Vitalii Bazurin
Kovpak Hlukhiv Agrotechnical Institute of Sumy National Agrarian University, Ukraine
ORCID: 0000-0002-6614-4889
vbazurin@gmail.com

THE PEDAGOGICAL CONDITIONS OF SUCCESSFUL SYSTEM PROGRAMMING STUDYING STUDENTS OF THE SPECIALITY "COMPUTER ENGINEERING"

ABSTRACT

System programming is an important component of the professional training of specialists specializing in "Computer Engineering". Various factors and pedagogical conditions affect the learning outcomes of student learning in Assembler. The article outlines those which, according to the author, are the most important: initial knowledge and skills of students; programming environment for Assembler language; means of training; training method; motivating students to study programming at Assembler.

Formulation the problem. *The role of computer networks in human life continues to grow. Computer networks are used in most industries. Specialists of the specialty "Computer Engineering" carry out planning, installation and maintenance of computer networks at enterprises, organizations, and settlements. Specialists of this specialty are in great demand in the labor market. However, the pedagogical conditions for successful learning of system programming remain insufficiently defined.*

Materials and Methods. *Analysis of research by scientists and methodologists, analysis of curricula, analysis of pedagogical experience.*

Results. *The importance of system programming is indicated in the computer science education professional program. Based on the analysis of own pedagogical experience, it is established that the success of learning system programming depends on many factors: the level of logical and algorithmic thinking of students, the ability to create programs using basic algorithmic constructions, motivation, the use of optimal programming environment, control by the teacher.*

Conclusions. *For successful training of students of the specialty "Computer Engineering," it is necessary to compliance system programming with the following pedagogical conditions: sufficient level of knowledge and skills in disciplines; choice of optimal programming environment; application of auxiliary means of training; appropriate training method.*

KEY WORDS: *system programming, maintenance of computer systems and networks, Assembler, pedagogical conditions, training, computer engineering.*

INTRODUCTION

The role of computer networks in human life continues to grow. Computer networks are used in most industries. Specialists of the specialty "Maintenance of computer systems and networks" carry out planning, installation and maintenance of computer networks at enterprises, organizations and settlements. Specialists of this specialty are in great demand on the labor market.

However, the pedagogical conditions for successful learning of system programming remain insufficiently defined.

In 2009, at Kovpak Glukhiv Agrotechnical Institute (Sumy National Agrarian University) began training specialists in the specialty "Maintenance of computer systems and networks". Now this specialty was renamed to "Computer Engineering". In the past, 6 issues have been made. On the basis of analysis of the developed pedagogical experience, the main conditions and factors on which the successful study of the discipline "System programming" students by the specialty "Computer Engineering" was determined.

Analysis of relevant research. There are a number of scientific findings related to the study of programming in low-level languages. For example, Hieu Vu (2016) proves the importance of learning low-level programming in the language Assembler. When creating programs in the language Assembler, the programmer better understand the principles of the operation of the processor, RAM. The scientist demonstrates examples of programs in the language of Assembler. In his view, the study of the language Assembler is not significantly different from the study of other programming languages (Vu, 2016).

The research (Chau && Fung, 2009) reveals the structure and principle of the environment for self-education of students of system programming in the language Assembler. Scientists have developed the appropriate equipment and software. In the course of research, the following software requirements are defined: user-friendly user interface and availability of instructions

for laboratory work, the ability to configure the program, support the equipment, the ability to generate assembling code. The developed software is implemented in C # language (Chau && Fung, 2009).

O. L. Levoshko and L. V. Konstantinov focus on the technical aspects of using the Assembler language in system programs (Levoshko && Konstantinov, 2007).

P. T. Dederkal discloses features of the initial stage of studying system programming in the language of Assembler:
1) familiarization with the operations carried out with binary and hexadecimal numbers;
2) forming an idea of the representation of negative numbers in addition to the code;
3) formation of the idea of storing data in the registers of the processor;
4) familiarization with the simplest Assembler commands (exchange, logical, arithmetic, bias);
5) familiarization with comparison commands (Dederkal, 2016).

P. T. Dederkal proposes to use the integrated FASM programming environment, since it contains the following tools: text editor, compiler, debugger (Dederkal, 2016).

The pedagogical conditions and factors on which the successful training of programming in Assembler language depends is not sufficiently developed.

The aim of the article. To clarify the pedagogical conditions for successful training in system programming of students of the specialty "Computer Engineering".

RESEARCH METHODS

Analysis of research by scientists and methodologists, analysis of curricula, analysis of pedagogical experience, analysis results of pedagogical experiment.

RESULTS

Formation of professional competences of future specialists in the specialty "Computer Egineering" in the process of studying the discipline "System programming".

The discipline "System programming" belongs to the disciplines of the cycle of professional training. In accordance with the educational-professional curriculum (Standard of a higher educational institution, 2013), 108 hours is given for the study of this discipline.

