Asian Journal of Information Technology 6 (11): 1132-1136, 2007
ISSN: 1682-3915

Web-Based Practice System for Programming Languages

Duk-Hoi Koo
Department of Computer Education, Daegu National University of Education,
Daegu, 705-715, South Korea

Abstract: The present study is about a web-based programming practice system that enables students to access processing resources in the compiling server through the hypertext transfer protocol (http) and practice programming languages. In the system, the user makes codes on a client through the hypertext transfer protocol (http), the server compiles the codes and the results are provided to the client. Therefore, programming languages are situated in the server rather than in a client and as a result users can use programming languages based on the compiling system of the server and this allows them to practice programming languages effectively through web interface regardless of time and space. In this research, we designed and implemented server-level service structure for providing services to trainees, middleware-level data-driven scheme and a processing system for end users.

Key words: Web-based practice system, programming languages, hypertext transfer protocol

INTRODUCTION

With the popularization of super-speed Internet and the heated trend of ubiquitous environment, information education is introducing conveniences free from the restriction of time and space (Kim, 2004; Shon, 2007). According to previous researches, many researchers argue about the effective programming learning (Hwang and Blandford, 2000; Lee *et al*., 2001).

However, the practice of programming languages has still been difficult because it requires the acquisition of preliminary knowledge and efforts to prepare the environment of practice. Thus, programming language learners have had to practice in a place or environment equipped with platforms and set up with programming languages at a high cost. What is more, although there is detailed guidance on how to use various programming languages and on education programs in many Internet sites, most of the contents are not adequate for learners to attain the effect of practice and furthermore the remote compiling service, with which the user makes a source code in a programming language and sends it to a remote place and gets the results back through a batch process, cannot improve the effect of programming language practice because programming language learners need to check the results as soon as they make a source code. Accordingly, they have difficulties in understanding the structure and characteristics of languages as they cannot examine the output of each source program on the spot and figure out computer processing and in such environment they cannot attain a high effect from practicing programming languages. Thus, in order to overcome such technological limitations and in particular inefficiency resulting from the batch processing of source files, the present study purposed to develop a more effective practice system composed of a compiling server on the network that supplies processing resources to a client on which the learner practices programming languages and a middleware server as an intermediate processor through which the learner can be provided with the output of compiling in real-time while editing source codes.

Related works: Lee Jae-seon (2001) suggested that a Web-based programming system reduces the cost and burden of installing programming language processors by individual learners and allows the learners to practice anytime and anywhere using the characteristic of the Web. Park Seong-jin (2002) also emphasized the advantage of a web-based programming practice system that does not require the installation of a compiler by individual learners.

According to these previous studies, it is necessary to reduce learners' burden of compiler installation for practicing programming languages and thus the present study is focused on satisfying the requirement.

System design: The objective of the present study, is a real-time programming practice system for practicing programming in network environment and the system was designed as in Fig. 1.

Corresponding Author: Duk-Hoi Koo, Department of Computer Education, Daegu National University of Education, Daegu, 705-715, South Korea

