
REVISTA
INCLUSIONES
SEMINARIO BRASIL
AS REPRESENTAÇÕES SOCIAIS NO CONTEXTO DO BRASIL
Revista de Humanidades y Ciencias Sociales
Número Especial Octubre / Diciembre
2019
ISSN 0719-4706

Editorial Cuadernos de Sofía
Santiago – Chile
Representante Legal
Juan Guillermo Estay Sepúlveda Editorial

Indización, Repositorios y Bases de Datos Académicas

Revista Inclusiones, se encuentra indizada en:

CATÁLOGO

BIBLIOTECA UNIVERSIDAD DE CONCEPCIÓN

REVISTA INCLUSIONES
REVISTA DE HUMANIDADES
Y CIENCIAS SOCIALES

CUADERNOS DE SOFÍA
EDITORIAL

ISSN 0719-4706 - Volumen 6 / Número Especial / Octubre – Diciembre 2019 pp. 570-580

COMPLEX OF PROGRAMS FOR SIMULATING MODELS OF TASKS OF RECONFIGURABLE COMPUTER SYSTEMS USING PRIORITY MASS SERVICE NETWORKS

Dr. (C) Alexey I. Martyshkin
Penza State Technological University, Russia
alexey314@yandex.ru

Fecha de Recepción: 30 de mayo de 2019 – **Fecha Revisión**: 11 de junio de 2019
Fecha de Aceptación: 27 de agosto de 2019 – **Fecha de Publicación**: 25 de septiembre 2019

Abstract

The paper provides refined mathematical models of task managers for reconfigurable and distributed computing systems. The developed models are based on queuing networks with FIFO and priority disciplines. The description of the created software package for the simulation of the specified classes of dispatchers as part of a reconfigurable system with the specified parameters is presented. The proposed software package has the ability to conduct simulation and obtain the characteristics of the entire reconfigurable system, as well as its individual nodes. The main difference between the developed software package and its analogues is the ability to simulate a queuing network which includes systems with a queue length limitation before the serving device and priority queuing systems, which allows a wider range of data of interest to obtain. The complex is easy to use and does not require knowledge of the modelling language. A set of programs can be used by developers of operating systems for their research and debugging at the development stage.

Keywords

Task manager – Service discipline – Reconfigurable computing system – Software package

Para Citar este Artículo:

Martyshkin, Alexey I. Complex of programs for simulating models of tasks of reconfigurable computer systems using priority mass service networks. Revista Inclusiones Vol: 6 num Especial (2019): 570-580.

Introduction

When developing and improving computation systems, it is necessary to conduct studies of their characteristics. Building a real device requires a lot of time and money, which is critical when the structure of the computation system is changed many times[1]. Therefore, to study the characteristics of the reconfigurable computation system, software models are used that is not a physical device but are described by data and algorithms in computer memory. Therefore, their construction and modification are carried out without financial costs.

Modelling of the computation system is an important part of the system design phase. Since, according to the obtained characteristics of various models, it is possible to compare various computation systems, analyse their performance under a changing load, etc. In addition, the cognitive value of modelling is important[2]. The paper proposes a simulation modelling software system of a reconfigurable computation system, which is characterised by a convenient visual user interface simplifying the procedure for setting the parameters of the simulation object and obtain a larger number of experimental data per unit time[3].

Purpose of the work is the creation of a software package for the simulation modelling of computation systems presented in the form of queuing networks and collecting statistics on the nodes of the mathematical model and the entire network as a whole. Using the created software package, simulation modelling of the process and thread management functions of the kernel of reconfigurable and distributed operating systems will be carried out, in particular, the task scheduling and dispatching functions.

Formulation of the problem

In the works[4], analytical mathematical models of task managers with time-sharing and space separation strategies for reconfigurable, multiprocessor, and distributed operating systems were considered, a description of the developed software package for the analytical calculation of characteristics of these task manager classes within the computation systems according to the given parameters[5] was given. Here we will focus on the implementation of the software package for the task manager simulation modelling using priority queuing network. To evaluate the performance indicators of the task manager, the author developed a mathematical model of *n*-processor reconfigurable

[1] A. I. Martyshkin, “Hardware buffer memory of the multiprocessor system”, ARPN Journal of Engineering and Applied Sciences Vol: 13 Issue 23 (2018): 9151-9156.

