2주차 결과보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20192135 이름: 윤영인

**1.**

1) 실습 1

animal.h 헤더파일

텍스트이(가) 표시된 사진

자동 생성된 설명

dog.c

텍스트, 실내, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명

blackcow.c

텍스트, 실내, 스크린샷이(가) 표시된 사진

자동 생성된 설명

turtle.c

텍스트, 실내, 스크린샷이(가) 표시된 사진

자동 생성된 설명

main.c

텍스트이(가) 표시된 사진

자동 생성된 설명

Makefile

텍스트이(가) 표시된 사진

자동 생성된 설명

make 실행

텍스트이(가) 표시된 사진

자동 생성된 설명

make clean 실행

텍스트이(가) 표시된 사진

자동 생성된 설명

2) 실습 2

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

정수형 i를 2로 나누면 소수점은 생략되고 정수형 값이 나온다. 따라서 i/2를 double로 casting한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

수정한 코드

3) 실습 3

Remove\_Blanks\_At\_The\_End 함수

텍스트, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명

아래의 사진은 LIMIT을 36으로 설정한 결과화면이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**2.**

1) Remove\_Blanks\_At\_The\_End

줄 바꿈 전의 blank를 지우는 함수로, R2에서 LIMIT보다 작은 글자로 구성된 L1에 이어 L2의 단어를 같은 줄로 출력할 때, 하나 이상의 blank가 있어도 하나의 blank만 출력해야 한다. 따라서 인자로 받아온 line 배열에서 line[k]가 줄바꿈 문자거나 ‘\0’이면 for문을 종료하는데, 줄바꿈 문자일 경우 newline\_flag를 1로 설정한다. k가 멈춘 지점에서 다시 감소하면서 line[k]가 blank인지 확인하여 blank이면 for문을 종료한다. flag가 1로 설정되어 있으면, 줄바꿈 문자와 blank를 line에 추가하고 그렇지 않으면 ‘\0’ 문자를 추가한다.

2) Get\_Blanks\_Chars

Print\_Line 함수에서 빈칸과 char형 문자를 세는 함수이다. 인자로 받아온 start 지점에서 for문을 돌면서 line[i]가 줄바꿈이거나 ‘\0’문자이면 for문을 종료하고, char가 나오기 전까지의 blank와 char의 수를 세며 다시 blank가 나오면 멈춘다.

3) Print\_Line

현재의 줄을 Count의 크기에 따라 같은 줄에 출력할 지 말지를 결정하여 출력하는 함수이다. N\_Blanks는 char가 나오기 전의 blank의 개수로 0이 아닐 때만 다음 라인에 출력하고, B\_Flag가 1이면 같은 줄에 출력한다. B\_Flag가 1이고 (\*Count + N\_Chars + 1) <= LIMIT이면 처음에 blank를 추가하고 N\_Chars의 개수만큼 line을 출력한다. (\*Count + N\_Blanks + N\_Chars) <= LIMIT이면 N\_BLANKS + N\_Chars만큼 line을 출력한다. 이 경우에 모두 해당하지 않으면, Count가 0일 경우엔 N\_Blanks + N\_Chars만큼 line을 출력한 후 줄바꿈을 한다. 0이 아닐 경우엔 줄바꿈을 하고 N\_Blanks + N\_Chars만큼 line을 출력한다.

4) main

입력 파일을 열어 Remove\_Blank\_At\_The\_End 함수와 Print\_Line을 호출하여 B\_Flag와 B\_line의 값을 결정하면서 LIMIT에 맞춰 줄바꿈을 출력한다. 만약 마지막 줄이 비어있지 않으면 마지막 문자가 줄바꿈인지 확인하여 출력 파일에 넣어준다.

**3.**

CC = gcc // 컴파일 도구로 gcc를 사용하기 위해 매크로를 CC로 선언한다.