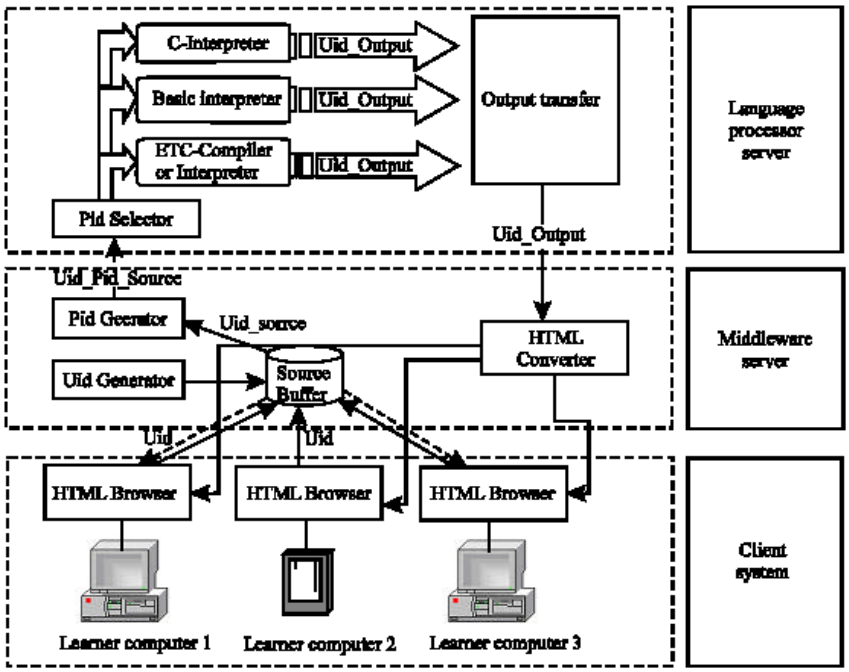

Fig. 1: System design

First, in the environment where various types of client computers (PC, PDA) are used by learners, the client system includes a HTML browser fit for the OS of the client.

The learners' computers do not execute any processor or program separately and the results of compiling are also presented to the learners in the HTML format fit for the computer.

Second, the middleware server includes a source buffer that stores programming language sources temporarily, UID generator that provides a Unique ID (UID) to each HTML browser when a learner's computer is connected to the server, PID generator that distinguishes among languages studied by learners and HTML converter that converts the output of compiling into a HTML document fit for each learner's HTML browser. If a client system (computer) attempts to connect, the UID generator assigns a UID to the HTML browser of the computer through the source buffer. The learner chooses a programming language to practice on the HTML browser and based on the choice, the PID generator produces a Processor ID (PID). The user makes a source code on the HTML browser using the selected programming language and the source code is combined with UID in the source buffer and again with PID and then sent to the language processor server. The processed source code, UID and the output data of processing are sent back to the HTML converter, which forms display data optimized for the learner's HTML browser identified through UID and sends the data to the client system.

Lastly, the language processor server, which has language processors such as C language compiler, Basic interpreter and other compilers and interpreters as demanded by the learners, includes PID selector that forwards the source code with PID from the middleware server to the corresponding language processor and output transfer that combines the output data of compiling and UID and sends the data to the HTML converter in the middleware server.

The process of unique identification: The process of unique identification means the process that a learner edits a source code in a programming language using UID generated by the UID generator and waits for output data.

Figure 2 is the flow chart showing the process of identification. This process allows the user to log on the system and to use two or more browsers at the same time in case the user wants to practice multiple programming languages. If the user does not have a UID, the user is assigned a UID by the UID generator in the middleware server. If the user has a UID, it is checked whether the user has sent a source code and is waiting for the results of processing from the server. This process is for determining conditions when the user uses 2 or more browsers. In case a user is going to edit a new source code, the user is requested to choose a programming

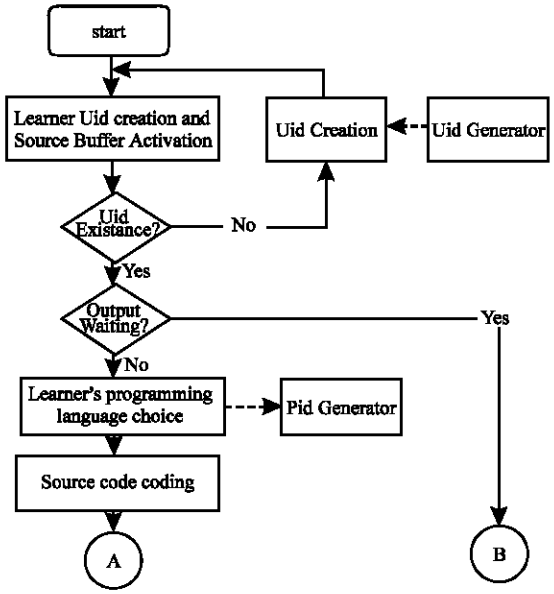

Fig. 2: The process of unique identification

language to practice so that the PID generator in the middleware server generates a PID and then the user edits a source code in the language. In case the user is waiting for output data for a source code that has already been sent, he/she receives the output data from the HTML converter in the middleware server. The learner connects to the designated network site for practice, chooses a programming language and views the output data displayed in the format fit for the user's HTML browser together with the source code.

