EDUCATIONAL MODEL OF COMPUTER AS A BASE FOR INFORMATICS LEARNING

Evgeny Eremin

Abstract: It is proposed to use one common model of computer for teaching different parts of the informatics course, connected with both hardware and software subjects. Reasoning of such slant is presented; the most suitable themes of the course, where it is practical, are enumerated. The own author's development (including software support) – the educational model of virtual computer "E97" and compiler from Pascal language for it – are described. It is accented, that the discussed ideas are helpful for any other similar model.

Author Keywords: education, teaching, learning, informatics, computer model, compiler.

ACM Classification Keywords: K.3.2 Computers and education: computer and information science education; I.6.3 Simulation and modeling: applications

Introduction

One of essential and principled difficulties in studying informatics course is the complex character of the contents of this young and fast-developing science. In particular, a computer itself, being in point of fact the indissoluble unity of various technologies, for simplicity of study is divided into software and hardware (see the review of computer subjects in [1]). The distance between these two groups of disciplines shows a tendency to grow: numerous program layers (ROM BIOS, operating system, high-level languages and visual systems, application software) increasingly separate people, who work on computer, from hardware. Kaleidoscopic interchange of hardware models and software versions complicates the selection of material to study yet more.

As a result, in the conventional courses we see user interface and elements of the high-level language on the one hand, and computer machinery with its binary, absolutely "invisible" for user in practice, on the other. Qualified lecturer certainly demonstrates and accents their linkage, but not all students thoroughly recognize this actual unity. Unfortunately, the mentioned above separation increases every year, and it makes more difficult to form adequate students' outlook.

One of the possible ways of logical join of all courses, connected with software and hardware, is to study them on some common base. As real computers has complex, different in details and fast-changing organization, the deepest courses often replace computer hardware with a simpler educational model [2-5]. Such models are demonstrative and easy to understand on the one hand, and retain all most important and invariable features of real machines on the other one. Apparently the most particular model, called MIX, was created by the famous mathematic Donald Knuth as some abstract base for studying the fundamental principles of programming and computer calculations [2]. It's worth attention that recently the author of this classical multivolume book has upgraded his model [6, 7] to make it much more actual.

The aim of this paper is to demonstrate that the educational model of computer may be used in informatics wider and practically become a connecting-link between learning of software and hardware. Although the author used his own educational model "E97" [8, 9] first, all developed ideas can be fully realized on any other similar model.

Possible Usage of Educational Models in Informatics Learning

The most evident part of the course, where teaching may be based on an educational model, is computer organization. As every well-constructed model certainly reflects the most essential features of its original, this subject needs no proof.

Another suitable theme is number notations, where we can explore the theoretical algorithms of converting from, for example, decimal system into binary or vice versa, on our model base. The technique of model's usage depend on pedagogical aims: students may develop the program for converting themselves, analyze a code

which was prepared for them, test a ready program especially for the most interesting cases (signed or unsigned data, negative numbers, large values and overflow) and so on. Consequently we obtain a possibility to learn this matter in practice instead of dull abstract exercises on the paper sheet.

Expanding these ideas, we may offer to learn non-numeric information coding (text, graphics) with the help of an educational computer model too. Several other fundamental problems – byte organization of memory, storing of multi-byte values (big or little endian mode), ASCII or Unicode character sets – will necessarily be discussed and practically assayed on the side.

The detailed study of logical operations such as AND, OR, XOR, NOT on the base of the model is also an interesting application. This is particularly important because except high-level logical operations with Boolean data (widely used in all conditions and queries), computer instruction set always contains bitwise commands with the same names.

At last the same educational computer model can serve as a base of software study. Foremost we can mention fundamental ideas of compiling (will be demonstrated below), text proceeding, data compressing and other software foundations. The visual analysis of several samples on the computer model (program of changing letters' case, converting number to string and vice versa, compressing some sequences of identical codes, advantage of variable-length coding) often improve students' knowing much better than several abstract lectures or solving a lot of problems with the help of application software.

If an educational model of computer is developed enough, we may even try to demonstrate logic of the parallel computations or multitasking.

So the main novelty in the described slant is not the idea to use computer models in hardware learning (they are traditionally applied in this field), but the possibility to study other subjects of the informatics course on the common base of one educational model. In particular such method of teaching for software subjects is not practiced yet.

"E97" as an Example of the Base Model

It was already accented above that the concrete choice of the model for learning is not too essential. Nevertheless an educational model must closely accord to the contents of informatics course, be simple and demonstrative, but on the same time contain the most representative features of the modern computers. Hence not every model of computer is suitable as a common base of the course.

Making no pretence to the unicity of solution, in 1997 the author developed an educational model called "E97" [8]. Its description was later published in the books with greater edition [4, 9]. The model is used in the teaching of different themes of the informatics course in several educational institutions in Russia.

The following features, inset into "E97", differentiate it from other ones:

- adequacy to real principles of organization of personal computers (byte memory structure, information exchange via input/output ports and so on);
- very simple, but full-range instruction set;
- possibility to process non-numeric data with different memory dimension;
- actual memory addressing, including indirect method by means of processor's registers;
- wide use of the subroutine library that simulate ROM-BIOS; these subroutines can additionally demonstrate the samples of programming;
- several levels of learning from introductory to full-scale processor language programming.

The architecture of the virtual computer "E97" is similar to the legendary PDP computer family [10], known by its exact consistency and clearness. It consists from processor unit, two kinds of memory – RAM and ROM, and also simulates hardware exchange with keyboard and display. "E97" model has a capability to process numeric information (two bytes) as well as text (one byte) data.

Thoroughly developed ROM is another important feature of the model. Its existence essentially facilitates student's work with external devices, actually reducing it to the standard subroutine call. Such technique really takes place in modern computers: in IBM PC, for example, this kind of memory is called ROM-BIOS [11].

