

**МЕТОД ОПИСАНИЯ ПРОЦЕССОВ ВЫПОЛНЕНИЯ ЗАДАНИЙ  
В МУЛЬТИПРОГРАММНЫХ И МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМАХ****В.М. Чёрненький, П.С. Сёмкин**МГТУ им. Н.Э. Баумана, Москва, Российская Федерация  
e-mail: hernen @ bmstu.ru; semkin@bmstu.ru

*Изложена методика моделирования процессов функционирования операционных систем, основанная на концепции потоковых схем. Отсутствие формализованных описаний функционирования операционных систем в терминах теории процессных описаний существенно затрудняет анализ характеристик этих систем методами имитационного моделирования. В качестве языкового средства описания процессов функционирования предложен метаязык, в котором использована идея псевдоязыка описания сцепленных процессов. Для удобства описания указанной предметной области предложен ряд макрооператоров. Применение метаязыка позволило сравнить варианты описания функционирования систем. Наличие описания функционирования системы на псевдоязыке дает возможность без особого труда провести имитационное моделирование на любом алгоритмическом языке имитационного моделирования. В настоящей работе проведено имитационное моделирование по блочно-параметрической схеме с рассредоточенными функциями одного из алгоритмов работы диспетчера операционной системы на языке GPSS. По результатам моделирования построены кривые зависимостей, с помощью которых можно оценить эффективность алгоритмов диспетчеризации в мультипрограммных и мультипроцессорных системах.*

**Ключевые слова:** операционные системы, процессы функционирования, псевдоязык описания, макрооператоры, алгоритмы диспетчеризации, методика моделирования.

**METHOD FOR DESCRIPTION OF TASKS IN MULTIPROGRAMMING  
AND MULTIPROCESSOR SYSTEMS****V.M. Chernen'kii, P.S. Semkin**Bauman Moscow State Technical University, Moscow, Russian Federation  
e-mail: hernen @ bmstu.ru; semkin@bmstu.ru

*A method for simulating the processes of functioning of operating systems is set forth, which is based on the concept of streaming schemes. The lack of formalized descriptions of functioning of operating systems in terms of theory of process description substantially impedes the analysis of characteristics of operating systems using the imitating simulation methods. As a linguistic means for description of processes of functioning, the metalanguage is offered, in which the idea of pseudo-language for description of coupled processes is used. For sake of convenience in describing the indicated knowledge domain, a number of macro operators are offered. The metalanguage application allowed variants of the system-functioning description to be compared. The system-functioning description available in pseudo-language provides the possibility to conduct the simulation without particular difficulty using any algorithmic language of the imitating simulation. In this work, the simulation is performed according to the block-parametrical scheme with dispersed functions of one of algorithms of the operating-system dispatcher in the GPSS language. From*

*the simulation results, curves of dependences are constructed, using which it is possible to estimate the efficiency of dispatching algorithms in multiprogramming and multiprocessor systems.*

**Keywords:** operating systems, processes of functioning, pseudo-language, macro operators, algorithms of dispatching, modeling method.

При изучении процессов функционирования операционных систем и их отдельных компонентов возникает проблема построения адекватного и формализованного описания этих процессов. Такое описание необходимо в учебных целях, для анализа вариантов, построения математических моделей, проведения имитационного моделирования. В работах [1–7] описаны принципы функционирования операционных систем и их компонентов, однако отсутствует формализованное описание процессов функционирования. Последнее существенно затрудняет анализ характеристик этих систем. В связи с этим изложенные в настоящей статье предложения по методам описания взаимосвязанных процессов в операционных системах востребованы в практике моделирования.

Эффективность и производительность вычислительной системы чаще всего оценивают по трем показателям:

- 1) пропускная способность, определяющая число выполненных системой заданий в единицу времени;
- 2) статистически усредненное время выполнения заданий;
- 3) коэффициент использования процессора [6, 7].

Важнейшую роль в формировании перечисленных показателей в мультипрограммных и мультипроцессорных системах играет эффективность алгоритмов планирования и диспетчеризации параллельно выполняющихся процессов. Особое значение эти алгоритмы приобретают при рассмотрении мощных сетевых рабочих станций и серверов. В этих случаях наиболее распространенным способом анализа является имитационное моделирование.