According to the educational-professional curriculum, the following skills of future specialists of the specialty "Maintenance of computer systems and networks" as a result of studying the discipline "System programming" should be formed:
− to master Assembler language;
− create multimodal programs;
− use static and dynamically linked DLLs;
− create own libraries;
− to organize intermodular interactions and interactions with libraries of the programming environment;
− handle interruptions, intercept and handle exceptional situations;
− to program interaction with the equipment, organize low-level and high-level I/O;
− to process dynamic data structures (Standard of a higher educational institution, 2013).

The discipline contains the main modules:
1) tools for building system programs;
2) the development of system programs (Standard of a higher educational institution, 2013).

The first module contains the following topics:
1) Assembler language as a means of effective programming;
2) architecture and command system of the base Processor;
3) programming subprogramms in the language Assembly;
4) technologies of development of multimodal system programs;
5) use of software libraries;
6) processing of data structures in system programs (Standard of a higher educational institution, 2013).

The second module contains the following topics:
1) programming table and graph processing in system programs;
2) programming of transformations in broadcast programs;
3) basic concepts of grammar theory;
4) the basics of lexical and parsing programming;
5) types of semantic processing in translators;
6) construction of control program elements (Standard of a higher educational institution, 2013).

This discipline is taught for 7 years.

Pedagogical conditions and factors influencing the success of the system programming study. The discipline "System programming" is studied after the disciplines "Programming", "Operating systems" and "Computer Architecture". Based on the analysis of the teaching experience of this discipline, certain requirements were set for the level of knowledge and skills of students who are beginning to study system programming.

Students who are beginning to study system programming should know:
− the basic notions of the Boolean algebra;
− concept of the algorithm;
− basic algorithmic constructions;
− high-level programming language syntax (C ++, C #, or other);
− principles of data exchange between variables and subprogramms;

− the concept of flows and processes;
− principles of representation of numbers in positional and non-positional systems of calculation;
− the principles of translating numbers from one system to another;
− features of the architecture of processors IA-32;
− peculiarities of functioning of peripheral devices (keyboard, mouse, monitor);
− components of the Windows operating system;
− features of file systems FAT32, NTFS.

Beginning with the study of system programming, students must be able to:
− create programs in a high level programming language;
− to develop algorithms of programs using the basic algorithmic constructions: follow, branch, repetition;
− to develop programs with the output of the data on the screen;
− develop programs for reading data from a file and storing data into a file;
− develop programs using routines;
− organize the data exchange between the main program and subprograms;
− write numbers in different systems of calculation;
− convert numbers from one computing system to another.

Consequently, for successful mastery of system programming, students must have a certain level of input knowledge and skills.

The pedagogical experiment was conducted in 2017-2019 at the Hlukhiv Agrotechnical Institute.

To find out the impact of the level of knowledge and skills in programming on the success of studying system programming, we performed control work: at the beginning of the semester of programming, at the end of the semester - from system programming. The results of both controls were compared using Pearson's correlation. It is established that there is a reliable correlation between programming and system programming test scores. The correlation coefficient is 0.56.

The following pedagogical conditions are the availability of appropriate teaching aids. Learning tools are divided into two groups: basic and auxiliary. The main means of training include a computer with system software, an environment for programming in Assembler language. In our opinion, while studying system programming, you need to use integrated programming environments for Assembler: MS Visual Studio, FAsm, SAsm, and others. These environments have a friendly-looking interface and provide an overview of the process of creating the Assembler program: from typing the program to compilation into an executable file.

The use of an integrated programming environment has the following advantages over traditional Assembler programming tools:
− simplification of the process of development of system programs;
− a friendly orientation programming environment interface;
− allocation of errors in the code of the program;
− the availability of data output tools that work correctly on the Windows operating system;
− availability of software libraries for modern versions of Windows.

In addition to these programming environments, the ideone.com network-oriented environment (Ideone, 2019) was used in the learning process. However, using this environment, it is expedient to learn only Assembler Syntax. This environment is limited to study the features of the program directly from hardware.

Students chose Microsoft Visual Studio as IDE. This is a modern programming environment that students are introduced to while studying programming. 80% of those surveyed found Microsoft Visual Studio the best IDE and worked in it, while others worked in SASM.

Auxiliary means of teaching system programming include: textbooks, manuals, articles on the sites, instructions for laboratory work, tests.

In our opinion, the textbook on system programming (Kravets && Rysovany, 2008), almost completely corresponds to the educational-professional program of the specialty "Maintenance of computer systems and networks". It is advisable to apply this tutorial for teaching system programming students. This tutorial contains all the necessary theoretical material, examples of programs in Assembler language and instructions for laboratory work.

In our opinion, the success of teaching students of system programming also depends on the chosen learning methods. These methods include:
− the method of expediently selected tasks;
− the method of demonstration cases;
− the method for finding errors in the finished programs.

The next important factor affecting the outcome of the Assembler study is the student's motivation. Assembler programming language is limited (compared to C ++, C #, or Java). This negatively affects students' motivation to learn Assembler.

The students' motivation level was determined using the Ehlers success motivation test. We compared students' level of motivation and their level of knowledge and skills using Pearson's correlation coefficient. The correlation coefficient is 0.9. Therefore, there is a strong direct link between students' motivation and their level of knowledge and skills in system programming.

