2주차 결과보고서

전공 : 컴퓨터공학과 학년 : 2학년 학번: 20211558 이름: 윤준서

**1. 목적**

실습 과정에 개발한 fmt에 대하여 결과 보고한다.

**2. 실습 결과 화면**

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

text.txt 파일에 텍스트 내용을 입력하고, fmt를 통해 이를 출력한다.

한 줄 당 글자 수 제한은 72이지만, R4와 R6에 따라 줄마다 서로 조금씩 차이가 발생한다.

**3. 함수 설명**

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

먼저 파일 읽기를 시도한다. 실패 시 "File open error." 를 출력한다.

파일의 이름이 없을 경우, "Usage : fmt filename | > outfile" 를 출력한다.

이후 line1과 line2에 메모리를 할당한다. 실패 시 "Memory allocation error." 를 출력한다.

이후 while 반복문을 통해 함수들을 반복적으로 시행한다.

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

문자열 뒤에 공백을 모두 제거한다.

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

문자열의 공백 개수와 문자 개수를 센다.

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

문자열을 limit에 맞게 출력한다.

**4. Makefile 설명**

**cc = gcc >**gcc를 이용해 컴파일한다.

**cflags = -W -g >**컴파일 옵션을 설정한다. 모든 경고를 활성화하고, 디버그 정보를

포함한다.

**target = fmt >**실행 파일 이름을 설정한다.

**objects = main.o stringM.o output.o >**오브젝트 파일을 나열한다.

**$(target) : $(objects)**

**$(cc) $(cflags) -o $(target) $(objects)** **>**오브젝트 파일을 생성한다.

**%.o : %.c**

**$(cc) $(cflags) -c -o $@ $< >**gcc를 이용해 컴파일한다.

**main.o stringM.o output.o : Header.h**

**.PHONY : clean >**clean을 명령어로 사용한다. 디렉토리 내 파일 이름 중복을 방지한다.

**clean :**

**rm $(target) $(objects) >**clean을 입력하면 타겟과 오브젝트 파일을 삭제한다.

**5. R5 알고리즘 설명**

main 함수 내부에 알고리즘이 있다.

if line2[0] == '공백' or Count != 0 인 경우, 다음 줄의 문자를 출력하고, B\_Flag와 Count를 0으로 초기화한다.

else if line2[0] == '\n' 일 경우,

if B\_Flag == 1 인 경우, 다음 줄의 문자를 출력하고, B\_Flag = 0으로 초기화, 다시 다음 줄 의 문자를 출력한다. 이후 B\_line = 1, Count = 0으로 초기화한다.

이후 line1과 line2를 temp를 이용해 바꾼다.

이후 if line1[0] != '\n' 인 경우,

line[i] == '\n' or NULL일 때까지 i를 0에서부터 늘려가며 반복문을 실행한다.

이를 만족 시 break 한다. 이때 line[i] == '\n' 일 경우 '\n' 을 출력한다.

**6. Make 옵션 설명**

make 옵션들에 대하여 다음과 같다.

**• -c dir** Makefile을 읽지 않고 디렉토리로 이동한다.

**• -d** Makefile을 수행하면서 정보를 출력한다.

**• -h** 옵션에 관한 설명을 출력한다.

**• -f file** file에 해당하는 파일을 Makefile로 간주한다.

**• -r** 내제된 규칙을 없는 것으로 간주한다.

**• -t** 파일의 생성날짜를 현재 시각으로 설정한다.

**• -v** make의 버전을 출력한다.

**• -p** 내부의 세팅된 값을 출력한다.

**• -k** 에러가 나도 계속 실행한다.