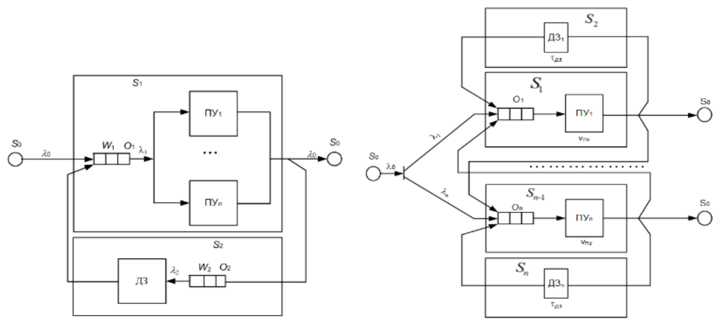
[2] R. A. Biktashev; A. I. Martyshkin and N. G. Vostokov, “A complex of programs for determining the characteristics of task managers in multiprocessor systems using priority stochastic queueing networks”, Fundamental Research num 10 (2013) 13-20.

[3] A. I. Martyshkin; R. A. Biktashev and N. G. Vostokov, “A software package for simulation of task managers for multiprocessor systems using priority queuing networks”, Basic research num 10-11 (2014): 2155-2159.

[4] A. I. Martyshkin and O. N. Yasarevskaya, “Mathematical modelling of Task Managers for Multiprocessor systems on the basis of open-loop queuing networks”, ARPN Journal of Engineering and Applied Sciences Issue 16 (2015): 6744-6749 y A. I. Martyshkin and O. N. Yasarevskaya, “Queries Service Time Research and Estimation during Information Exchange in Multiprocessor Systems with "Uni Bus" Interface and Shared Memory”, National Academy of Managerial Staff of Culture and Arts Herald Issue 3 (2018): 790-797.

[5] R. A. Biktashev; A. I. Martyshkin and N. G. Vostokov, “A complex of programs…

computation system with process scheduling algorithms with a strategy of separation in time (Fig. 1, a) and separation in space (Fig. 1, b). The mathematical model is described in detail[6].



a) б)

Fig. 1

Diagram of the model of n-processor reconfigurable computation system with process scheduling algorithms for the time-sharing strategy (a) and space separation strategy (b)

We write an expression for a reconfigurable computation system with a common task manager, obtained in more detail.[7]

$$W = \frac{w_1}{p_{10}} + \frac{w_2 \cdot p_{12}}{p_{10}} = \frac{w_1 + w_2 \cdot p_{12}}{p_{10}}, \qquad (1)$$

Where w_1 is waiting time in a queue before processor nodes; w_2 is the timeout for the processor node before occupying the task manager; p_{10} is the probability of a request exit from the reconfigurable computation system; p_{12} is the likelihood of a task transition to service in the task manager.

Response time in a reconfigurable computation system with task manager and shared task queue in the reconfigurable computation system

$$U = \frac{w_1 + k \cdot (t_k + \delta) + P_{12} \cdot (w_2 + k \cdot (\tau + \varsigma))}{P_{10}}, \qquad (2)$$

Where k - the number of quanta to complete one request; t_k - the duration of one quantum; δ - the time required to reload the cache; ς - the working time of the task manager, τ - the time required to switch the context.

Queue time before processor node in a reconfigurable computation system with distributed task manager and FIFO service discipline

---

[6] A. I. Martyshkin, "Development and research of open-loop models the subsystem "processor-memory" of multiprocessor systems architectures UMA, NUMA and SUMA", ARPN Journal of Engineering and Applied Sciences. Vol: 11 Issue 23 (2016): 13526-13535 y A. I. Martyshkin and O. N. Yasarevskaya, "Mathematical modelling of Task Managers…

[7] A. I. Martyshkin and O. N. Yasarevskaya, "Mathematical modelling of Task Managers…

$$w_i = \sum_{i=1}^{n} \frac{\lambda_i + \sum_{\forall n(n \neq i)} p_{ji} \cdot \lambda_i}{\lambda_0} \cdot \frac{L_i}{\lambda_i}, \quad (3)$$

Where λ_i is the intensity of the input flow at the i-th stage, p_{ji} - the probability of transition from queue j to queue i, L_i is the length of the i-th stage.

