

Compiler on the Web: Remote Laboratory for e-Learning (IEC2012)

N. Manus¹ and P. Chongstitvatana²

¹Department of Computer Engineering Chulalongkorn University, Bangkok, Thailand
(korn007_751@hotmail.com)

²Department of Computer Engineering Chulalongkorn University, Bangkok, Thailand
(prabhas@chula.ac.th)

ABSTRACT

In learning computer programming, an important aspect is about acquiring programming skill. Students must be able to write, compile and test their programs and see the results in order to gain that skill. Most of supporting software for e-Learning is available as standalone software installed on a PC. However, for the ease of use, for uniformity and up-to-date maintenance, the software as web applications are most suitable. We present a case study of a compiler to support learning embedded system programming. The students can access the compiler via a web browser. The target machine is available online for testing of students' program. This is a remote laboratory that support e-Learning. It unifies the learning environment. The laboratory experiments can be performed remotely and inexpensively.

Keywords

Compiler, e-Learning, Remote Laboratory.

1) INTRODUCTION

To acquire programming skill, tools must be available for students to practice and to get immediate feedback. For a normal laboratory classroom, personal computers provide invaluable tools for programming practice. Most programming classes taught popular programming languages such as Java and C. The compilers are easily available for such machines. The "off-line" mode of learning programming is the most widely used method because of

the simplicity in its arrangement. However, for a specialized class in programming such as programming for an embedded system, the equipment may not be available in quantity. The reason may be the cost or the need for special equipment. Such laboratory session usually includes the set up for external peripherals such as the motion of motors, the sound and the sensors. These peripherals are attached to the computing platform. Students practice on this kind of platform requires feedback through watching the behavior of the peripherals. It is possible to share this kind of environment to several students in the on-line fashion. Students can watch the peripherals through the video. They can practice programming and experiment with the system through the web.

2) THE ENVIRONMENT

We present a case study of a compiler to support learning embedded system programming. The students can access the compiler via a web browser. The target machine is available online for testing of students' program. We will call the station for students a "workstation." A workstation includes an embedded system board, a PC for controlling the board, a camera to observe the result. The PC also works as a server that provides access to the workstation.

The embedded system board is a 32-bit multi-core embedded processor. The processor, named Propeller, is made by Parallax [1]. It has eight independent processors. These processors can work simultaneously and independently with a

central shared memory. We are interested in this processor because it represents a high performance but low cost system that can be used in industry. Programming this new system also posts a challenge. Therefore it becomes a target of our classroom on the web.

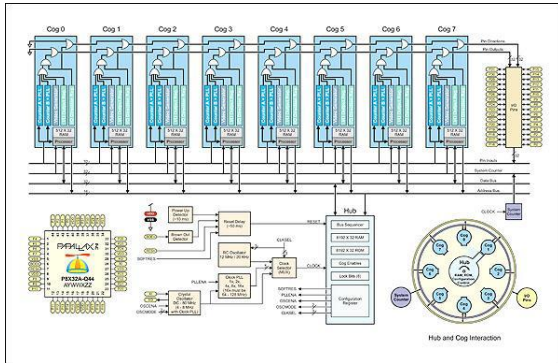


Figure 1: propeller block diagram

To program this system, we modify our existing teaching language, Rz [2]. A compiler has been written to target Propeller (Rz has quite a few other target machines). To give a flavor of the language, a short introduction to the language will be given here. Rz has a C-like syntax. The language is a subset of C without type. There is only one type, Integer. There are two kinds of variables: global and local. Global variables have to declare before use. Local variables are automatically declared. The array of data can be declared as a global variable. The array can only have one dimension. Various kind of operators are available: +, -, *, /, ==, !=, <, <=, >, >=, !, & &, ||, * (Dereference), & (address). For example, Program adding number in Array RZ as follows.

```
main (){
    sum = 0;
    for( l = 0 ;i< N ;i++ ) {
        sum = sum + a[i] ;
        print ( "Sum is ", max );
    }
}
```

When using the parallel compile (Option added in to compile the RZ) can be adjusted to this (Pseudo Code).

```
for (j = 1 ; j < log2N; j++) {
    for all CPUs in Properller {
        if ((k + 1) mod 2j) = 0 {
            x[k] := x[k - 2j] + x[k]
        }
    }
}
```

We will not discuss the compiler for parallel computation here. The main aim of this paper is to illustrate the work on wrapping the "workstation" and make it available to the web.