Для целей моделирования предложено использовать методику, основанную на процессных описаниях [8]. Рассмотрим ее применение к исследованию многоуровневого циклического алгоритма планирования и диспетчеризации заданий, являющегося основным в современных операционных системах. Новые задания попадают во входную очередь системы, в которой они ждут освобождения основного ресурса — адресного пространства оперативной памяти. Из входной очереди после получения доли оперативной памяти задания попадают в очередь номер 1 готовых к выполнению процессов. Задания в этой очереди ожидают освобождения ресурса “процессорное время”. Каждому процессу выделяется фиксированный квант времени  $q$ , в конце которого, если задание к этому времени не закончится, оно снимается с процессора и помещается в очередь к процессору, но с номером, на

единицу больше. Задания в  $i$ -й очереди обслуживаются, если пусты все предыдущие очереди.

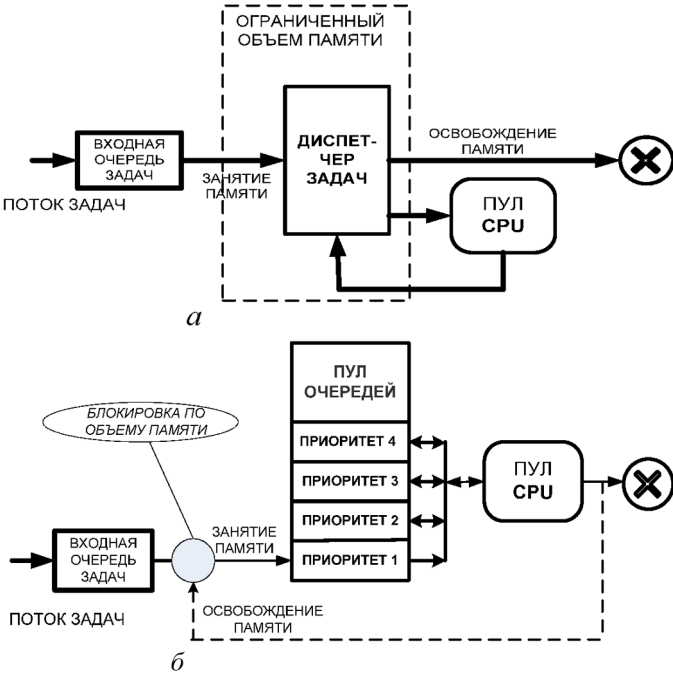
Для анализа эффективности алгоритмов планирования и диспетчеризации используем имитационные модели. В соответствии с методикой проведения имитационного моделирования прежде всего необходимо описать совокупность параллельных взаимосвязанных процессов, отражающих функционирование операционной системы. За основу примем концепцию потоковых схем.

Согласно изложенному выше, можно предложить два варианта функциональной структуры (рис. 1).

В потоковую схему с максимальным числом сосредоточенных функций (рис. 1, а) входит три функциональных блока: 1) блок генератора потока; 2) блок диспетчера задач; 3) блок процессоров. Все функции управления потоком данных сосредоточены в блоке диспетчера. Остальные два блока выполняют простейшие функции: подача очередной задачи в систему и задержка задачи на некоторое время обработки.

В потоковой схеме с рассредоточенными функциями вдоль потока задач (рис. 1, б) каждый блок несет свою логическую нагрузку, а блок диспетчера преобразован в блок управления очередями.

Далее в соответствии с потоковыми схемами предложены блочно-параметрические схемы, отображающие блоки, которые реализуют



**Рис. 1.** Потоковые схемы с сосредоточенными (а) и рассредоточенными (б) функциями

указанные функции и связующие их параметры. Реализация блока предполагает его описание в виде некоторого процесса.

