171220 고찰

# 51.c 고찰

Getch()함수에 관한 소스코드를 공부했다. Getch() 함수를 쓰기위해서는 #include<conio.h>를 해줘야한다. 그런데 나는 우분투 환경에서 c언어를 하다보니 책에 있는 내용을 그대로 하는 것과는 차이가 조금 있었다. 우분투에는 conio.h 라는 헤더파일이 없다는 사실을 인터넷 검색을 해보다가 알게되었고, getch 함수를 쓰기위해서는 getch.h라는 헤더파일을 만들어서 #include “getch.h”를 해주는게 가장 편한 방법이란 걸 알게되고 getch.h 헤더파일을 추가했다 코드는 아래와 같다.

#include <stdio.h>

#include <termio.h>

int getch(void){

int ch;

struct termios buf, save;

tcgetattr(0,&save);

buf = save;

buf.c\_lflag &= ~(ICANON|ECHO);

buf.c\_cc[VMIN] = 1;

buf.c\_cc[VTIME] = 0;

tcsetattr(0, TCSAFLUSH, &buf);

ch = getchar();

tcsetattr(0, TCSAFLUSH, &save);

return ch;

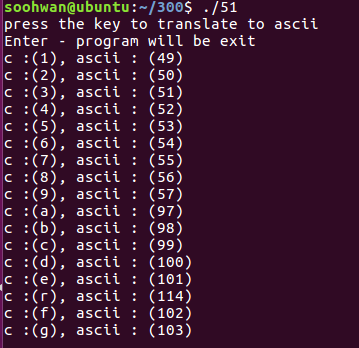
}

사실 위의 코드내용이 무슨 내용인지 거의 모르겠지만 차차 알아가도록 하자. 그리고 그 외에도 배울만했던 점이 ENTER를 누르면 프로그램이 끝나도록 하는 방법을 알게되었다. 책에서는 ENTER의 아스키 코드값이 13이라고 했지만 내 우분투 상에서는 찍어본 결과 10이라는 사실을 알게되었다.(아스키 코드값이 환경마다 다를수가 있나라는 생각이 들긴 하지만 찍어본 결과가 그러니…) 이 예제 51에서는 ENTER를 define을 해준 후 ( #define ENTER 10) do while 문에서

While(ch!=ENTER) 로 정의를 해주었다. 즉, ch가 enter가 아닐 때에 계속 돌아가도록 설정을 해준 것이였다. 그리고 본격적으로 getch()함수가 무엇인고 하니, 문자를 한 개 입력받아서 int 형으로 반환해주는 함수이다. 즉, getch함수를 실행되면, scanf없이도 문자를 입력 받은 뒤, 그 값을 어떠한 변수에 저장을 하면 그 값을 int형으로 바꿔주는 것이다. 내가 컴퓨팅사고 시간에 들은 atoi (ascii to inteager) 함수와 비슷한 것 아닐까란 생각이 든다. 그럼 atoi와 getch의 차이는 무엇일까??

검색을 해봤는데 잘 나오지 않는다… 나중에 다시 한번 알아보도록 하자.

실행결과



# 53 scanf 고찰

Scanf는 매우 친숙한 함수지만 가끔 컴파일 할 때 warning으로 많이 마주치게 되는 함수이다. 개인적으로 그래서 scanf를 많이 쓰게 될 때면 약간의 거부감이 들기도 한다. 이 책의 소스코드는 아니지만 scanf함수에 관한 설명을 읽다가 문득 scanf로 %s %s %s 받을 때는 어떤 식으로 작동되는지 궁금해져서 한번 실험을 해보았다. (교수님이 c언어는 궁금한거 다 해봐야 는다고 했다) 근데 컴파일은 되는데 자꾸 warning과 함께 프로그램 실행시 core dumped라는 익숙한 멘트가 자꾸 나오길래 뭐가 잘못됬나…한참 봤더니 문자열을 입력하는데 저장 변수를 int형으로 써놓았었다. 빨리 char형으로 바꿔준 뒤 컴파일 및 실행을 해보았지만 결과는 똑같이… core dumped.. 그래서 뭐가 문제인지 (사실 우분투가 이상한가보다라는 생각이 들었다..) 생각해보다가 책을 읽어보았는데 문자열을 scanf로 입력받을 때 char s1[10]; 이런식으로 선언해야 한다는 것을 알게되었다. 아! Scanf로 문자열을 입력받을때는 크기를 어느 정도 적어줘야 하는구나! 매우 많이 써본 함수지만 아직도 이런걸 모르는 것 보면 난 아직 한참 공부를 해야 할 것 같다.

