2주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20191619 이름: 이동석

**1. 실습 결과**

텍스트이(가) 표시된 사진

자동 생성된 설명

<실습 1>

텍스트이(가) 표시된 사진

자동 생성된 설명

<실습 2>

텍스트이(가) 표시된 사진

자동 생성된 설명

<실습 3> ( LIMIT 36으로 설정했다.)

**2. fmt**

**2-1. main.c**

int main(int argc, char \*argv[])

메인이 되는 함수이다. 파일의 입출력을 담당하고 에러가 날 경우 프로그램을 종료한다.

또한, string\_manipulation.c 와 Output.c에 구현되어 있는 함수들을 언제 사용할 지 정해주고, 변수들을 초기화 해준다. 이후 마지막 줄이 개행문자인지 아닌지를 확인하고 출력 후 프로그램을 종료한다.

**2-2. string\_manipulation.c**

void Remove\_Blanks\_At\_The\_End ( char \*line)

규칙R1과 R2를 위해 공백을 제거하는 함수이다. 변수 newline\_flag를 통해 개행문자인지 공백인지 판단한다.

void Get\_Blanks\_Chars ( char \*line, int Start, int \*N\_Blanks, int \*N\_Chars)

Start 변수는 현재 어디까지 출력되었는지 나타내고, N\_Blanks는 Start이후 존재하는 공백의 개수를 나타내며, N\_Chars는 다음 단어의 개수를 나타낸다. Blank\_flag를 이용해 단어의 끝인지 아닌지를 판단한다.

**2-3. Output.c**

void Print\_Line( char \*line, int \*Count, int \*B\_Flag)

N\_blanks 와 N\_chars, Start 변수를 선언하고 Get\_Blanks\_Chars 함수를 통해 값을 얻어온다. main.c 에서 정의된 변수B\_Flag는 줄을 바꿀지, 아니면 계속 출력할지 결정하고, count 는 현재 줄에 출력할 글자의 개수를 저장하고 있는 변수이다. 공백을 포함해 총 글자수의 개수가 LIMIT보다 작아야한다.

**3. Makefile 코드**

cc = gcc

cc는 C 컴파일러에 대한 매크로 이며, 미리 정해져 있는 매크로이다.

CFLAGS = -W -g

2번째 줄 역시cc와 마찬가지로 미리 정해져 있는 매크로이며, C컴파일러의 옵션에 대한 매크로이다. -W 옵션은 모호한 코딩에 대해서 경고를 보내주고, -g 는 디버깅을 위한 옵션이다.

TARGET = animal.exe

TARGET 은 최종적으로 빌드하여 생성할 파일의 이름이다. Makefile의 rule에서 목표에 해당한다.

objects = dog.o blackcow.o turtle.o main.o

objects 는 오브젝트 파일들을 한번에 지정하는 매크로이다.

$(TARGET) : $(objects)

$(cc) -o $(TARGET) $(objects)

목표가 되는 파일과 오브젝트 파일들은 서로 의존관계이다. $를 이용해 매크로를 사용한다. 이후 수행할 명령(recipe)을 TAB문자로 공백을 주어 작성한다.

.c.o :

$(cc) $(CFLAGS) -c $< -o $@

Make의 확장자 규칙(suffix rule)이 있다. 이 규칙에 의해서 .c 파일을 .o 파일로 자동으로 처리한다. 그 규칙을 $< 와 $@로 표현했다. 이 둘은 내부 매크로로 불리는데, $<는 현재의 TARGET 보다 더 최신의 TARGET, $@는 현재 TARGET을 말한다. 없어도 make는 잘 동작한다.

.PHONY : clean

clean:

rm $(TARGET) $(objects)

PHONY는 실제파일이름과의 충돌을 해결하기 위해 사용된다. PHONY로 가짜타겟임을명시하지 않으면, clean의 이름을 가진 파일이 있을 때 제대로 명령(recipe)를 수행하지 않는다.

**4. 규칙 R5 알고리즘**

규칙 R5는 main 함수에서 확인한다.

1. Remove\_Blanks\_At\_The\_End(line2) 전처리 작업 진행
2. line[2]=’ ‘ && Coun!=0

YES-> putchar(‘\n’) , B\_Line=1, Count=0

NO -> if 문 break

이후 line1 <-> line 2

**5. make 옵션**

make

-C DIRECTORY : makefile을 읽지 않고 해당 디렉토리로 이동한다.

-d : makefile을 실행하면서 나오는 정보를 출력한다. 디버그 옵션이다.

-h : 옵션 도움말을 출력한다

-v : 현재 make의 버전을 출력한다.

-s : recipe 를 출력하지 않는다. (--silent, --quiet)

-k : 타겟을 만들지 못해도 계속 실행한다. (--keep-going)

-t : 타켓을 remake하지 않고 최신화한다.

이 외에도 여러 옵션이 있다. 옵션은 앞서 말했듯이 make -h를 통해 볼 수 있다.