A Study on the Web-based Cost-Effective Education System for C Programming

KwanSun Choi, HeungGoo Jun, Dongsik Kim, Sunheum Lee
Division of Information Technology Engineering
Soonchunhyang University, Asan-si, ChoongNam-do, Republic of Korea
e-mail : cks1329@sch.ac.kr

Abstract: We have implemented an interactive multimedia education system for C programming. The system consists of two parts. One is multimedia contents to help students better understand C language syntax, programming style, and program logic and the other is a web-based compilation support system that compiles C programs at the server side which are submitted through Web by students and returns their execution results to the students' PC.

Although there are currently some restrictions that students should replace the inputs functions such as scanf(), getc() and getch() with assignment statements or fscanf(fp, ,), since the system has been implemented using general web technologies and shareware C compiler, the education system could be one solution that education institutions seek to reduce annual immense expenditure of money on C complier.

1. Introduction

Increased use of computers and the Internet has created a basis for new kinds of learning. A web based interactive learning is an important emerging educational trend. To enhance a quality of engineering education we have been working on the design and implementation of an interactive web based educational system using java applet.

In this paper, we report on the development of a course module on C programming language through the internet, which is offered to undergraduate participating in the education curriculum being established at Soonchunhyang University. Our System for C programming, uses none of the commercial software packages. This makes the system independent of the process of commercial software development, in which a version of the software might become obsolete very quickly. Our system uses serverside compiler under the LINUX through Internet instead of using desktop commercial compiler. Therefore All of students need not purchase and install commercial compiler in their PC in order to study C Programming Language. They just connect our system using web browser and send their C language source program to the server side, they will acquire the result of programming execution. Our proposed system is fig 1. Students use web borwer to make C source programs and send them to server. In the server CGI program requests for GCC compiler to compile, excute students program and execution results is sent to client.

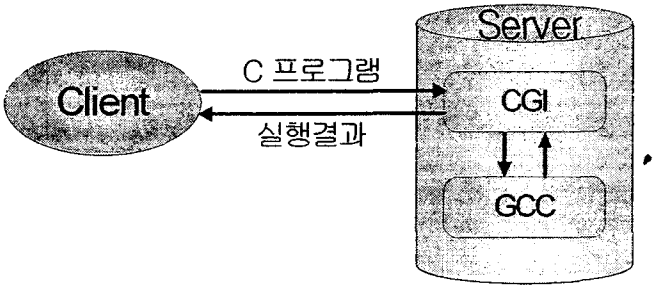

Fig . Proposed system

This paper 's configuration is as follows. This paper is explained C language syntax, programming concept using animation and java applets in the chapter 2, the method of implementation and solving of problems for our proposed system in the chapter 3, conclusion and future's research in the chapter 4.

2. Creative animations

We developed Flash animations and Java Applets by which students could better understand C language syntax, programming style, and program logic. Figure 1. shows a Flash animation which explains how many memories an int type variable occupies, when the overflow and underflow happen and what then happens. Students could easily catch the concept of the overflow and underflow by seeing the animation in Fig 2.

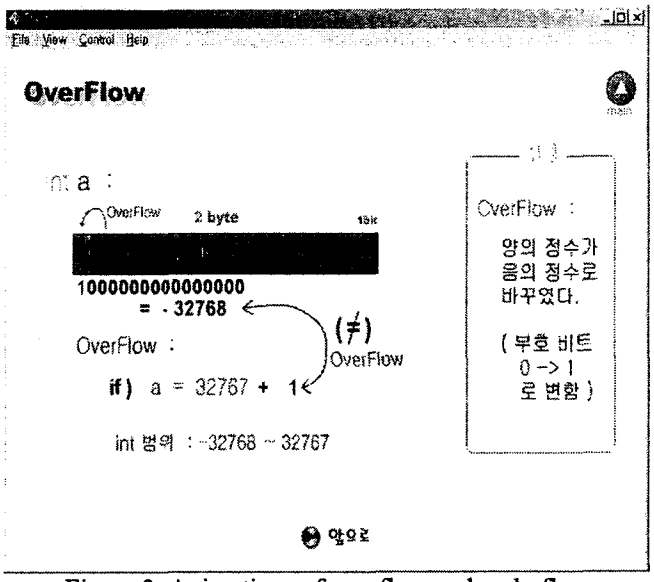

Figure 2. Animations of overflow and underflow

Figure 3. shows a Java Applet that helps students learn the meaning of variable declarations and format control characters for printf statement. Student can enter initial values for variables a, b and c. and can choose the output format for the printf statement by selecting on of the two radio buttons. When you choose the radio button captioned Enter Output Format, then you must key in your own output format control characters for the printf statement. If you finished typing and pushed run button, then you could see the results of Prinf.cpp program at the bottom. Using the Applet in Fig 3. students can repeatedly check out how a variety of output format control characters function, with a little burden of keying.

