4주차 과제 보고서

전공: 컴퓨터공학 학년: 2학년 학번: 20221559 이름: 박준우

1. 사용한 함수는 헤더파일에 정의된 대로 총 3개가 있다.

1) Remove\_Blanks\_At\_The\_End

해당 함수는 fgets로 \n이나 \0이 입력될 때 까지 입력을 받은 string에서 뒤에 있는 공백문자를 무시하기 위해 호출하는 함수이다. 끝을 찾고, 뒤에서부터 공백문자가 아닐 때 까지 탐색 후 그 뒤에 바로 \n/\0로 채우는 방식으로 구현했다.

2) Print\_Line

처리를 마친 줄을 출력한다. 이 과정에서 문제에서 주어진 LIMIT를 확인해 가며, 이어질 공백과, 단어의 길이를 확인해 LIMIT를 넘는지 확인해야 한다. 이 때 다음에 이어질 공백과 단어의 길이를 확인하는 역할을 하는 것이 이어질 Get\_Blanks\_Chars 함수이다.

3) Get\_Blanks\_Chars

앞서 설명한 것처럼 이어지는 빈칸의 개수와 빈칸 뒤에 나오는 단어의 길이를 확인한다. 단어가 시작할 때 까지(빈칸이 이어지는 동안) 빈칸의 개수를 세주고, 단어가 시작하면 단어가 끝날 때 까지(공백문자, 개행, NULL을 만나기 전까지) 문자의 개수를 세준다.

2-1. cc, cflags, target = 사용할 컴파일러 프로그램, 컴파일러에 전달할 옵션, 최종적 원하는 실행파일의 이름을 정의했다.

1) cc = gcc, gcc 컴파일러를 사용하기로 한다.

2) cflags = -W -g, 컴파일 중 warning을 표시하는 옵션인 -W와, gdb 사용 가능하도록 디버깅 정보를 포함한 컴파일을 수행하도록 하는 옵션을 전달하기로 한다.

3) target = main, main이라는 실행 파일을 만들기로 한다.

2-2. objects = main.o String\_Manipulation.o Output.o

main 실행파일로 링크 하는 모든 목적 파일을 나열해 놓는다.

2-3. $(target) : $(objects)

target은 objects에 의존한다. 즉 objects의 생성이 먼저임을 의미한다.

의존하는 객체들이 준비되면 아래의 명령어를 실행한다.

2-4. %.o : %.c

.o 파일은 어떻게 만들어지는지, 정의한다. 이 때 %는 와일드카드로, 과정에서 필요한 .o파일들을 만드는 만드는 규칙을 설명하는 것이다.

밑에서 $@, $<는 각각 목표 파일(.o), 첫번째 의존 파일(.c)을 의미한다.

2-5. main.o string\_manipulation.o output.o : Header.h

이 줄도 위와 동일하게 의존성을 나타낸다. 세 .o파일은 헤더파일이 변경되면 새로 준비해야 함을 의미한다. 헤더파일을 만들기 위한 명령은 따로 없다.

2-6. .PHONY : clean

Clean이라는 대상이 실제 파일이 아닌 명령임을 보장해주기 위한 키워드이다.

clean : 의존 대상이 없이 실행하는 명령이고, 아래 줄의 명령을 실행하게 된다.

명령은 타겟과 생성된 object들을 삭제한다. (rm $(target) $(objects))

3. 앞서 1번에서 설명한 대로 구현하였다.