Simulation Model of the Computer's Units and Peripherial Equipment Interaction

Natalia Efimushkina
Samara State Technical University
Samara, Russia
nvefimushkina@mail.ru

Sergey Orlov
Samara State Technical University
Samara, Russia
orlovsp1946@gmail.com

Nikolai Khrisanov
Samara State Technical University
Samara, Russia
samhnn@mail.ru

***Abstract*—The paper deals with the principles of constructing a simulation model for studying the basic processes and devices in the computer I / O subsystem. The procedure for selecting the composition of the studied characteristics of input-output devices, as well as the structure and level of detail of the parameters of the objects under study is considered. Typical computer hardware with keyboard and monitor is offered as a peripheral device. When modeling, probabilistic characteristics of data processing in the input-output subsystem are set. The modeling program is developed in Java in NetBeans 8.1. A user interface based on the animation of the processes in I / O subsystem is described. This provides obtaining a visual representation of interrupts, generation of control signals, and transfer of codes from peripheral devices to the memory and the central processor. The user interface provides the setting of the necessary modes and simulation parameters. The simulation results are presented in the form of time charts of devices and the numerical values of these characteristics. The input / output system under study can operate in one of three modes: only with information input, with data input and output to a monitor, and also only with data processing and output to a monitor. The paper presents the study results of the simulated system in various modes, confirming the adequacy of the developed model.**

Keywords—simulation, personal computer, processor, input-output subsystem, peripheral

I. Introduction

The modern personal computer is a complex technical system containing multiple devices and subsystems where several processes generated by the user programs can occur simultaneously. The study of such processes is the subject of the theory of computing systems with three main classes of methods: analytical, imitational and experimental [1 - 3]. Experiments performed directly with the object provide the most reliable results, however, the high complexity and cost of equipment limits the use of these methods for training. In this case, the simulation methods are most promising [4 - 6].

Modern federal educational standards allow the replacement of complex laboratory equipment with its virtual counterparts. The proposed simulation model can be assigned to this class. It is designed to train students, but can also be used for research purposes.

The method of simulation software models of this type has been developed by the authors over the years. In [7 - 10], the simulation software models were presented for the study of various computer subsystems.

II. Problems of Simulation Model Development

When developing the model, the following tasks were solved:

- goal of the study and the choice of the characteristics obtained by the model;
- selection of the device composition that must be displayed;
- levels of detail in the structure and processes in the object;
- evaluation of the model adequacy.

The most important characteristics of computing systems are their performance and related timing data. Therefore, when developing the models, the authors evaluated the busy and idle time slots of the computer and its central devices when solving some typical tasks. This approach has defined the features of solving other tasks.

The general computer structure is shown in Fig.1. The model includes only devices that have a significant impact on its operation and determine its time characteristics. However, minor elements complicating the process of research and perception were not included. Typical equipment of the computer was selected as peripheral devices: a keyboard and a monitor. The model contains the following elements that are a typical set of the computational process: a processor; random access memory; a keyboard; a monitor; a video card; buses and a chipset.

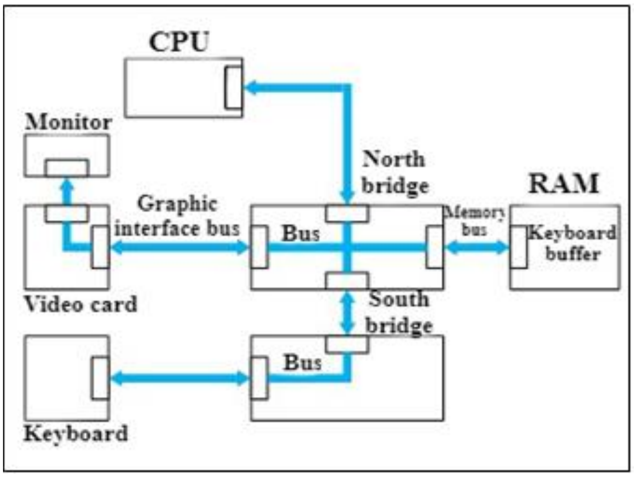


Fig. 1. The general structure of the personal computer

The simulation model contains the minimum number of elements that affect the operation of I / O subsystem. As for the level of detail of parameters, the model simulates devices without representing their internal structure. The computational process generated by a program that runs on computer devices is represented at the level of processor commands. Among these commands, keyboard and screen

978-1-7281-6700-8/19/$31.00 ©2019 IEEE

input operations can be performed. The number of I / O commands is set by the user, and the position in the total mix of commands is randomly determined. This allows one exploring a wide variety of computing processes.