Response time in the reconfigurable computation system with the distributed task manager and FIFO service discipline

$$U = \sum_{i=1}^{n} \frac{\lambda_i + \sum_{\forall n(n \neq i)} p_{ji} \cdot \lambda_i}{\lambda_0} \cdot \left(w_i + k \cdot \left(t_k + \delta + \tau + \varsigma \right) \right), \quad (4)$$

Where w_i is the waiting time in the i-th queue.

Average queue time for priority tasks in network queues in a reconfigurable computation system with distributed task manager and relative priorities.

$$W = \sum_{i=1}^{n-1} \alpha_i \cdot \sum_{k=1}^{H} w_k^{ОП},\ n = 1, 3, ..., n-1, \quad (5)$$

Where α_i is a transition ratio into the i-th queue, $w_k^{ОП}$ is a waiting time for tasks with relative priorities in the *i*- th queue.

The response time of the reconfigurable computation system with distributed task manager and relative priorities will be.

$$U^{ОП} = \sum_{i=1}^{n} \alpha_i \cdot u_i \quad (6)$$

Where $u_i = W_i^{ОП} + \tau_{обслi}$, $W_i^{ОП} = \sum_{i=1}^{k} w_i^{ОП}$, $\tau_{обслi} = t_{ДЗ} + \nu_{ПУi}$.

Average wait time for priority tasks in network queues in a reconfigurable computation system with distributed task manager and absolute priorities

$$W = \sum_{i=1}^{n-1} \alpha_i \cdot \sum_{k=1}^{H} w_k^{АП},\ n = 1, 3, ..., n-1, \quad (7)$$

Where $w_k^{АП}$ is average waiting time for tasks with absolute priorities in the *i*- th queue.

The response time in the reconfigurable computation system with distributed task manager and absolute priorities will be.

$$U^{АП} = \sum_{i=1}^{n} \alpha_i \cdot u_i \qquad (8)$$

Where $u_i = W_i^{АП} + \tau_{обслі}$, $W_i^{АП} = \sum_{i=1}^{k} w_i^{АП}$, $\tau_{обслі} = t_{ДЗ} + \nu_{ПУі}$.

Implementation of the software package.

The paper proposes a software package for visual simulation modelling of the computation system, in particular, the reconfigurable computation system implemented in Delphi 7[8].

The software package includes a mathematical model of the request source, a mathematical model of the queue and a mathematical model of the service device.

The mathematical model of the request source is used to form the transaction flow in the simulated network according to the following distribution laws for the receipt times (Fig. 2)[9]:

B. Exponential. $f(x) = \frac{1}{\mu} e^{-x/\mu}$ where μ is the expectation E [X].

C. Normal. $f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{-\mu}{\sigma}\right)^2\right]$, where σ is the standard deviation; μ is the mathematical expectation of E [X].

D. Poisson. $f(x) = \frac{e^{-\mu}\mu^x}{x!}$.

E. Uniform. $f(x) = \begin{cases} \frac{1}{b-a}, & x \in [\mathrm{a}, \mathrm{b}] \\ 0, & x \notin [\mathrm{a}, \mathrm{b}] \end{cases}$.

---

8 A. Ya. Arkhangelsky, Programming in Delphi 7. Binom Publishing House. 2003.

9 T. I. Aliev, Fundamentals of discrete system modelling (St. Petersburg: St. Petersburg State University ITMO, 2009) y M. A. Matalytsky; O. M. Tikhonenko and E. V. Koluzaeva, Queuing systems and networks: analysis and requests: Monograph (Grodno: GrSU, 2011).

