PROGRAMMING LANGUAGE TEACHING ENVIRONMENT BUILDING AND TEACHING EFFICIENCY EVALUATION

[1]CHUNAG JAN CHANG, [2]MING HONG LIAO

[1,2]Ming Chi University of Technology, New Taipei City, Taiwan (R.O.C.)
E-mail: [1]cjchng@mail.mcut.edu.tw, [2]m01158015@mail2.mcut.edu.tw

Abstract- An automatic online assessment procedure for classing in C programming language lab was developed. The self-developed software program, integrated & associated with GPL open source utilities, graded students' echoes from students in no delay to profile learning status of each student, and alert simultaneously whom were in absented mind. Consequently, the teacher could be based to evaluate if students acquire the teaching contents. Under such real time intervention, the teacher would step forward the teaching pace with confidence or put immediate supplement to some troubled contents. After years' aggregation the students' achievements, the big data have been able to objective assessment of teaching methods and to enhance student's learning effectiveness.

I. NTRODUCTION

C programming lab is fundamental in the engineering department of the fresh year in the undergraduate level. Usually it is conducted in a classroom where installed Internet-linked computer cluster. While classing, the teacher couldn't catch what students were doing behind the screen.

They might be playing computer games, surfing on net, chatting with other students, sleeping...etc. Practically, the academicals level of the recruited students are various very much; some had ever learned C-Language in high school and some are total blind to any programming language. It is not suitable to evaluate the learning or teaching effect based on the outcome of the middle or final exam. The teacher has to promote methods in time per the real situation. Therefore, we launched a series of teaching environment reform and effectiveness evaluation of student learning

1.1. DRBL (Diskless Remote Boot in Linux)

DRBL is a diskless classroom server that developed by NAR Lab. It assemble by many kinds of services (NIS NFS、DHCP、TFTP….etc.). By using this server, we can do not only control this classroom but also provide bridging course of " NCLA License " and centralized storage student's code. Based on this feature, we also created a server that help us judging student's program.By building student's grade database, we can observe student's learning efficiency and the outcome of changing teaching methods.

1.2. PC^2

It's developed by PC^2 team at CSU. It's very common using in many programming contest. It's well known as cross-platform judgement and supported multiple programming-language. Though this program seems like a good solution of this research but based on these two problems, we still have to rebuild the teaching environment.

1. It wasted too much time to reset PC^2 if teacher wanted to judge student's program.
2. Even though PC^2 can make a grade transcripts but it's not automatically. Our ideal is Teaching-Automatically classroom, such as student's program judgement, grade transcripts database.

The emphasis of this article is not only just created a program but also built an automatic teaching classroom system. By accessing the outcome of student's grade curve, teacher can improve the teaching methods.

1.3. Correcting Program

This system was started at 2010. At the very beginning, it was trying teaching invisibility at class and kept on modifying. We create this system in order to enhance "Observation" (To grasp learning situation) and "Controlling" (To point out mistakes and distractions). We hope to improve the effect of "Teaching" and "Learning" further. After many trial and error, this system finally worked online last year, and teacher finally controlled whole class alone without T.A. (Teaching Assistant).

In DRBL server, student's program and code are all restored so that our program can directly access student's program and judge immediately. By comparing the result of teacher's and student's program, we have 7 states to describe student's program.

STATE0	error: output syntax isn't correct
STATE1	error: no project folder
STATE2	error: no binary
STATE3	error: no execution file
STATE4	error: answer is too long
STATE5	error: answer isn't correct
EXACT	

After judgement, the result will output to student's file directory immediately, so that student can know their state and finished time.