The process of source code: In the process of source code, the server for compiler service receives a source code together with UID and PID and forwards the code to the corresponding language processor. Then it receives the output with UID and sends it back to the HTML converter.

Figure 3 is the flow chart showing the process that a source code with UID and PID is obtained from the middleware server and sent to the corresponding language processor identified using the PID and the output data and UID are sent from the output transfer to the HTML converter in the middleware server.

The process of output transfer: In the process of output transfer, on receiving the output data with UID from the compiler server, it is checked whether the user is still connected and if the connection has been terminated the data is discarded and if not the data is converted into output format optimized for the user's browser and sent to the browser.

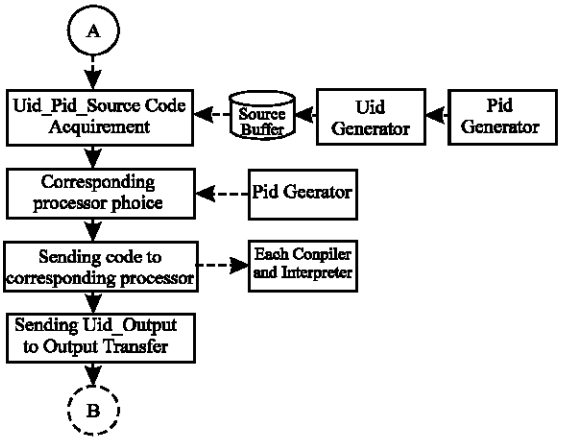

Fig. 3: The process of source code

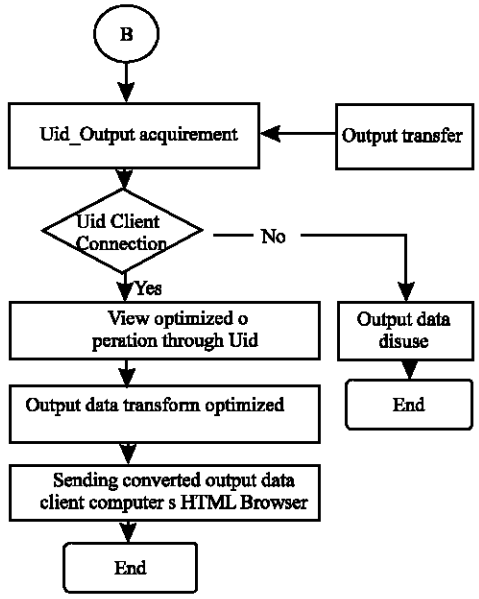

Fig. 4: The process of output tranfer

Figure 4 is the flow chart showing that process that the middleware server acquires output data with UID from the output transfer in the language processor server, checks if the user's computer is still in connection and if it is the server converts the data into the output format and sends to the HTML browser of the client system and if not it discards the data.

System implementation: Figure 5 shows the output display screen of the browser in a client computer that implemented the real-time programming language practice system designed in this study.

In the implemented system, all processes including the editing and completion of a source code in a programming language and the finding of errors in the