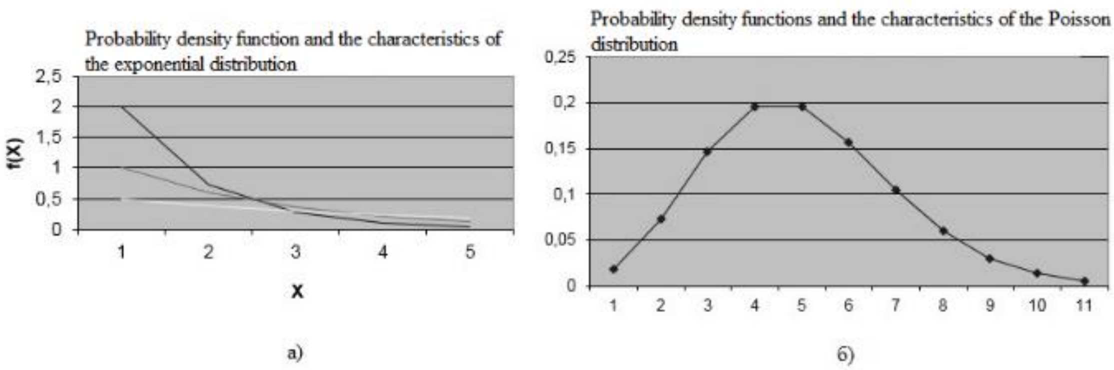


Fig. 2
Graphs of the probability density function and the characteristics of the exponential distribution for μ = 0, 0.5, 1, 1.5, 2 (a) and the probability density functions and the characteristics of the Poisson distribution for μ = 4 (b)

The request source is executed in the form of a random number generator issuing not the requests themselves, but the time intervals in accordance with the specified distribution law in which the request should appear. It includes the following fields: name, the intensity of receipt of requests, variance, type of distribution of request receipt times, and priority level of requests.

The mathematical model of a queue is used to queue the requests for their further processing in the servicing device. It contains the following fields: name, maximum queue length, discipline type of selecting a request from the queue (FIFO and LIFO without priority, FIFO and LIFO with priorities). The queue is presented in the form of a list; a check for its finiteness is implemented.

The mathematical model of the service device is used to simulate the processing or delay of transactions in this device. If the device is free, the task from the previous queue is automatically loaded. If the queue is empty, the device is in standby mode until a new transaction arrives. The mathematical model of the service device operates in accordance with the laws of distribution of the delay time: exponential, normal, Poisson, and uniform. It includes the following fields: name, the average service time of requests, variance, distribution law, and the number of service channels.

The movement of requests according to the created mathematical model can be arranged in the following way: the request movement will be controlled by the procedures of the source, queue, servicing device and absorber. The request is generated in the request source procedure. It is also planned to move it to the next block of the mathematical model. Information about the planned event is placed in the future event list, the time, block number, a pointer to the request, and the type of action are indicated. These entries are sorted by time and priority of the event. Each block through which the request passes, registers this or that data in the future event list, only in the absorber block a request is deleted without planning any action for the future. The modelling process consists of the execution of the first line of the future event list with its subsequent removal. In this case, the model time is assigned the time value in this line of the future event list. When the list is empty, the simulation stops and the calculated data is output.

The developed software package has a visual interface that allows vividly to represent and modify the studied mathematical model during the experiments. Along with this, the mathematical model is saved on the disk as a file which is a full description of the mathematical model. This implies the versatility of usage of the mathematical model, the ability to increase the functionality of the software package without changing the presentation format of the mathematical model, which is calculated continuously until the following conditions are met: generate a certain number of requests or achieve a certain amount of model time. It is possible to calculate systems with failures and without. The result is a generated HTML-file of software package containing the main characteristics obtained in a table form. It is also possible to obtain a detailed description of the entire modelling process in the form of a protocol.

The developed mathematical model is set either by entering the description text in the corresponding field, or using the visual editor when it is presented in the form of a diagram containing the symbols of devices, queues and generators on the computer screen, which also contains possible transitions of requests between elements of the mathematical model. The maximum number of elements in a mathematical model can be selected based on the amount of RAM in the computer, because along with statically allocated memory for the parameters of elements in the modelling process, free memory is dynamically allocated for each request (by default, the maximum number of elements in the mathematical model created is 450).

Before calculating the mathematical model, it is checked for validity. This is necessary because there is the possibility of constructing a mathematical model in which requests will accumulate and stand idle over time. When a request is generated, memory is allocated for it, and when it is deleted, the memory is freed. With an unlimited increase in the number of requests in the reconfigurable computation system, RAM overflow can occur, which leads to a software failure[10].