III. Description of the Model of Interaction between Computer Units and Peripherals

The proposed model is a program for simulation of interaction of input and output devices and the central part of the modern personal computer [11]. It is designed for laboratory work on discipline "Computer Hardware" for the bachelor's program "Informatics and Computer Science" and "Software Engineering". The model can be useful in the study of real computers that have a similar architecture. It allows you to study the features of computational processes in the input-output subsystem. It is possible to investigate the influence of a variety of factors on the performance of subsystem devices. The program provides an estimate of the time characteristics. It also has a developed interface and uses animation.

In this case, system and application programs are in RAM during computer operation. The executable program and its data are placed in the processor's cache. I / O operations in modern computers are carried out with the active participation of the central processor. It initiates the keyboard input, generates a ASCII-code character and puts it in an especially reserved area of RAM. The connection between the processor, I / O devices and main memory is provided by the system bus and the chipset consisting of controllers: North Bridge and South Bridge.

An application requiring keyboard input implements a cyclical sequence of processor instructions and I / O operations. After the execution of a group of commands in the processor, the interrupt associated with the operation of entering data from the keyboard occurs. This operation is carried out with the participation of the user, and the processor controls the operation of all devices at this time. At the end of the input, command processing continues until the next interrupt, and so on.

Entering information into the computer from the keyboard can be done in one of two modes:

- with screen output;
- without screen output.

There are tasks that do not require input from the keyboard. They perform only processor processing and display on the monitor. Only the processor, random access memory and input-output subsystem are used in it.

The modeling program was developed in the Java in the NetBeans 8.1 environment. The initial data for the simulation are:

- CPU speed (million op / s);
- executed processor instruction quantities;
- number of I / O operations in the simulated program;
- number of input / output characters in one operation;
- mode of operation of the simulated program (only with input, with input and output of characters, and also without input operations).

An example of the screen form of the modeling process in this mode is shown in Fig. 2. In the mode without input, the keyboard is not used in the simulation process, and the characters that need to be processed by the processor are taken from the cache. In the mode without displaying characters, the graphics subsystem (video card and monitor) is also not involved in the modeling process.

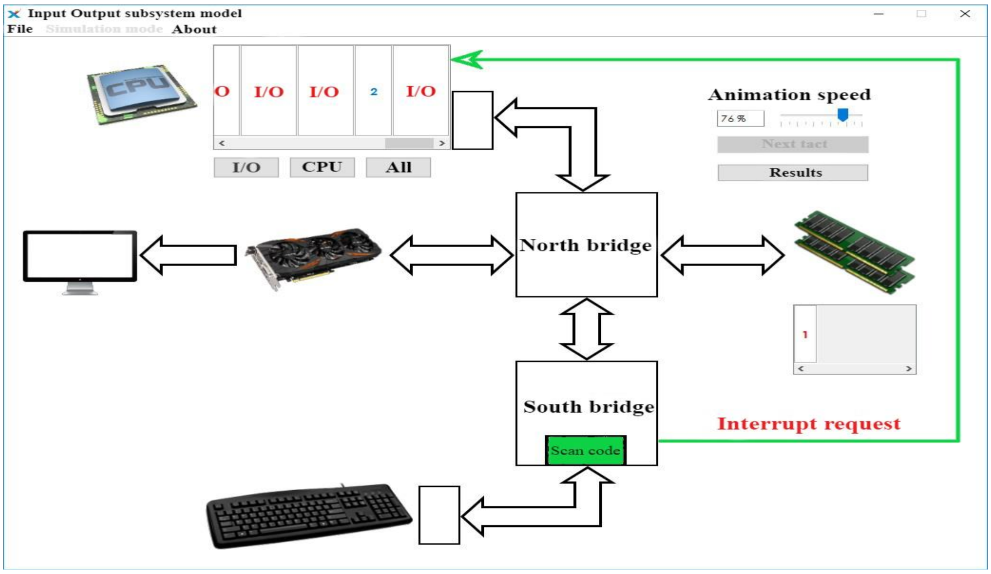


Fig. 2. The screen form of the modeling process in the input-output susystem of a persanal computer

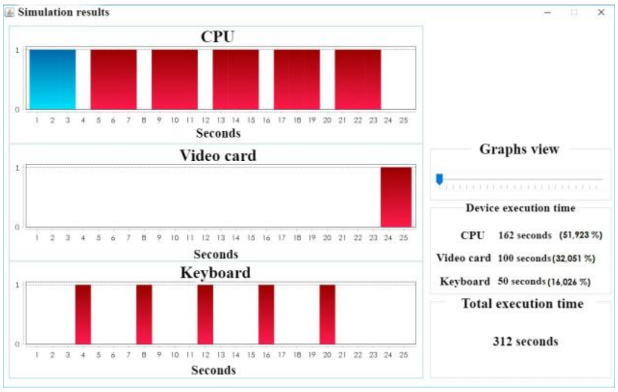
The model can work in one of two modes: “Continuous” or “Cycle by cycle”. The cycle mode allows you to track all stages of the execution of input-output commands. A device participating in the next clock cycle is highlighted on the user's screen.