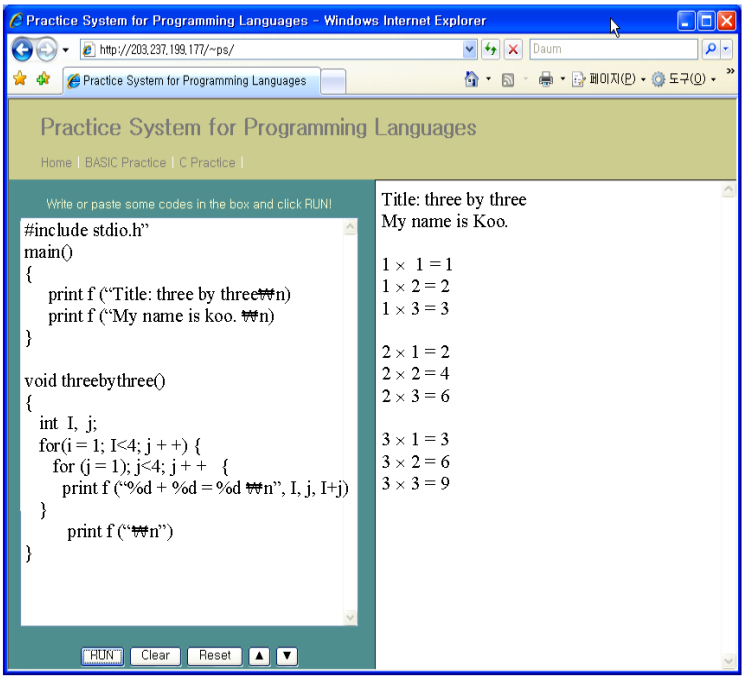

Fig. 5: System implementation on PC

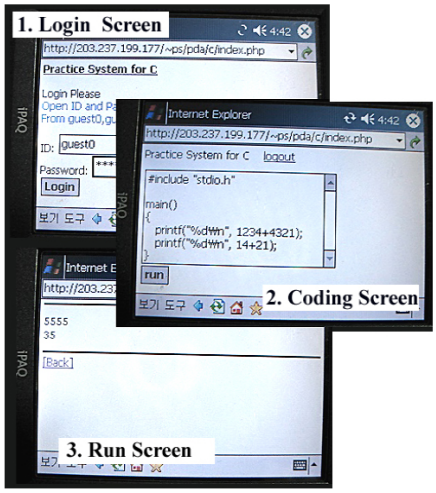

Fig. 6: System implementation on PDA

source code are displayed effectively as if the language processor is installed on the learner's computer and this solves the problem in existing systems that a source code is processed in batch and only the result of compiling is provided to the learner. What is more, the HTML converter converts output data fittingly not only for ordinary HTML browsers but also for PDA HTML browser with a small display as in Fig. 6.

CONCLUSION

The client/server-based real-time programming language practice system assigns UID to each learner when the learner logs on the system and provides the user with compiling service by request through the client terminal. The multiple-compiling service of the server, which provides programming practice environment and the intermediate conversion system, which converts output data into a HTML document, enable real-time execution of programming language practice environment on the learner's computer. Using the system, program language learners can study languages and be provided with various real-time services including compiling for language practice, which is the main purpose of the system, without understanding the system platform. What is more, with the system, language learners do not need to install compilers or to know how to use the compilers and they can practice programming languages on any computer as long as the hypertext transfer protocol is supported. Furthermore, the system is economically efficient as the learners do not need to install an additional compiling system and it allows a larger number of users to practice programming languages concurrently and this is expected to contribute to the efficient education of high-quality program developers.

REFERENCES

Hwang, D.J. and D.K. Blandford, 2000. A multidisciplinary team project for electrical engineering, computer engineering and computer science majors. In Proceedings of the ASEE Annual Conference, St. Louis, Missouri, USA, Session 2553.

Kim, Chisu, 2004. The Development of the Distance Education System for PDA Applied to Making Internet Software. J. Korean Assoc. Inform. Edu., 8: 91-100.

Lee, Jaeseon, 2001. Design and Implementation of a Web-Based Programming Practice Environment for BASIC. Master's Theses, Sookmyung Women's University.

Lee, T.W., I.H. Yoo and C.H. Lee, 2001. ICT Educational Theory. Seoul: Hyung seul Press.

Park, Seongjin, 2002. Tutor System for BASIC Programming Education on the WWW. Master's Theses, Daegu National University of Education.

Shon, Mi, 2007. Possibilities and Considerations of Experiential Learning in Ubiquitous Learning Environments. Res. Inst. Curri. Inst., 11: 143-172.