# 예제 54 printf

아직까지는 매우 쉬운 함수들이 나오고 있다. 이번에는 printf다. 거의 맨 처음 접하고 가장 많이쓰는 함수 아니겠는가. 물론 이게 끝이 아니다. 이번 장에서는 %d, %5d, %10d, %-d 와 같은 활용을 조금 배우게 된다. 근데 이것은! 컴퓨팅사고 중간고사에서 나왔던 문제에 많이 공부한 내용이다. 내가 필기했던 내용을 보자면….

%nd - n만큼의 공간을 확보하고 오른쪽 정렬

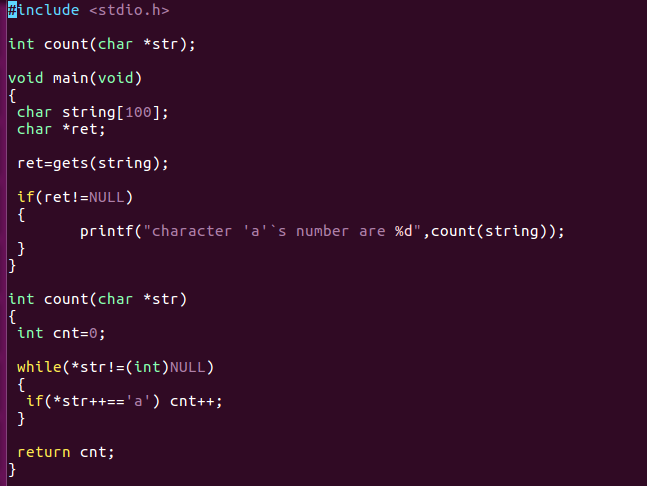
%-d – 왼쪽정렬

%0d – 남는 자리를 0으로 채운다.

이다. 이게 핵심이다. 이것만 알면 printf 활용은 거의 마스터일것이다. 근데 보다보니 이 책 오타도 많은 것 같고, 엄청 잘 만든 책 같지는 않지만… 예제 푸는 용도로 산 것이니 그냥 보자.

Printf를 구현해보자! 하고 호기롭게 도전했다가 몇분만에 내가 지금 그럴 클라스가 아님을 깨닫고 인터넷에 검색을 해봤더니 아직 내가 모르는 함수들고 구성되어 있다. 그래..나중에 다시 해보자.

# 예제 55 gets



와우.. 오늘 한 것 중에 가장 아름다운 코드인 것 같다. 뭔가 이 코드는 어디선가 써먹기 좋을 듯 하다. Gets 함수보다도 gets 함수를 설명하기 위해 만든 저 소스코드가 마음에 든다. 분석을 해보자면 내가 입력한 문장 ex) this is a program test plan a 라고 해보자. 위의 코드는 저 문장 중에서 a의 개수를 세어서 몇 개인지 나타내는 코드이다. 씨프나 고프 숙제나 시험문제 정도로 딱 적당할 것 같다. 일단 그런 함수를 만들기 위해 메인 함수 위쪽에 함수 선언을 해뒀다. 그리고 메인함수 밑쪽에는 함수에 대한 정의가 되어있는데, 내가 만든 count라는 함수에 gets함수를 이용해 내가 입력한 문자열을 count함수에 넣어주면, 이 count라는 함수는 while을 이용해 이 문자열이 끝날때까지 즉, \*str==NULL일 때까지 반복한다. 무엇을? \*str은 str주소 안에 들어있는 값이므로 그 값을 반복마다 ++를 해주며, 만약 그 값이 str++==’a’라면 미리 선언해둔 변수도 같이 ++를 해줌으로써 ‘a’를 카운트해준다. 그리고 미리 선언해둔 변수로 return을 해줌으로써 그 값을 출력한다. 크~멋지지 않은가. 근데 갑자기 궁금한게 생겼다. \*str++가 예를 들어 this이면 처음에 t에서 시작해 다음은 h 그 다음은 i이런 식으로 갈 것 아닌가. 그럼 \*(str++)와는 다를까? 실험해보자.

실험결과 똑같이 나온다 결과가.. 아무래도 내가 컴퓨팅사고 기말 대비하면서 배열관련 포인터 공부를 하다보니 궁금하게 된 것 같은데 위 코드는 문자열이므로 \*str++나 \*(str++)나 똑 같은 듯 하다. 배열과 문자열의 차이.. 그걸 생각 못했다. 오늘 고찰 끝~