CFLAG = -c // 링크하지 않고 컴파일하여 오브젝트 파일을 생성하는 옵션인 -c를 사용하기 위해 매크로를 CFLAG로 선언한다.

TARGET = animal.exe // 실행파일 이름을 animal.exe로 만들기 위한 매크로인 TARGET을 선언한다.

OBJS = main.o blackcow.o turtle.o dog.o // 오브젝트 파일들의 매크로를 OBJS로 선언한다.

all : $(TARGET) // make all을 하면 TARGET 파일이 생성된다.

$(TARGET) : $(OBJS) // TARGET 파일을 생성하기 위해 OBJS 파일에 의존하

$(CC) -o $@ $^ //-o 옵션을 사용하여 의존 목록 파일 전체에 대응하여 TARGET 이름으로 실행파일을 생성한다.

%.o : %.c

$(CC) $(CFLAG) -o $@ $< // .c 파일로 .o 파일을 생성하기 위해 -c -o 옵션을 사용하여 .c 파일과 이름이 같은 .o파일을 생성한다.

main.o blackcow.o turtle.o dog.o : animal.h // .o 파일들은 animal.h에 의존한다.

clean :

rm $(OBJS) $(TARGET) // make clean을 하면 오브젝트 파일과 실행파일이 지워진다.

.PHONY : clean // 파일 중에 clean이라는 파일이 있어 make clean이 수행되지 않는 것을 방지한다.

**4.**

R5:입력 줄의 첫 글자가 빈칸이면 앞줄과 합쳐지지 않는다. 여러 개의 빈칸이 줄의 처음에 있으면 다음 줄에 출력하며, 첫 부분의 빈칸은 첫번째 빈칸을 포함하여 그 개수만큼 출력한다.

1) Remove\_Blanks\_At\_The\_End(line1)을 호출하여 line1을 전처리 하고 무한루프를 돌기 시작한다.

2) B\_Line이 0이면 1) 과정 후에 읽은 줄이 빈 줄이 아니라는 뜻이므로 Print\_Line 함수를 호출하여 line1을 출력한다. 1일 경우, B\_Line을 0으로 세팅한다.

3) 2)에서 호출한 Print\_Line 함수에서 Get\_Blanks\_Chars 함수를 호출하여 빈칸과 char형 문자의 개수를 세어 fmt 규칙에 맞게 출력한다.

4) Remove\_Blanks\_At\_The\_End(line2)를 호출하여 line2를 전처리 하고, 첫 글자가 빈칸이거나 Count가 0이 아니면 줄바꿈을 한다.

5) line2를 fmt에 맞게 출력하기 위해 line1에 넣어주고 무한루프의 처음으로 돌아가 위의 과정을 반복한다.

6) eof를 읽으면 무한 루프를 종료한다.

7) 마지막 문장이 비어있지 않을 때, 마지막 문자가 줄바꿈인지 확인하여 맞다면 출력 파일에 넣어준다.

**5.**

1) -o: 실행 파일명을 결정한다. 옵션을 추가하지 않으면 기본적으로 a.out으로 실행파일이 생성된다.

2) -c: 링크하지 않고 컴파일하여 오브젝트 파일을 생성한다.

3) -g: 컴파일하면서 생성한 디버깅 정보를 개체 코드에 포함한다.

4) -d: Makefile을 수행하면서 각종 정보를 모두 출력한다.

5) -h: 옵션에 관한 도움말을 출력한다.

6) -v: make의 버전을 출력한다.

7) -W: 경고 메시지를 모두 출력한다.

8) -w: 경고 메시지를 출력하지 않는다.

9) -l: 라이브러리 이름을 지정한다.

10) -k: make를 수행하다가 에러가 발생하여도 멈추지 않고 계속 수행한다.

**6. 참조**

1) 강의자료, 2주차\_unix(linux) 에서의 컴퓨터 프로그래밍.pptx

2) 강의자료, 부록\_Makefile.pdf

3) 강의자료, UNIX-2.pdf