After the mathematical model is created and checked for validity, its calculation is performed. The request generators are explicitly initialized first, which leads to the appearance of the following events in the initially empty future event list: queuing of the requests generated by the generator procedure and the next request generation in accordance with the parameters of the generators. Further control is provided by the future event list: its first line is executed and is immediately deleted. Modelling process stops as soon as there would be no single line in the future event list.

In the simulation cycle, such procedures as *a generator, queue_in, queue_out, seize, release, terminator*, are called, using the device number, a pointer to the request, etc. These procedures correspond to actions with requests: g*eneration, enqueue, exit the queue, assignment for service, end of service* and *deletion of request*. After completion of the simulation, an HTML text file with a report is generated.

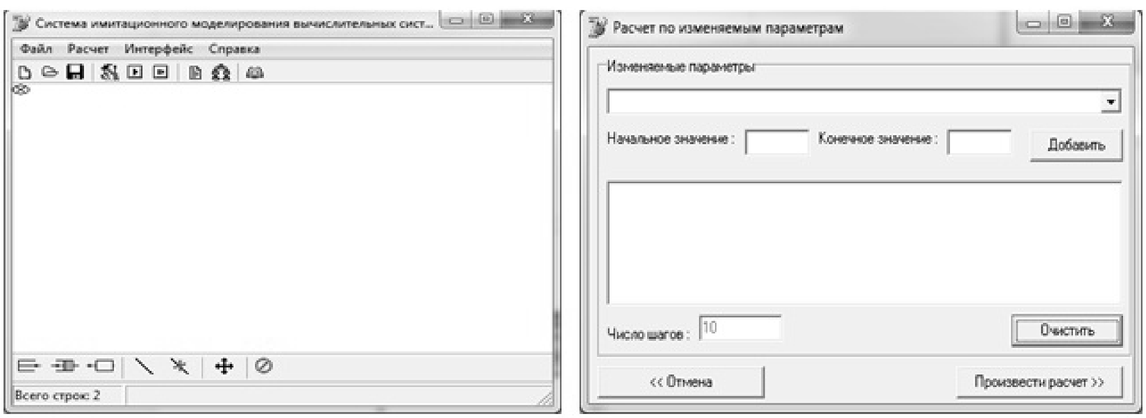
Fig. 3, a, presents a window of the developed software package working in the “Visual editor” mode. In the upper menu, we can select the main operating modes of the software package: in the File menu we create a new mathematical model, load the

---

[10] E. Tanenbaum and H. Bos, Modern Operating Systems. 4th edition (SPb: Peter. 2015) y A. I. Martyshkin, “Mathematical modelling of Tasks Managers with the strategy in space with a homogeneous and heterogeneous input flow and finite queue”, ARPN Journal of Engineering and Applied Sciences Issue 19 (2016): 11325-11332.

mathematical model from disk, save it to disk, and perform an exit; in the Calculation menu we change the simulation parameters, make a calculation, make a calculation for the variable parameters (Fig. 3, b); in the Interface menu we change the representation of the mathematical model (the mathematical model can be represented as a text or as a diagram).

The mathematical model is edited using the toolbar at the bottom of the software package window (Fig. 3, a). The first group of three buttons is used to add an element: a request generator, a queue generator, or a service device. When adding an element, a window is displayed in which we need to enter the basic parameters of the element. The second group of editing tools contains two buttons: the first connects the two elements of the mathematical model, and the second disconnects them. To connect two elements, for example, a queue and a service device, we must first click on the button in the toolbar, and then on the desired queue, then on the device and, after entering the probability of such a transition in the dialogue box, the elements will be connected. To disconnect elements, left-click on the toolbar button, then on the element from which the transition occurs. Next, in the drop-down list, we need to select the transition that we want to delete. The next button allows us to move items around the edit field. The last button of the editing toolbar is needed to delete items.



a) б)

Fig. 3
Window of the developed software package in the "Visual editor" mode (a) and the window "Calculation by variable parameters" (b)

The model is specified either by entering the text describing the model in the appropriate field, or using the visual interface. In the second method of assignment, the model should be presented in the form of a diagram showing the symbols of devices, queues and generators on the computer screen (Fig. 4), which also contains possible transitions of requests between model elements. The calculation results are saved in HTML format and displayed on the screen using the default web browser (for example, Internet Explorer) (Fig. 5). Before the calculation, the model is checked for correctness. This is necessary because it is possible to build without verification a model, in which requests will accumulate over time. When a request is generated, memory is allocated for it, and when it is deleted, the memory is freed. With an unlimited increase in the number of requests in the reconfigurable computation system, RAM overflows; as a result, areas of the system cache are affected, which leads to a software failure.