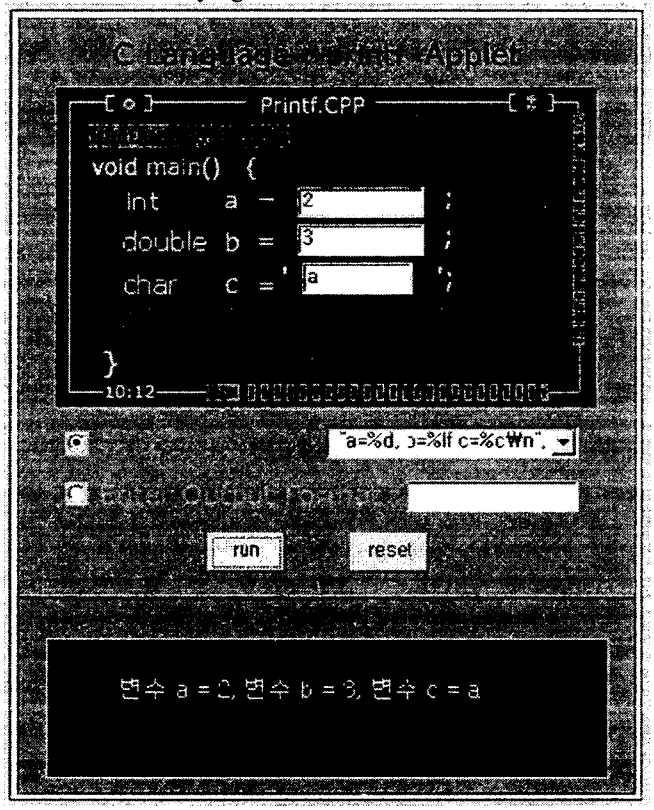

Figure 3. Java Applet for exercises for output format control characters functions

3. The web-based compilation support system

3.1 Basic operations

We implemented an web-based compilation support system that compiles C programs at the server side which are submitted through Web by students and returns their execution results to the students' PC. The system provides the text-box shown in Fig 4. in which Students can key in their own C programs on the web. Because we use the shareware C compiler and web technologies to economically implement the web-based compilation support system, there are currently some restrictions that students should replace the inputs functions such as scanf(), getc() and getch() with assignment statements or scanf(fp, ,). Therefore, to check out whether students violate the restrictions imposed by the system, the system provides a source code checking function to users. If students use restricted functions such as scanf(), getc(), etc, the system will warn students of replacing restricted functions with proper substitutes.

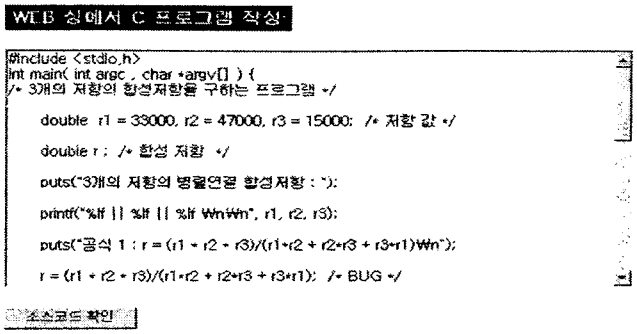

Figure 4. Text-box for keying in source program

When students pass the source code checking step, they will receive the echoed source codes shown in Fig 5. from the system. Students can execute their programs by clicking the compile and execution button on the echoed source codes.