На блочно-параметрической схеме (рис. 2, а), описывающей потоковую схему с сосредоточенными функциями, хорошо видно, каким способом блок диспетчера связывается с другими блоками и какие

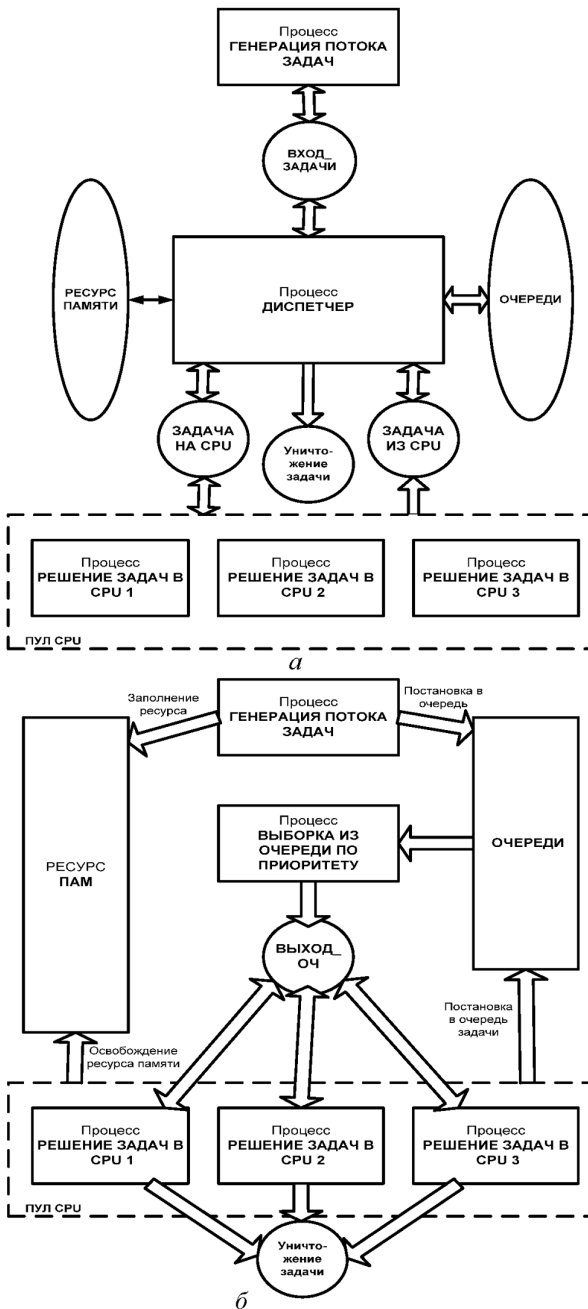


Рис. 2. Блочно-параметрические схемы с сосредоточенными (а) и рассредоточенными (б) функциями

параметрические структуры задействованы в этой схеме. Блок генератора потока выдает поток позадачно, помещая каждую задачу в параметр ВХОД\_ЗАДАЧИ. Аналогично блок CPU взаимодействует с блоком диспетчера с помощью параметров ЗАДАЧА\_НА\_CPU и ЗАДАЧА\_ИЗ\_CPU.

В блочно-параметрической схеме (рис. 2, б), отражающей потоковую схему с рассредоточенными функциями, распределение функций привело к изменению состава параметров схемы.

Для описания процессов в каждом блоке будем опираться на процессную концепцию описания функционирования систем, изложенную в работе [8]. Однако скорректируем метаязык изменением в нем некоторых операторов и служебных слов с учетом рассматриваемой предметной области [9]. Понятие “агрегат” по-прежнему соответствует концепции и означает блок, в котором развивается один замкнутый процесс [8]. Поскольку в предлагаемых ниже описаниях использованы лишь агрегаты, понятие “инициатор” опущено. Веденные изменения не меняют концепцию в целом и хорошо воспринимаются при чтении программы. Программа на псевдоязыке, реализующая блочно-параметрическую схему с сосредоточенными функциями приведена ниже:

#### **ОБЩИЕ ОПИСАНИЯ**

**тип W:** вектор (ОБЪЕМ, НОМЕР\_ОЧЕРЕДИ, ЧИСЛО\_КВАНТОВ);

**ресурс** — ПАМ(Q);

**тип : массив очередей** ОЧЕРЕДЬ (1 – N);

КВАНТ = K;

перед моделированием определить значения Q, K, N

процесс-агрегат ГЕНЕРАЦИЯ ПОТОКА ЗАДАЧ;