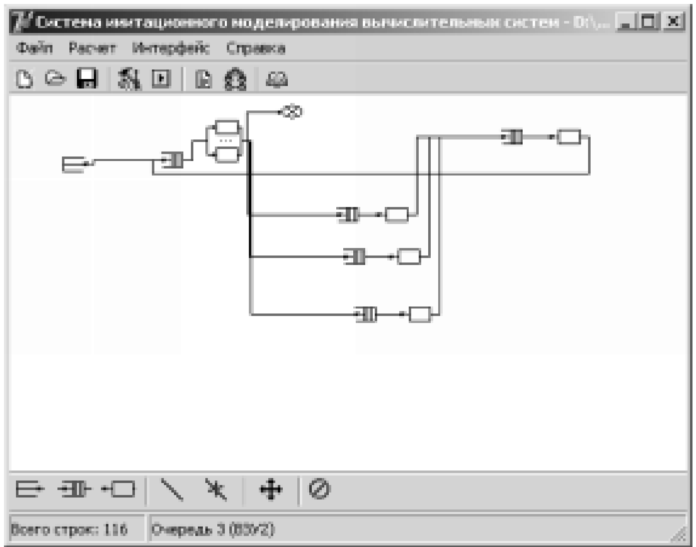


Fig. 4
General view of the program window with the built model

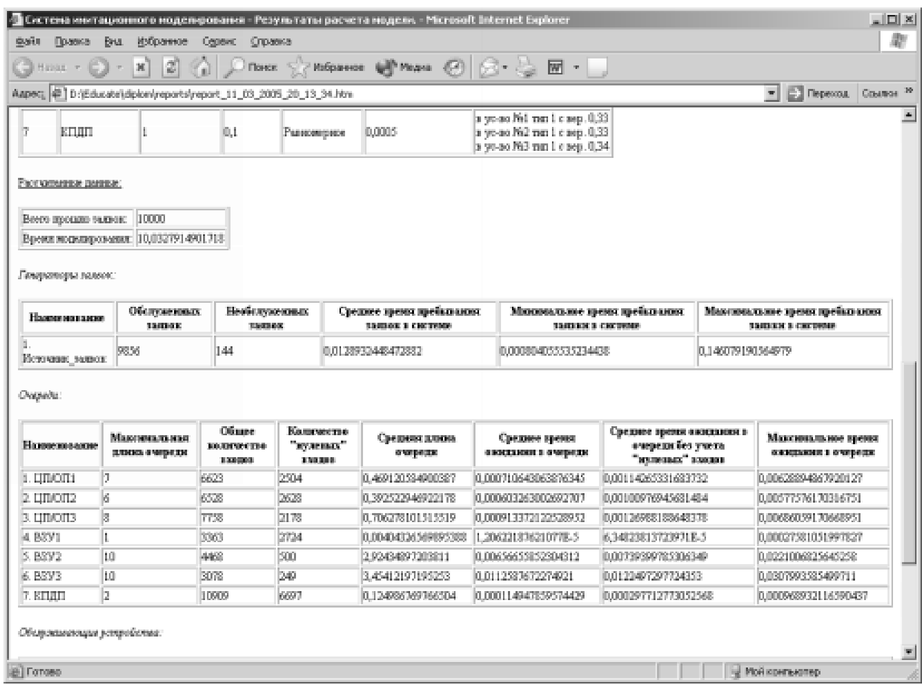


Fig. 5
Presentation of calculation results

Testing and debugging of the software package was carried out using the software package of analytical modelling of a stochastic queuing network[11]. The developed software package is used for simulation modelling of the task manager mentioned above with the strategies of time-sharing and separation in space, having the following characteristics: the

[11] Certificate of state registration of computer programs No. 2013611117. A software package for calculating the probability-time characteristics of stochastic queueing networks.

input task flow intensity λ_0 = 0.03; 0.06; 0.09 tasks / μs (low, medium and high CPU load); task manager context switch time τ = 9 μs, which corresponds to the average values of the context switching time of the operating system in soft real-time[12]; processing time of one CPU task ν = 10 ms. The size of the general queue before the processors is N = 128 tasks; the number of CPUs ranged from 4 to 50.