The simulation ends after the execution of a specified number of instructions in the simulation program. Its results are presented in the form of time diagrams of the average occupancy rate of the main devices and numerical values of the times of their operation. An example of the screen form of the simulation results is shown in Fig. 3.

The simulation model was used to study the typical I / O subsystem of a modern computer. The study results are shown in Fig. 4 – Fig. 6.

The monitor Sysinfo measured the characteristics of the computing processes in the system consisting of the central processor, RAM, keyboard, video card and monitor. The measurement results show that the modeling errors do not exceed 20%, which is quite acceptable for training purposes.

The second model is designed to study the external memory subsystem operating in the direct memory access mode (DMA) in the random access memory [12].It is known that with such access the central processor does not participate in the exchange, but only initiates and terminates it. The model allows you to study the features of the functioning of the subsystem DMA in single-and multi-program mode. The initial data for the simulation are the parameters of the subsystem's devices and the tasks (programs) that it serves.



Device execution time

CPU 162 seconds (51,923 %)

Video card 100 seconds(32,051 %)

Keyboard 50 seconds(16,026 %)

Total execution time

312 seconds

Fig. 3. The screen form of the simulation results

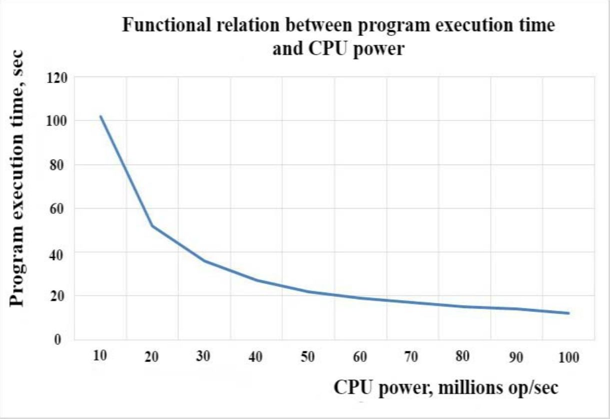


Fig. 4. Chart of the program exucation time

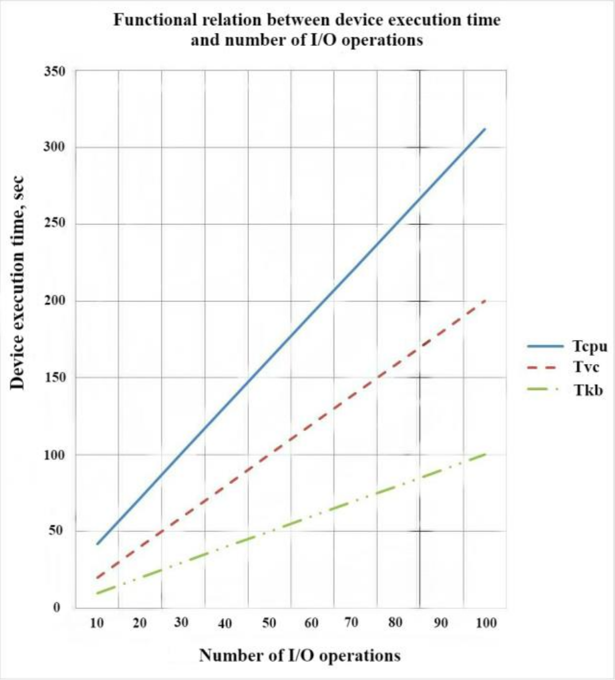


Fig. 5. Chart of the device execution time

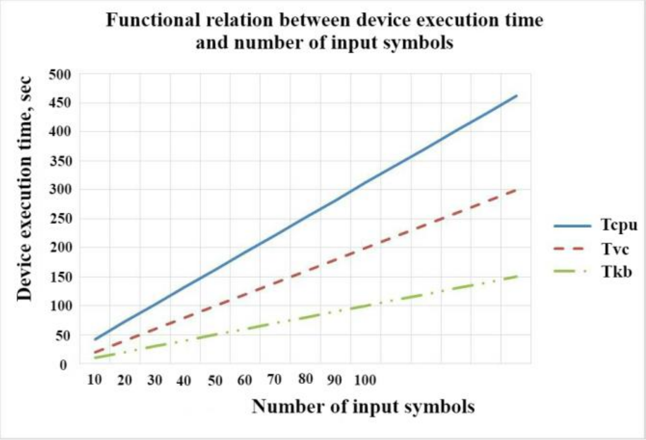


Fig. 6. Chart of the functional relation between device exucation time and number of input symbols

Processor speed, bus and bridge capacities, as well as RAM and external memory capacity were specified for devices. Task parameters are: the total number of commands and the number of file accesses on the HDD, as well as the type of operation (read or write).In addition, file parameters such as the length and length of a single record, as well as the number of the HDD, where it is stored, are set for them. In the single-program mode, the system runs one task, and in multitasking - up to three tasks. The simulation results are the time characteristics of service tasks: the total execution time of a given set of programs, as well as graphics and device processing ratio. The simulation results for the multiprogramming mode are presented in Fig. 7.