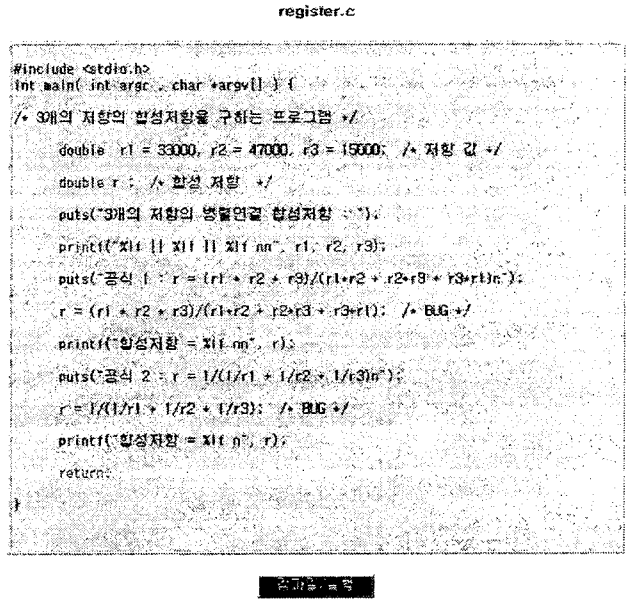

Figure 5. Echoed source program with compile and execution button

After clicking the compile and execution button, students would receive the executed results shown in Fig 6. if their programs did not include semantic errors.

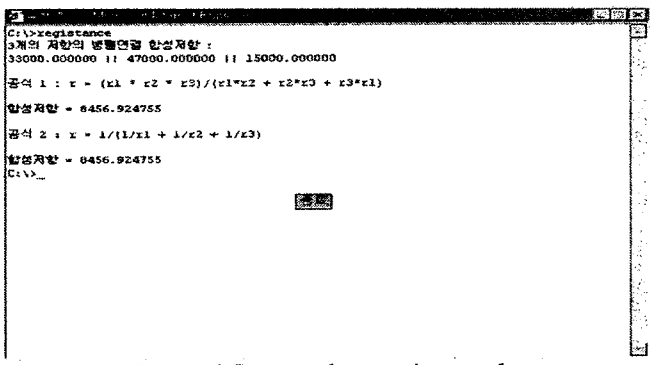

Figure 6. Returned execution results

If any, error messages would be returned shown in Fig. 7.

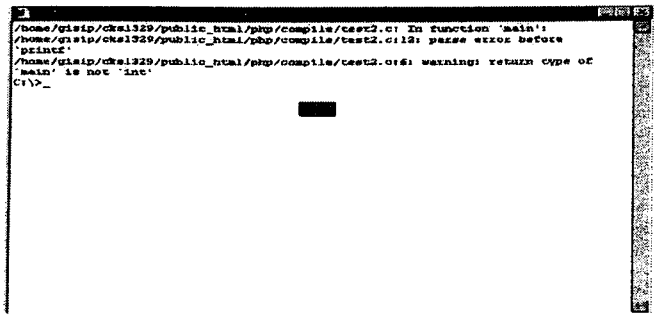

Figure 7. Returned error messages

3.2 Restrictions and Substitutes

To economically and simply implement the web-based compilation support system, we imposed some restrictions on the system. When your source program includes the restricted functions, you must replace the functions with proper substitutes. For example,

if your source program includes a function scanf("%d %f", &a, &b) then replace the function with assignment statements a=20; b=25.233;. If it includes c=getch(); then replace it with c='A';. If it includes s=gets(); then replace it with s="Web based C programming Language using GCC on UNIX";.

If your source program includes loop statements in which input functions are located, then replace the input functions with file input functions. Figure 8. shows how to replace. The program shown in Fig. 8(a) is to sum 10 integer data from keyboard. The program shown in Fig. 8(b) is substitute for the program shown in Fig. 8(a).

```
#include <stdio.h>
void main(int argc, char *argv[]){

  int n, a[10], sum=0;

  for(n=0; n<10; n++){
    printf("enter %d :", n);
    scanf("%d", &a[n]);
    sum += a[n];
  }
  printf("\n입력된 데이터의 합=%d", sum);
}
```

```
#include <stdio.h>
int main(int argc, char *argv[]){
    FILE *rpt;
    int n, a[10], sum=0;
    rpt=fopen("test.dat","r");
    for(n=0; n<10; n++){
      fscanf(rpt, "%d",&a[n]);
      sum += a[n];
    }
  printf("\n입력된 데이터의 합=%d", sum);
  fclose(rpt);
  return;
}
```

(a) original program (b) substitute for original source program

Figure 8. Proper substitution

Although the web-based compilation support system currently do not support interactive executions, it is very useful to implement the Web-based Cost-Effective Education System for C Programming

4. Field test of the system

Qualitatively, the system effectively help students to easily learn C Language syntax and programming. The integration of multimedia education contents with the web-based C compiler support system allows students to exercise himself in C programming anytime and anywhere if they can have access to the internet.