3) DISTANCE-LEARNING REMOTE LABORATORIES [3]

The remote laboratory is defined on the basis that the operation of experiments are conducted remotely by computer. Students do not need to be demonstrated at the real lab but they can access and control via the network medium. The operation of the remote laboratory is shown in Figure 2. The workstation can be accessed remotely and students perform laboratory experiments in the laboratory via the Internet. Setting up the propeller experiments can be performed by different people and they can test the experiment in their own time.

All results are stored in a database that can be retrieved, analyzed and displayed later.

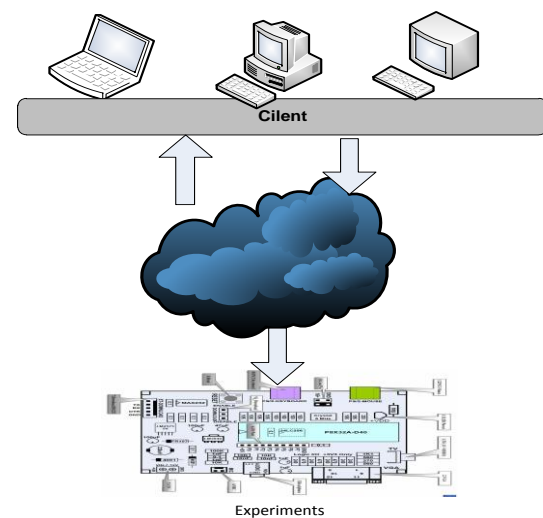


Figure 2. Network Control of a Remote Laboratory Model

2.2.2. Distance-Learning Remote Software Description.

The software of remote operation focuses on a web-application and works for the user interface and controls of the laboratory. They are shown in Figure 3.

The components of the remote laboratory are Java web-based infrastructure, which are created by EJB [4] application.

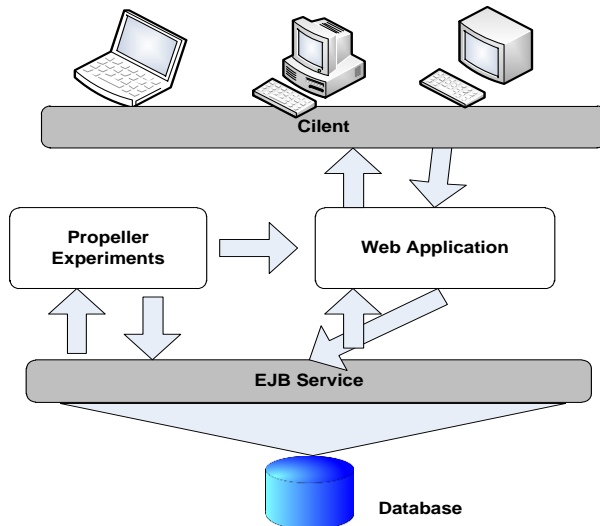


Figure 3. The Software Architecture.

Students access to workstation laboratory and code the program in a web page. It consists of the text area, button, and other components to accommodate the students to work easily (Figure 4).

Code Edit

Please insert code here:

Compile

Compile result

Figure 4. Webpage for coding

Students submitted the code to a compiler. The RZ compiler's command is called by EJB service and the compiled file is stored in database. The output of the compiler will be displayed in webpage and is stored into database.

Students can execute the compiled file for testing the program. The Propeller's command is called by EJB service. The result of program will be recorded in a video format. It is also stored in database. Students can display the result in webpage and they can post it into social network (Facebook, twitter) or web-board.

2.3) Conclusion

The concept of the laboratory over the Internet has been proposed and realized. The prototype for the software has been implemented in Java. The workstation was installed. Students can access to the workstation from the web. They can conduct experiments and collect data. The ability to access remotely is very important for e-Learning.

REFERENCES

- [1] P. Chongstitvatana, P. (2011, September) RZ language. [Online]. <http://www.cp.eng.chula.ac.th>
- [2] Parallax Inc. Web Site. [Online]. <http://www.parallax.com/>, Retrieved 2011, October.
- [3] Lieberman, D. and Cheung, T., "Distance Learning and Remote-Controlled Laboratories", Abstract, Conference on Computing in the Disciplines, March 2000.
- [4] Enterprise JavaBeans 3.1, <http://jcp.org/aboutJava/communityprocess/final/jsr318/index.html>