를 첨가한다.

`fgets()`는 한 문자라도 읽는 데에 성공하면 `str`를, 읽기에 실패하거나 파일의 끝에 다다른면 `NULL` 포인터를 반환한다.

■ 동적 메모리 할당

파일의 텍스트를 저장하기 위해 필요한 줄 수는 읽어보기 전에는 미리 알 수 없다. 불필요한 공간을 할당하지 않도록 효율적인 메모리 사용을 위해, 다음과 같은 메모리 구조를 사용한다.

```
#define WIDTH 60
#define MAX_LINES 1000
...
char* line[MAX_LINES];
...
```



각 줄은 길이 `WIDTH`의 문자열을 저장하기에 적절한 양의 메모리를 그때그때 할당하여 사용한다.

■ 워드 랩(word wrap) 기능

예를 들어, 다음 문단이 모두 한 줄로 이어져 있다고 하자.

Avengers: Endgame is a 2019 American superhero film based on the Marvel Comics superhero team the Avengers, produced by Marvel Studios and distributed by Walt Disney Studios Motion Pictures. It is the sequel to 2012's The Avengers, 2015's Avengers: Age of Ultron and 2018's Avengers: Infinity War, and the 22nd film in the Marvel Cinematic Universe (MCU), serving as the conclusion to the Infinity Saga.

이 줄을 단순히 60글자씩 끊어서 출력하면 화면에는 다음과 같이 나타난다. 일부 단어는 여러 줄에 걸쳐서 출력되고, 줄 맨 앞에 띄어쓰기가 오기도 한다. (아래의 빨간색 표시는 중간에 조각나는 단어에 해당)

Avengers: Endgame is a 2019 American superhero film based on the Marvel Comics superhero team the Avengers, produced by Marvel Studios and distributed by Walt Disney Studios Motion Pictures. It is the sequel to 2012's The Avengers, 2015's **A**
vengers: Age of Ultron and 2018's Avengers: Infinity War, **an**
d the 22nd film in the Marvel Cinematic Universe (MCU), **serv**
ing as the conclusion to the Infinity Saga.

이렇게 단어가 중간에 조각나거나 줄이 공백 문자로 시작하지 않도록 워드 랩 기능을 구현하자. 다음과 같은 문장을 fgets(str, WIDTH + 1, stream)으로 읽게 되면,

Pictures. It is the sequel to 2012's The Avengers, 2015's **Avengers**: ...

60자까지만 읽기 때문에 아래와 같이 잘린다.

Pictures. It is the sequel to 2012's The Avengers, 2015's **Av**engers: ...

워드 랩이 제대로 되었다면 아래와 같이 잘려야 한다.

Pictures. It is the sequel to 2012's The Avengers, 2015's **Avengers**: ...

파일에서 읽을 때 단어 단위로 끊어서 읽을 수는 없기 때문에 우선은 읽어 들인 60글자 중에서

Pictures. It is the sequel to 2012's The Avengers, 2015's **Av**

맨 마지막 빈칸 다음의 조각난 "**Av**"를 기억해두고, 다음 줄을 읽을 때 앞에 붙여서 다시 60자를 맞추도록 한다.

Avengers: Age of Ultron and 2018's Avengers: ...

이전 줄을 저장할 때는 "**Av**" 앞의 공백 문자 ' '를 null 문자 '\0'로 대체하여 처리한다.

Pictures. It is the sequel to 2012's The Avengers, 2015's

빈칸은 항상 스페이스 하나로 이루어져 있다고 가정하며, 연속된 띄어쓰기는 고려하지 않는다.

참고로, 이 과제에서 위와 같은 방식으로 구현하는 워드 랩의 경우 줄의 맨 앞에 띄어쓰기가 와서 출력되는 경우는 없다.