Quantitatively, Table 1 shows the response times of Bubble sorting program with 5000 data items corresponding to simultaneous users when Pentium III 1GHz is used as server computer. The response time includes network latency time and server's service time.

Person	5	10	20	30
Elapse Time(s)	16.04	32.037	60.588	95.358

Table 1. response times corresponding to users serviced

Table 2. shows the response times under the same network environment and test program as Table 1. when only one user is serviced by the system. In case of Pentium III 1GHz sever, the response time is 5.8484 seconds. When the test program is executed at stand-alone desktop computer(Pentium III 1GHz) the response time is 1.0 second. Therefore, performance of our system depends on network latency time.

PC Spec.	Elapse Time(s)
Pentium III 1GHz	5.8484
Pentium II 266MHz	6.084

Table 2. response times when a single user is serviced

5. Conclusions and Future Work

In this paper, we implemented an web application which compiles C programs submitted by students and returns their execution results to their PC. Based on this application, we built an interactive multimedia education system which help students to easily learn program logic and C programming. With the education system, students can first understand what C language syntax is or how program logic functions by being assisted by Flash animations and Java Applets, and then execute their C programs anytime and anywhere without their own compiler if they can have access to the education system through Inernet, and education institutes could provide effective C programming course without annual immense expenditure of money on complier.

But there are some restrictions that students should replace the input functions in the program with assignments and input functions in the loop with fscanf(fp, ,) functions. Also some system functions can not be used in programs. Despite the above mentioned drawbacks, 30 students better understood and simultaneously executed their programs successfully by obeying the proposed conversion rules.

he future, we plan to work out the input function restrictions and improve the response time of the system.

References

[1] Baldwin, D. and Koomen, J. "Using Scientific Experiments in Early Computer Science Laboratories." SIGCSE Bulletin 24 (March 1992): 102-106.

[2] Boroni, C.M., Goosey, F.W., Grinder, M.T., Lambert, J.L., Ross, R.J., "Tying it all Together Creating Self-Contained, Animated, Interactive, Web-Based Resources for Computer Science Education", Proceedings of the 30th

SIGCSE Technical Symposium on Computer Science Education, (March 1999): 7-11.
[3] Epp, E.C. "An Experimental Computer Science Laboratory." SIGCSE Bulletin 23 (March 1991): 11-14.
[4] Jones, P.R, Ruehr, F., Slater, R. "Web-Based Laboratories in the Introductory Curriculum Enhance Formal Methods", Proceedings of the 27th Technical Symposium on Computer Science Education (March 1996): 160-164.
[5] Joyce, D.T. "A Virtual Lab to Accompany CS1 and CS2." SIGCSE Bulletin 22 (March 1990): 40-43.
[6] Palakal, Mathew, Meyers, Fredrick and Boyd, Carla, "An Interactive Learning Environment for Breadth-First Computing Science Curriculum", Proceedings of the 29th SIGCSE Technical Symposium on Computer Science Education, (March 1998): 1-5.
[7] Penny, J.P. and Ashton, P.J. "Laboratory Style of Teaching Computer Science Laboratories." SIGCSE Bulletin 22 (March 1990): 102-106.
[8] Dongsik kim, Kwansun choi, sunheum lee, Hojoon seo , Yonghae kong,"Practical Implementation of a web-based virtual laboratory in the area of electical engineering",Proceeding of the IASTED international Conference COMPUTER AND ADVANCED TECHNOLOGY IN EDUCATION, June 27-29, 2001, Banff, Canada. pp5-7.
[9]Kazuaki goshi, Yong sun, "Student's behaviour sequences in intelligent tutoring system for hoare logic", Proceeding of the IASTED international Conference COMPUTER AND ADVANCED TECHNOLOGY IN EDUCATION, June 27-29, 2001, Banff, Canada. pp130-134