From the educational point of view it's important to accent, that ROM contents is stored in the text file with detailed comments, so such presentation may be used as a learning subject.

Being a multilevel model, "E97" allows creating the individual tasks of different complexity; hence this educational model may be useful in different kinds of educational institutions.

"E97" has several software realizations, including MS-DOS and Windows versions. The most universal realization was written on JAVA language, so it doesn't depend on hardware and able to run on any computer. The last mentioned version possesses one more advantage: its JAVA applet can be inserted into any Web-page with educational materials.

Demonstrational Compiler, Based on the Educational Model

The educational model of computer can be used as a base for the familiarity with the software principles. The importance of such slant springs out in many respects from the way of familization with computers at present. The question is that the computer specialists with long experience perfected their knowledge parallel to extension of calculating machinery. So they have naturally passed the way from processor codes to modern high-level languages and have seen full pallet of programming methods. Now most of people begin to study straight from high-level languages (or even fully ignore them, so the computer logic becomes mysterious for them). As a result, this missing of several technologies makes many concepts, such as methods of passing parameters or constructing of economical data structures, unappreciated.

To compensate the described above limitations, the author offers his specialized educational software, which demonstrates the major principles of automatic program generation. The professional systems are evidently unsuitable for mentioned above educational purposes, because they are absolutely closed and generate the code that is hard for human analyses.

The educational demonstrational compiler "ComPas" operates with some limited subset of Pascal language. This subset includes all algorithmical structures: assignment, conditional operator IF and three traditional types of cycles – WHILE, REPEAT and FOR. Standard input/output procedures READ and WRITE are realized as the call of subroutines from the special library; these subroutines in turn make the necessary preparations and redirect the call to ROM. The compiler supports standard data types and arrays of them.

The enumerated above possibilities allow to demonstrate our students the following essential features of high-level languages:

- variables, constants, typed constants and difference between them;
- different data types and organization of their storage in RAM (including arrays and access to their elements);
- conversion of values from one type into another (CHAR into INTEGER or so on);
- methods of the main algorithmical structures' realization;
- details of procedures usage

and other fundamental concepts and principles.

To illustrate how demonstration compiler works, let's consider the simplest Pascal program, presented below:

```
PROGRAM sample;
CONST x = 2;
VAR y: INTEGER;
BEGIN y := x + 10;
      WRITELN(y)
END.
```

As a result of translation, "ComPas" generates short and transparent code and shows it on the screen with detailed comments in the form of a table. The example of such table, shown below, is filled for Intel processor's codes (although the program for educational model looks easier, nevertheless its description requires more additional information, so we'll not discuss it in this paper).

Address	Code	Assembler	Actions	Comments
100	E97D01	jmp 0280		jump to the beginning
103	…			library with standard subroutines
280	B80200	mov ax,0002	2 ==> ax	constant *x*
283	B90A00	mov cx,000A	10 ==> cx	constant 10
286	01C8	add ax,cx	ax + cx ==> ax	*x* + 10
288	A3FE04	mov [04FE],ax	ax ==> [4FE]	save result into *y*
28B	A1FE04	mov ax,[04FE]	[4FE] ==> ax	load *y* value
28E	E8C6FE	call 0157	print integer	WRITE *y* (call subroutine from the library)
291	E8FDFE	call 0191	next line	LN (call subroutine from the library)
294	CD20	INT 20	return to system	END.

The analysis of the above program comes easy and cogitable even for beginners. Students can simply find every Pascal operator and carefully examine it (in the above table separate operators are marked by gray color; software has special navigation controls for this purpose).

The educational compiler is freely spread via the Internet and everybody can download it from Web-page [12]. You also may find links for detailed on-line documentation there.

Conclusion

Thereby we see that the idea of the usage of common educational model of computer as a base for learning is suitable for different themes of the informatics course. The discussed slant, as pedagogical experience attests, gives a possibility to refine students' knowledge and forms in their mind a more adequate picture of data processing by means of modern computer machinery.

Bibliography

[1] Computing Curricula 2004 (draft). URL: http://www.acm.org/education/curricula.html

[2] Donald E. Knuth. The Art of Computer Programming. Reading, Massachusetts: Addison-Wesley, 1997

[3] J. Glenn Brookshear. Computer Science: an overview. Reading, Massachusetts: Addison-Wesley, 2000

[4] Mogilev A.V., Pak N.I., Henner E.K. Informatics. Moscow: Academy, 1999 (in Russian)

[5] Lapchik M.P., Semakin I.G., Henner E.K. Methodics of informatics teaching. Moscow: Academy, 2001 (in Russian)

[6] Donald E. Knuth. MMIXware: a RISC Computer for the Third Millennium. Heidelberg, Springer-Verlag, 1999

[7] MMIX Homepage. URL: http://www-cs-faculty.stanford.edu/~knuth/mmix.html

[8] Eremin E.A. How modern computer works. Perm: PRIPIT, 1997 (in Russian)

[9] Eremin E.A. Popular lectures about computer organization. St.-Petersburg: BHV-Petersburg, 2003 (in Russian)

[10] Wen C. Lin. Computer Organization and Assembly Language Programming: PDP-11 and Vax-11. New York: John Wiley & Sons, Inc., 1985

[11] Peter Norton. The Peter Norton Programmer's Guide to the IBM PC. Microsoft Press, 1985

[12] ComPas Hompage. URL: http://www.pspu.ru/personal/eremin/eng/myzdsoft/compas.html

Author's Information

Evgeny Aleksandrovich Eremin – Perm State Pedagogical University. Russia, 614990, Perm, Sibirskaya str., 24. e-mail: eremin@pspu.ac.ru