НАЧАЛО: **создать** ЗАДАЧА **типа** W;

определить параметры вектора ЗАДАЧА;

ВХОД\_ЗАДАЧА := ЗАДАЧА;

**ждать условие** ВХОД\_ЗАДАЧА = 0;

**перейти на метку** НАЧАЛО;

конец описания процесса-агрегата ГЕНЕРАЦИЯ ПОТОКА ЗАДАЧ;

процесс-агрегат ДИСПЕТЧЕР;

НАЧАЛО: **ждать условие 1** (ресурс ПАМ достаточен для объема ВХОД\_ЗАДАЧА(ОБЪЕМ)) **с переходом на метку** ЗАППАМ,

**условие2** (ЗАДАЧА\_ИЗ\_CPU ≠ 0) **с переходом на метку** ОБРАБ;

ЗАППАМ: **занять ресурс** ПАМ **объемом** ВХОД\_ЗАДАЧА (ОБЪЕМ):

ВХОД\_ЗАДАЧА = 0;

**перейти на метку** НАЧАЛО;

ОБРАБ: ЗАДАЧА\_ИЗ\_CPU (ЧИСЛО\_КВАНТОВ) = ЗАДАЧА\_ИЗ\_CPU (ЧИСЛО\_КВАНТОВ) – 1;

ЗАДАЧА\_ИЗ\_CPU (НОМЕР\_ОЧЕРЕДИ) := n = ЗАДАЧА\_ИЗ\_CPU (НОМЕР\_ОЧЕРЕДИ) + 1; если n > N то n := N;

**если** ЗАДАЧА\_ИЗ\_CPU(ЧИСЛО\_КВАНТОВ) = 0 **то** освободить ресурс ПАМ **на объем** ЗАДАЧА\_ИЗ\_CPU (ОБЪЕМ) **иначе** **записать** ЗАДАЧА\_ИЗ\_CPU **в очередь** ОЧЕРЕДЬ(n);

ЗАДАЧА\_ИЗ\_CPU = 0;

**если условие** (ОЧЕРЕДЬ (1–N)) пуста **то перейти на метку** НАЧАЛО;  
**цикл** Ц1 **для**  $i := 1 (1) N$ ;  
**считать из** ОЧЕРЕДЬ (i) **в** РАБ\_ДИСП;  
**если** РАБ\_ДИСП  $\neq 0$  **то перейти на метку** ПРОД;  
**конец цикла** Ц1;  
ПРОД: ЗАДАЧА\_НА\_CPU := РАБ\_ДИСП;  
**перейти на метку** НАЧАЛО;  
конец описания процесса–агрегата ДИСПЕТЧЕР;

процесс–агрегат CPU;  
НАЧАЛО: **ждать условие** ЗАДАЧА\_НА\_CPU  $\neq 0$ ;  
РАБ\_CPU := ЗАДАЧА\_НА\_CPU;  
ЗАДАЧА\_НА\_CPU := 0;  
**задержать процесс на время** КВАНТ;  
ЗАДАЧА\_ИЗ\_CPU := РАБ\_CPU;  
**перейти на метку** НАЧАЛО;  
конец описания процесса–агрегата CPU;

Программа на псевдоязыке, реализующая блочно-параметрическую схему с рассредоточенными функциями, представлена ниже:

## **ОБЩИЕ ОПИСАНИЯ**

**тип** W : вектор (ОБЪЕМ, НОМЕР\_ОЧЕРЕДИ, ЧИСЛО\_КВАНТОВ)

**ресурс** – ПАМ(Q);

**тип** : массив очередей ОЧЕРЕДЬ (1 – N);

КВАНТ = K;

перед моделированием определить значения Q, P, K

процесс–агрегат ГЕНЕРАЦИЯ НЕПРЕРЫВНОГО ПОТОКА ЗАДАЧ;

НАЧАЛО: **создать** ЗАДАЧА типа W;

перед моделированием определить параметры вектора ЗАДАЧА;