Teachers can not influence the degree of use of Assembler by programmers when writing applications. However, students can be interested in the development of modules and dynamic libraries for applications.

In graduation thesis, in our opinion, it is advisable to formulate such themes of projects that students develop application programs. Also, students must develop (or use ready-made) dynamic libraries written in Assembler.

CONCLUSIONS

Therefore, for successful training of students of the specialty "Computer Engineering" it is necessary to compliance programming in Assembler language with the following pedagogical conditions: sufficient level of knowledge and skills in disciplines; choice of optimal programming environment; application of auxiliary means of training; appropriate training method.

This study can be continued in the following directions:

− the development and experimental verification of the effectiveness of the methodical system of training students of system programming in Assembler language;

− the development of the subjects of diploma design associated with the creation of programs that use sub-programs or libraries in Assembler language.

References

1. Vu, H. (2019) The Forgotten Assembly Programming Language. Journal of Scientific and Engineering Research, 3(1), 17-20, 2016. http://jsaer.com/download/vol-3-iss-1-2016/JSAER2016-03-01-17-20.pdf, Last accessed: 2019/01/31.
2. Chau, Chi-Fai, && Fung, Yu-Fai (2007) A Tool for Self-Learning Assembly Language Programming and Computer Architecture: Design and Evaluation. Computer applications in engineering education, 15 (1), 41-54 (2007) http://www3.interscience.wiley.com/journal/116835322/issue, Last accessed: 2019/01/31.
3. Levoshko, O.L.& Konstantinova, L.V. (2008) The programming language Assembler in tasks of system programming of the operating system Windows. Collection of scientific works of Kirovohrad national technical university in agricultural production, branch engineering, automation. Kirovograd, KNTU, 2007, 19, 81-85.
4. Dederkal, P.T. (2016) Assembler: the beginning of programming. Abstracts of the VIII International Scientific and Technical Conference "Informational and Computer Technologies 2016", Zhytomyr, April 22-23, 2016. Zhytomyr: ZhDTU, 2016: 228. (2016). https://conf.ztu.edu.ua/wp-content/uploads/2016/06/228.pdf Last accessed: 2019/01/31.
5. Standard of a higher educational institution (2013) Educational and professional program. Direction of preparation 6.050102 "Computer Engineering". Specialty 5.05010201 "Maintenance of computer systems and networks". Alexandria, 52 p. (2013).
6. Ideone.com – Online Compiler and IDE. https://ideone.com/ Last accessed: 2019/01/31.
7. Kravets, V.O., Rysovany, O.M. (2008) System programming. Assembler for Win32 API. Tutorial. Kharkiv, NTU "KPI", 512.

ПЕДАГОГІЧНІ УМОВИ УСПІШНОГО НАВЧАННЯ СИСТЕМНОГО ПРОГРАМУВАННЯ СТУДЕНТІВ СПЕЦІАЛЬНОСТІ «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

Віталій Базурін

Глухівський агротехнічний інститут ім. С.А.Ковпака Сумського національного аграрного університету, Україна

Анотація.

Системне програмування є важливою складовою професійної підготовки фахівців, що навчаються за спеціальністю "Комп'ютерна інженерія". Різні фактори та педагогічні умови впливають на результати навчання студентів програмування на мові Асемблер. У статті окреслено ті, які, на думку автора, є найважливішими: початкові знання та вміння учнів; програмування навколишнього середовища для мови Асемблер; засоби навчання; метод навчання; мотивація студентів до вивчення програмування в Асемблері.

Формулювання проблеми. *Роль комп'ютерних мереж у житті людини продовжує зростати. Комп'ютерні мережі використовуються в більшості галузей промисловості. Планування, установка та обслуговування комп'ютерних мереж на підприємствах, в організаціях та населених пунктах здійснюють фахівці спеціальності «Комп'ютерна інженерія». Спеціалісти цієї спеціальності користуються великим попитом на ринку праці. Однак педагогічні умови для успішного засвоєння системного програмування залишаються недостатньо визначеними.*

Матеріали і методи. *Аналіз досліджень вчених і методистів, аналіз навчальних планів, аналіз педагогічного досвіду, аналіз результатів педагогічного експерименту.*

Результати. *Важливість системного програмування зазначена в освітньо-професійній програмі спеціальності «Комп'ютерна інженерія». На основі аналізу власного педагогічного досвіду встановлено, що успіх навчання системного програмування залежить від багатьох чинників: рівень логічного та алгоритмічного мислення студентів, сформованість уміння створювати програми з використанням основних алгоритмічних конструкцій, мотивація, застосування оптимального середовища програмування, контроль з боку викладача.*

Висновки. *Для успішного навчання студентів спеціальності «Комп'ютерна інженерія» системного програмування необхідне дотримання таких педагогічних умов: достатній рівень знань і вмінь з дисциплін; вибір оптимального середовища програмування; застосування допоміжних засобів навчання; відповідна методика навчання.*

Ключові слова: *системне програмування, обслуговування комп'ютерних систем і мереж, Асемблер, педагогічні умови, навчання, комп'ютерна інженерія*