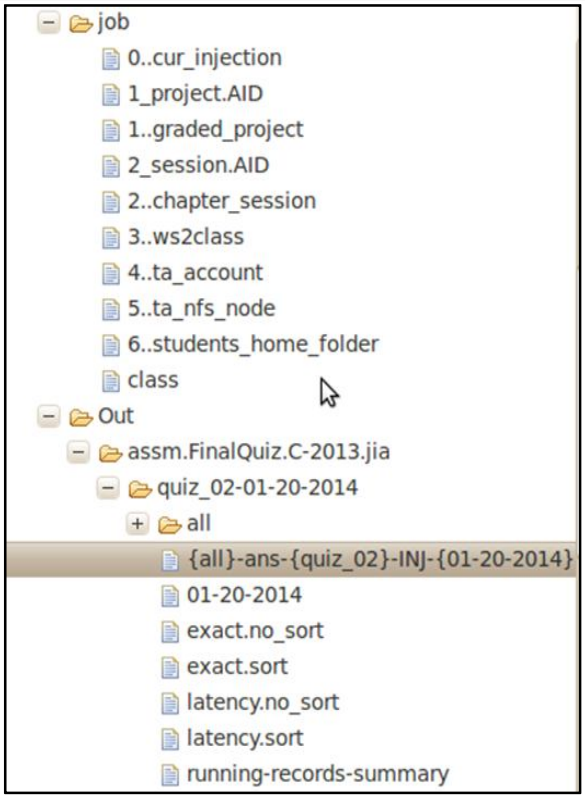


Fig1. is user interface under eclipse. "Job" directory input parameters, and "out" directory output the judgement result.

" job" user interface

0..cur_injection	judgement result output file name
1_project.AID	judge able project name of this chapter
1..graded_project	choose project
2_session.AID	all chapter of this semester
2..chapter_session	choose chapter
3..ws2class	choose class
4..ta_account	choose user who provide answer
5..ta_nfs_node	answer directory of user
6..students_home_folder	student's project directory

" Out" output result

{all}-ans--{quiz_02}—INJ—{01-20-2014}	show student's code and state
exact.no_sort	students of correct answer
exact.sort	students of correct answer and sort
latency.no_sort	show students that have answer
latency.sort	show students that have no answer
running-records-summary	all students state

II. DETAILS EXPERIMENTAL

2.1. Teaching effectiveness analysis:

C programming Language is an obligatory course in first grade last semester of MCUT, and the textbook is "C how to program 6/E". We had summarized the curse last four year and excerpted as follow.

In 2011~2013, our program just online, teacher arranged an exam as final achievement, and problem 1 is ungraded.

2.2. 2011~2012:

In this year, we started using Eclipse CDT as IDE. Teacher assigned project name and made students to do the project, and TA controlled the program to judge student's program. The judgement result is that no one got exact state. It doesn't make sense, so we judged student's program artificial and we found that about 75% student's program can't be found. We checked further and found that 80% of students among them created wrong project name, and 14% of them set to the wrong project path, and just only 6% didn't do the project. After this exam, we found that all students will set the wrong project path carelessly. It's very common if we teach students to use a new IDE so we have to improve our teaching methods that make students know well about using Eclipse CDT.

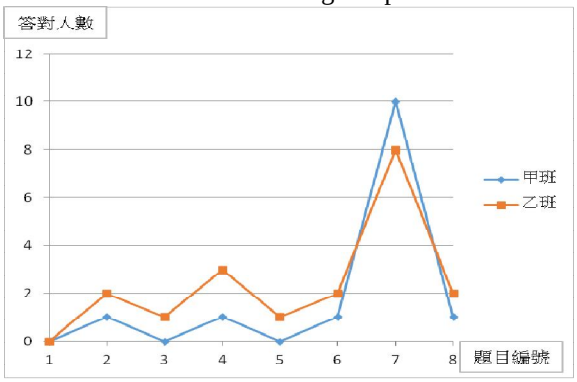


Fig2. 2011 The number of answering student

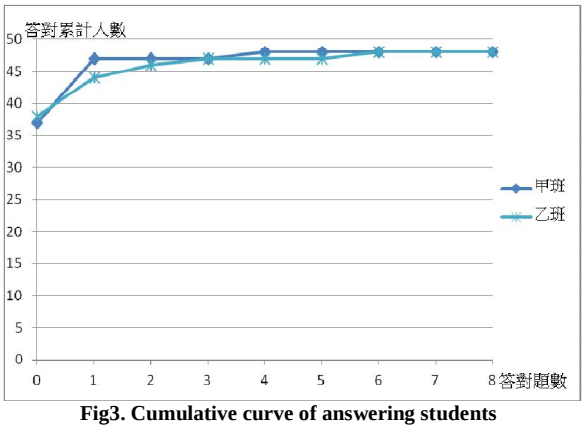


Fig3. Cumulative curve of answering students

2.3. 2012~2013:

In this year, we started to judge student's program every class. TA created shell script that could judge in time, but this work required two or more people so TA had to join every class. By this semi-automatic way, we could correct students' mistakes immediately. In this year, we extended curse to "Pointer" chapter but the system still not stable wherefore the examination question of "Pointer" were recessive and students could decide to use pointer or not and more applied problems that made students thinking about program way.