A simulated program is represented as a sequence of commands in which file access operations are included in random places. As a result, it is divided into blocks of processor operations, the boundaries of which are specified by the file access commands.

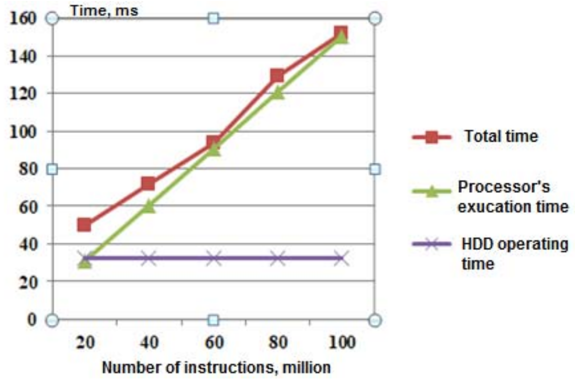


Fig. 7. The processing time of devices in the mode of direct memory access when servicing three programs

The execution of the program is reduced to the cyclic repetition of the stages of the processor processing, the search for the initial mark of the file on the HDD and the direct exchange of information between the operational and external memory (DMA).These processing steps are displayed in the model in the appropriate color.

IV. CONCLUSION

The paper describes models of interaction between central devices of a personal computer and a typical set of peripheral devices: keyboards, monitors and hard drive memory. The simulation model has a convenient user interface, makes extensive use of animation and provides an opportunity to study in detail the processes of data input and output. This provides a study of a wide range of processes generated by user programs, and to identify the most effective modes of operation of computers. The model can serve both for training students and undergraduates in areas related to computer science and computer technology, as well as for research purposes.

REFERENCES

[1] S. A. Orlov and B. Ya. Tsilker, Organizatsiya romp'uterov i sistem [Computer and systems organization]. St. Petersburg, Piter Publ., 2011. (in Russuan)

[2] G. R. Ganger and Y. N. Patt, "Using System-Level Models to Evaluate I/O Subsystem Designs", IEEE Transactions on Computers, June 1998, V.47. pp. 667-678.

[3] A.V. Petrov, Modelirovaniye processov i sistem [Modeling of Process and Systems].St. Petersburg, Lany Publ., 2015. 288 p. (in Russuan)

[4] J. Tsai and A. Agarwal, "Analyzing multiprocessor cache behavior through data reference modeling". Proceedings of the 1993 ACM SIGMETRICS conference on Measurement and modeling of computer systems, New York, NY, USA, 1993. ACM. pp. 236-247.

[5] J. M. Paul, A. Bobrek, et al. "Schedulers as model-based design elements in programmable heterogeneous multiprocessors", Proceedings of the 40th conference on Design automation, New York, NY, USA, 2003. ACM. pp. 408-411.

[6] I. Gluhovsky and B. O'Krafka, "Comprehensive multiprocessor cache miss rate generation using multivariate models", ACM Trans. Comput. Syst., 23(2):111–145, 2005.

[7] S.P. Orlov and N. V. Efimushkina, Organizatsiya vychislitel'nykh mashin i sistem [Organization of computer and systems]. Samara: SamGTU Publ., 2016. 280 p. (in Russuan)

[8] N.V. Efimushkina and S.P. Orlov, "Simulation models for studying the multiprocessor systems", Proceedings of Computer Modeling and Simulation, COMOD '2014. SPbGTU. St. Peterburg. 2014. pp. 145 - 149.

[9] S.P. Orlov and N.V. Efimushkina, "Simulation models for parallel computing structures", Proceedings of XIX IEEE International Conference on Soft Computing and Measurements ,SCM'2016. 2016. v.1. pp. 231-234.

[10] S.P. Orlov and N.V. Efimushkina, "Simulation models of computer subsystems", Proceedings of XX IEEE International Conference on Soft Computing and Measurements, SCM '2017. 2017. pp. 323 – 326.

[11] B. Gavish and U. Sumita. "Analysis of channel and disk subsystems in computer systems", Queueing Systems. Theory and Applications, 1988, V. 3, Issue 1, pp 1–23.

[12] J. J. Pieper, A. Mellan, et al. "High level cache simulation for heterogeneous multiprocessors", Proceedings of the 41st annual conference on Design automation, New York, NY, USA, 2004. ACM. pp.287-292.