4주차 과제

전공: 컴퓨터공학과 학년: 2학년 학번: 20241601 이름: 박재우

1.

- Remove_Blanks_At_The_End(char *line): 문자열의 끝(₩n or ₩0)을 찾고, 끝부터 역순으로 공백 ' '을 찾아 탐색하여 제거 대상 위치 결정. ₩n이 있었던 경우 줄 끝을 ₩n + ₩0로 재지정, ₩n이 없었던 경우엔 ₩0으로 종료

목적: 입력된 한 줄의 끝에서 불필요한 공백을 제거하고, 줄 끝의 ₩n 처리까지 수행

- Print_Line(char *line, int *count, int *B_Flag): line에서 단어와 공백을 구분하여 출력, 출력 줄의 길이(count)가 LIMIT을 초과하지 않도록 조절한다. 이어 붙이기가 가능한 경우(L1과 L2가 합쳐질 수 있는 경우) 단일 공백으로 연결, 줄이 꽉 차거나 L2가 새로운 단락이면 줄 바꿈 수행. 출력 후 count, B_Flag 값을 갱신하여 다음 줄 처리에 반영

목적: 줄 내용을 규칙(R1 ~ R6 등)에 따라 출력하며, 필요한 경우 개행(₩n)처리도 담당

- Get_Blanks_Chars(char *line, int start, int *N_Blanks, int *_Chars): Start 위치부터 확인하여 공백을 만나면 N_Blanks 증가(처음 연속된 공백만 계산). 공백이 아닌 문자를 만나면 그 이후를 단어로 간주하고 N_Blanks 증가, ₩n, ₩0 만나면 중단.

목적: line의 Start 위치부터 다음 단어까지의 공백 수와 문자 수를 계산.

2.

cc = gcc : 사용할 컴파일러 gcc를 cc로 지정

cflags = -W -g : -W는 기본 경고 메시지를 출력한다는, -g는 디버깅 정보를 포함한다는 (gdb에서 디버깅을 가능하게) 컴파일 옵션이고 이를 설정하는 코드

target = main : 최종 실행 파일의 이름을 설정하는 것으로, 최종적으로 main이라는 이름의 실행 파일을 생성하겠다는 의미.

Object = main.o String_Manipulation.o Output.o : 오브젝트 파일 목록 지정, 프로그램을 구성하는 .o파일들의 리스트이다.

$(target) : $(objects)

$(cc) $(cflags) -o $(target) $(objects) : 목표(target) 파일 생성 규칙으로, main 실행 파일을 main.o, String_Manipulation.o, Output.o로부터 생성한다.

gcc -W -g -o main main.o String_Manipulation.o Output.o와 동일한 기능

%.o : %.c

$(cc) $(cflags) -c -o $@ $< : 패턴 규칙(모든 .c -> .o) 자동처리

$<는 현재 규칙의 소스 파일(.c), $@는 현재 규칙의 타겟 파일(.o)을 의미.

main.o string_manipulation.o output.o : Header.h : 헤더 파일 의존성 선언으로 Header.h가 변경되면 관련 .o파일들이 자동으로 다시 컴파일된다.

.PHONY : clean : 가짜 타겟 선언, clean은 실제 파일 이름이 아니고 명령어라는 것을 make에게 알려주는 용도

clean :

rm $(target) $(objects) : 클린 명령으로, make clean을 사용할 시 main, .o 파일들을 모두 삭제하여 초기화한다.