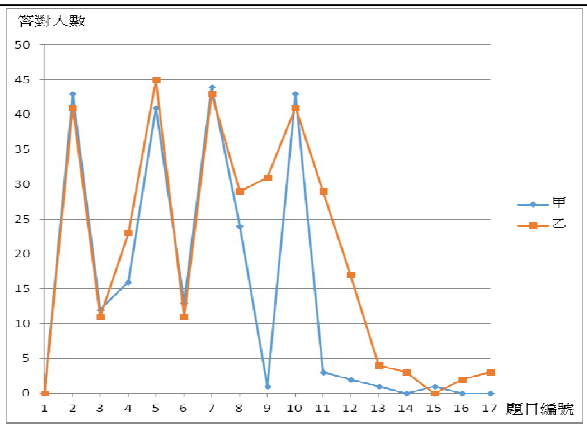


Fig4. 2012 The number of answering student

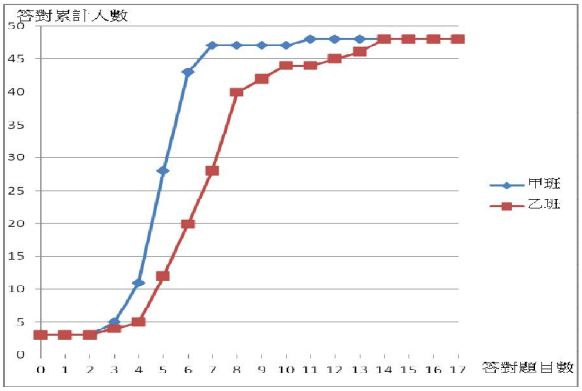


Fig5. Cumulative curve of answering students

2.4. 2013~2014:

In this year, our program had user interface so teacher could judge student's program on his own, and TA just needed to tutor students. We had taught all chapter in this year.

The examination questions in this year were all applied problems include "C Pointer", "C Characters and Strings", "C Formatted Input/Output"...etc. And we clearly asked students to use C Pointer to do the project, and some data structure but student's answer rate about "C pointer" were very low so we found that "C Pointer" is an obstacle of learning C Language. Our teaching schedule was much substantial lead over the years so our program and teaching methods were better over the years but it's not enough. We found that the students in the top of the score had learned about C Language in high school so students were very different from the level distance what that mean is even though we realized teaching automation but students learning efficiency still not very well.

We want all student could understand during class but not spent much time review after class because MCUT has very special system that crowd curse so we still needed to improve our teaching methods to save student's review time.

In Fig6 and Fig7 we can know that two achievement are close because we taught all chapter in this year so we found some shortcomings of our program and we will enhance it

In Fig4 and Fig5 we can discover that the achievement of class B is better and that's because class B attended class late in a week wherefore the problems of teaching will first meet in class A and we had solved. It's not fair to class A but this issue help us to improve teaching methods to prevent from endowing class A's rights and interests.



Fig6. 2013 The number of answering student

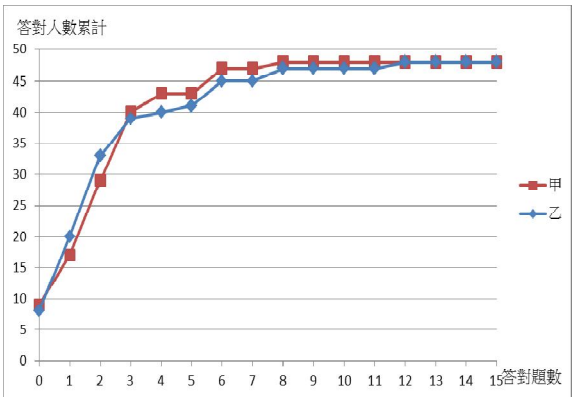


Fig7. Cumulative curve of answering students

2.5. 2014~2015:

According to the outcome last year, teacher put "C Pointer" and the rest chapter to advance class. We should lay the foundation to students so we had regular quiz every class to force students focus on class and no final exam that saved students review time after class.

2.6. 2015~2016:

Teacher always met a teaching problem over the years: students don't familiar with Eclipse CDT and Ubuntu. Ever since they were little, they all use Microsoft Windows so they don't familiar with Linux, and they always got state 1(no project folder). Students always complain about: I had done project, why you can't judge my project.

But learning outcomes is not all about students, many students didn't ever learn about any programming language so teacher must make students interesting in it. We review the class in this year and we decided to make transcripts after every class next year to prevent from students complain every end of semester about: "I worked very hard every class", "I had done every project, your program fail" to gain higher grade.