The results of mathematical modelling showed that a task manager with the specified parameters has a service wait time not exceeding 13 μs. Such a delay corresponds to many existing real-time operating systems, such as LinuxRT[13]. The adequacy of the simulation models for the task manager is confirmed by real data obtained on the prototype system. The error of the obtained mathematical model does not exceed 10%, which is quite acceptable for evaluating the options for implementing a task manager at the system design stage.

Conclusions

The developed software package is designed for modelling a computation system, and attributes of the graphic package; it is easy to use and do not require knowledge of a modelling language. The main difference between the developed software package and the existing ones is the ability to simulate a queuing network with a limited queue length and a priority queuing network. This software package provides a convenient user interface, as well as a visual representation of the simulation results in the form of tables.

Acknowledgements

The paper is published with the support of the scholarship granted by the President of the Russian Federation to young scientists and graduate students for 2018-2020. (SP-68.2018.5).

Bibliography

Aliev, T. I. Fundamentals of discrete system modelling. St. Petersburg: St. Petersburg State University ITMO. 2009.

Arkhangelsky, A. Ya. Programming in Delphi 7. Binom Publishing House. 2003.

Biktashev, R. A.; Martyshkin, A. I. and Vostokov, N. G. “A complex of programs for determining the characteristics of task managers in multiprocessor systems using priority stochastic queueing networks”. Fundamental Research num 10 (2013) 13-20.

Certificate of state registration of computer programs No. 2013611117. A software package for calculating the probability-time characteristics of stochastic queueing networks.

Martyshkin, A. I.; Biktashev, R. A. and Vostokov, N. G. “A software package for simulation of task managers for multiprocessor systems using priority queuing networks”. Basic research num 10-11 (2014): 2155-2159.

[12] V. Mikhalev, “QNX Neutrino Performance Test Results”, Modern automation technologies: Scientific and technical journal num 2 (2012): 82-88.
[13] V. Mikhalev, “QNX Neutrino Performance Test…

Matalytsky, M. A.; Tikhonenko, O. M. and Koluzaeva, E. V. Queuing systems and networks: analysis and requests: Monograph. Grodno: GrSU. 2011.

Mikhalev, V. “QNX Neutrino Performance Test Results”. Modern automation technologies: Scientific and technical journal num 2 (2012): 82-88.

Martyshkin, A. I. “Development and research of open-loop models the subsystem "processor-memory" of multiprocessor systems architectures UMA, NUMA and SUMA”. ARPN Journal of Engineering and Applied Sciences. Vol: 11 Issue 23 (2016): 13526-13535.

Martyshkin, A. I. “Hardware buffer memory of the multiprocessor system”. ARPN Journal of Engineering and Applied Sciences Vol: 13 Issue 23 (2018): 9151-9156.

Martyshkin, A. I. “Mathematical modelling of Tasks Managers with the strategy in space with a homogeneous and heterogeneous input flow and finite queue”. ARPN Journal of Engineering and Applied Sciences Issue 19 (2016): 11325-11332.

Martyshkin, A. I. and Yasarevskaya, O. N. “Mathematical modelling of Task Managers for Multiprocessor systems on the basis of open-loop queuing networks”. ARPN Journal of Engineering and Applied Sciences Issue 16 (2015): 6744-6749.

Martyshkin, A. I. and Yasarevskaya, O. N. “Queries Service Time Research and Estimation during Information Exchange in Multiprocessor Systems with "Uni Bus" Interface and Shared Memory”. National Academy of Managerial Staff of Culture and Arts Herald Issue 3 (2018): 790-797.

Tanenbaum, E. and Bos, H. Modern Operating Systems. 4th edition. SPb: Peter. 2015.

CUADERNOS DE SOFÍA
EDITORIAL