실험 UNIX-2 결과보고서

전공: 컴퓨터공학 학년: 2 학번: 20211599 이름: 주현수

2-1.

1. Main

프로그램의 가장 핵심 부분이다. 전체적인 흐름을 가지고 외부 함수를 사용할 수 있게 호출한다.

1. line1과 line2를 읽어들인다.
2. line1이나 line2가 빈 값이면 프로그램을 종료한다.
3. 그렇지 않은 경우 line1의 끝부분의 blank를 제거한다.
4. line1[0]==’\n’이라면 \n을 출력한다.

(while 반복문 안 / 초기 Count 값 =0)

1. B\_Line==0일 경우 line1을 Print\_Line함수로 출력하고, 그렇지 않은 경우 B\_Line=0을 해준다.
2. Count 값이 0이 아니라면 B\_Flag=1
3. line2을 읽어들인다.
4. line2가 비어있으면 while문 탈출
5. line2가 비어있지 않으면 맨 끝의 blank 삭제
6. line2[0]가 공백이고 Count가 0이 아니면 ‘\n’을 출력한다. B\_Flag=0/Count=0으로 설정
7. 그렇지 않고 line2[0]==’\n’이고 B\_Flag==1인 경우 ‘\n’을 출력하며 B\_Flag=0으로 설정한다.
8. \n을 출력하고 B\_Line=1/Count=0으로 설정한다.
9. line1과 line2를 바꾼다.
10. 5~13을 무한으로 반복시킨다.
11. line1[0]이 \n이 아니고 line1이 \n으로 끝나면 \n을 출력한다.
12. line1[0]이 \n이거나, line1[0]이 \n은 맞는데 끝이 \n이 아니라면 프로그램을 종료시킨다.
13. Print\_Line

출력을 할 때 사용하는 함수이다. 줄의 맨 앞에 긴 공백이 있으면 그냥 이어 출력하지 않고 줄바꿈을 해서 그대로 출력하는 R5규칙을 수행한다.

1. Remove\_Blanks\_At\_The\_End

문장 끝의 공백을 지우는 함수이다.

1. Get\_Blanks\_Chars

문장 중간의 공백과 길이를 확인하는 함수이다.

2-2.

실습1)

텍스트이(가) 표시된 사진

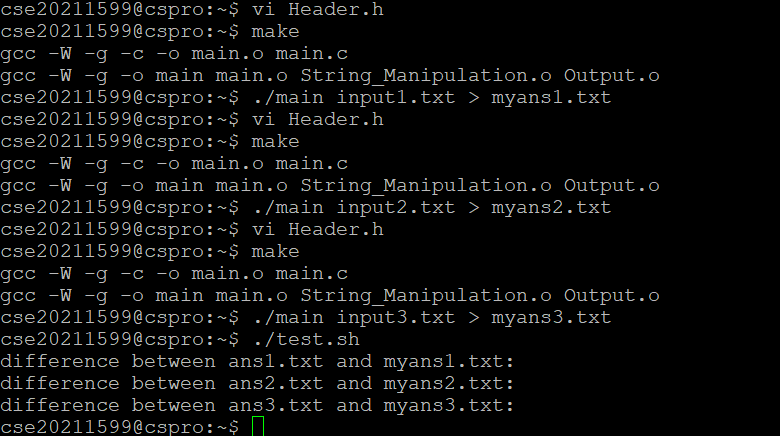
자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명실습2)

실습3)

텍스트이(가) 표시된 사진

자동 생성된 설명

3.

cc=gcc //gcc로 컴파일을 하겠다.

target = animal\_exe //최종 파일명은 animal\_exe 다.

objects = dog.o blackcow.o turtle.o //필요한 파일은 다음과 같다.

$(target) : $(objects)

$(cc) -o $(target) $(objects) //컴파일 한다.

$(objects) : some.h //필요한 헤더파일로 합친다.

.PHONY : clean //같은 이름이 있을 만일의 상황을 대비한다.

clean :

rm $(target) $(objects) //make할 때 발생하는 target 파일과 object파일들을 삭제한다.

4. make의 옵션

-k : error가 발생해도 실행시킨다.

-d : make하는 정보를 보여준다.

-f file : 만든 파일을 make할 수 있는 대상이라고 여긴다.

5. 과제 코드

1. main.c

#include "Header.h" //헤더파일

int main() //메인함수

{

int T,N,i; //각각 책 권수, 페이지 수, 반복문에 사용되는 변수

printf("how many book :"); //입력해야할내용 풀력

scanf("%d",&T); //실행해야 할 책의 권수 받기

for(i=0;i<T;i++) //책의 권수에 대해서 각각의 페이지 수 입력받기

{

printf("number of pages :"); //입력해야 할 내용

scanf("%d",&N); //사용자가 페이지 수 입력하기

printf("\n"); //줄바꿈

page(N); //페이지를 받고 그 안에 있는 0~9까지의 숫자 카운트 함수

}

return 0; //종료

}

1. page.c

void page(int N) //N장의 페이지가 있는 책에서 0~9까지 숫자 갯수 카운트하는 함수

{

int i; //반복문에 쓰일 변수

int a[10];//0~9까지 갯수를 받을 배열 생성

for(i=0;i<10;i++) //배열 생성

{

a[i]=0;}//생성된 배열을 일단 모두 0으로 설정

for(i=1;i<=N;i++)//1페이지부터 N페이지까지 반복

{

int b; //i의 값을 변화시키면 반복문에 오류가 생길 수 있으니 임의의 변수에 i값을 넣어서 사용

b=i; //i값을 임의의 변수 b로 대체한다

while(1) //무한루프

{

a[b%10]+=1; //주어진 페이지의 일의자리 수를 찾고 이를 카운트한다

b=b/10;//10으로 나눈 몫을 b로 설정

if(b==0)break; } //이런식으로 반복되다가 b가 0이되면 반복문 탈출

}

print(a);}

1. print.c

#include "Header.h"//헤더파일

void print(int a[])//찾은 결과를 출력하는 함수

{

int i;//반복문에 사용되는 변수

for(i=0;i<10;i++)//0~9까지 각각의 갯수 출력

{

printf("%d : %d \n",i,a[i]); //숫자 0은 페이지를 통틀어서 몇개 나오는지 출력

}

}

1. Header.h

#include<stdio.h>

void page(int N);

void print(int a[]);

1. Makefile

cc=gcc #gcc를 이용해 컴파일을 하겠다

cflags = -W -g

target = answer #answer의 이름으로 make를 하겠다.

objects = main.o page.o print.o #사용되는 파일

$(target) : $(objects)

$(cc) $(cflags) -o $(target) $(objects) #컴파일한다

%.o : %.c

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

main.o page.o print.o : Header.h #헤더파일 사용

.PHONY : clean #만일의 경우를 대비한다.

clean : # 나온 파일을 지운다.

rm $(target) $(objects)