According to 2013~2014 experience, to prevent "native" students from occupying high score we focus

on teaching C Language statement because we discover that even though we had many applied problems in exam but students problem-solving approaches are still very simple and ineffective so we speculate that students learning efficiency isn't well. The examination questions this year are focus on Language statement and we force students to answer question by specifying methods.

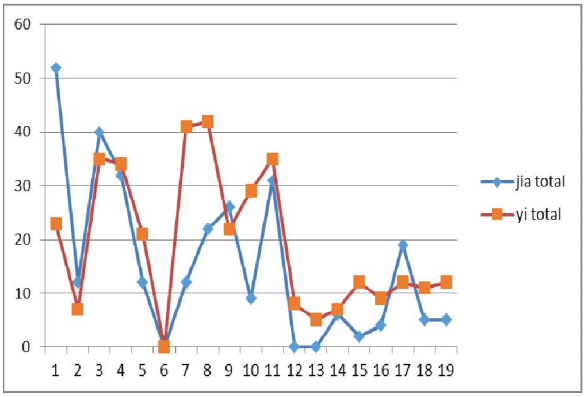


Fig8. 2015 The number of answering student

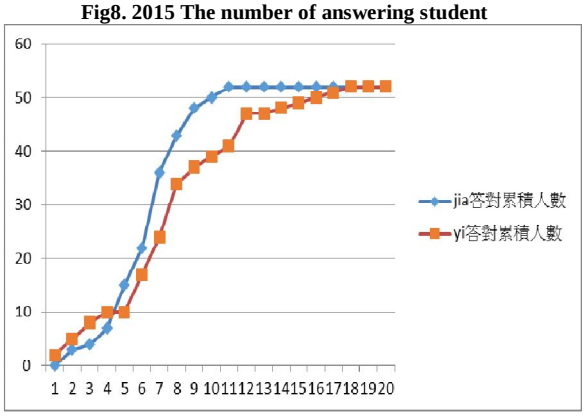


Fig9. Cumulative curve of answering students

CONCLUSION

We tried this new teaching methods this year and we plan to focus on these point next year:

1. User manual of class environment.
2. Stabilize servers of classroom.
3. Teaching basic C Language statement to observe students learning efficiency, and application Tips leave to "Advance C Language" and "Data Structure" class.

4. Transcripts of every class.

The achievement of class B always a little lower than class A over the years. That's because class B's curse schedule always conflicts with national holiday, and this issue need to consult with department. TA can only have curriculums every Wednesday to complement the gap between two classes.

REFERENCES

[1]. Chuang-Jan Chang, Shu-Lin Hwang, Chih-Chun Lin, Chiang-Wei Huang, Tsung-Kai Chou, Ming-Hong Liao, "Using DRBL to Improve the Classing Efficacy of C Programming Language", Proceeding of 2013 Consumer Electronics Forum(CEF2013), 新北市, 中華民國, 2013/5/17。

[2]. Using DRBL to improve learning environment of computer classroom, 林志駿, advisor：Chuang-Jan Chang, Shu-Lin Hwang

[3]. 2012/06/30

[4]. 蕭志榤 黃國連(2003)，自由軟體教學環境集中管理的自由軟體教學環境DRBL (Diskless Remote Boot in Linux)簡介。國家高速網路與計算中心 開放原始碼計畫。

[5]. DRBL DRBL 簡介 上網日期：2016.04.01 取自：http://drbl.nchc.org.tw/introduction/

[6]. PC 2 PC 2 wiki 上網日期：2016.04.01 取自：http://pc2.ecs.csus.edu/wiki/Main_Page , https://en.wikipedia.org/wiki/PC%C2%B2 ,

[7]. C how to program 6/E by Harvey M. Deitel (Author), Paul Deitel (Author) ISBN: 0132404168 © 2010, pp. 966

[8]. The Measurement and Prediction of Teaching Efficiency: A Summary of Investigations A. S. Barr The Journal of Experimental Education Vol. 16, No. 4, The Measurement and Prediction of Teaching Efficiency: A Summary of Investigations (Jun., 1948), pp. 203-283

[9]. Measuring teaching efficiency in higher education: An application of data envelopment analysis to economics graduates, from UK Universities 1993 Department of Economics, Lancaster University, Lancaster LA1 4YX, UK Received 25 February 2003, Accepted 2 February 2005, Available online 17 May 2005

[10]. The Prediction of Teaching Efficiency, Leo Joseph Lins, The Journal of Experimental Education, Vol. 15, No. 1, The Prediction of Teaching Efficiency (Sep., 1946), pp. 2-60.

★★★