**ждать условие** (ресурс ПАМ достаточен для объема ЗАДАЧА(ОБЪЕМ));

**занять ресурс** ПАМ объемом ЗАДАЧА(ОБЪЕМ);

**записать** ЗАДАЧА **в очередь** ОЧЕРЕДЬ(1);

**перейти на метку** НАЧАЛО;

конец описания процесса–агрегата ГЕНЕРАЦИЯ НЕПРЕРЫВНОГО ПОТОКА ЗАДАЧ;

процесс–агрегат БУФЕРИЗАЦИЯ;

НАЧАЛО: **ждать условие** ОЧЕРЕДЬ (1–N) не пуста;

**ждать условие** ВЫХОД\_ОЧ = 0;

**цикл** Ц1 **для**  $i := 1 (1) N$ ;

**считать из** ОЧЕРЕДЬ (i) **в** РАБ;

**если** РАБ  $\neq 0$  **то перейти на метку** ПРОД;

**конец цикла** Ц1;

ПРОД: ВЫХОД\_ОЧ := РАБ;

**перейти на метку** НАЧАЛО;

конец описания процесса–агрегата БУФЕРИЗАЦИЯ;

процесс–агрегат CPU;

ЗАДАЧА\_CPU – тип W;

НАЧАЛО: **ждать условие** ВЫХОД\_ОЧ  $\neq 0$ ;

ЗАДАЧА\_CPU := ВЫХОД\_ОЧ;

**ВЫХОД\_ОЧ := 0;**  
**задержать процесс на время КВАНТ;**  
**ЗАДАЧА\_CPU (ЧИСЛО\_КВАНТОВ) := ЗАДАЧА\_CPU(ЧИСЛО\_КВАНТОВ) – 1;**  
**ЗАДАЧА\_CPU(НОМЕР\_ОЧЕРЕДИ) := n:= ЗАДАЧА\_CPU(НОМЕР\_ОЧЕРЕДИ) + 1; если  $n > N$  то  $n := N$**   
**если ЗАДАЧА\_CPU(ЧИСЛО\_КВАНТОВ) = 0 то освободить ресурс ПАМ на объем ЗАДАЧА\_CPU(ОБЪЕМ) иначе записать ЗАДАЧА в очередь ОЧЕРЕДЬ(n);**  
**перейти на метку НАЧАЛО;**  
конец описания процесса-агрегата CPU;

Анализируя рассмотренные программы, можно сделать следующие выводы:

1) программа, реализующая схему с рассредоточенными функциями, короче и проще по сравнению с программой, реализующей схему с сосредоточенными функциями;

2) параметрическое окружение схемы с сосредоточенными функциями более концентрировано около блока диспетчера, тогда как остальные блоки имеют слабую параметризацию;

3) если в программе, реализующей схему с рассредоточенными функциями, блоки имеют достаточно простую и равномерно распределенную сложность алгоритмов, то программа блока диспетчера в схеме с сосредоточенными функциями явно перегружена логически условиями и дополнительными операторами.

Для моделирования процессов функционирования операционной системы было выбрано описание с рассредоточенными функциями. Программная имитационная модель, построенная на языке GPSS, выполняется в среде системы имитационного моделирования GPSS World [10]. Такая модель позволяет определять временные характеристики мультипрограммных и мультипроцессорных систем в зависимости от параметров планирования и диспетчеризации. Программная модель на языке GPSS приведена ниже:

```

RAM_ProcFUNCTIONRN1,C2
0,40/1,301 ; Требуемый объем ОП для процесса(Мбайт)
N_KVANT FUNCTION P$RAM_Proc,C2
40,1/300,10 ; Количество квантов процессорного времени для процесса(пропорционально объему ОП)
CLASS FUNCTION P$N_KVANT,D3
3,101/6,102/9,103 ; Класс процесса в зависимости от количества квантов (короткий, средний, длинный)
RAMSTORAGE512 ; Максимальный объем ОП(мб)
CPUSTORAGE1 ; Число процессоров
T_KVANTEQU 20; Величина кванта(мс)
N_PRIOREQU1 ; Количество очередей(приоритетов)
GENERATE (Exponential(1,0,150)),,,N_PRIOR
ASSIGN RAM_Proc,FN$RAM_Proc

```

```

ASSIGN N_KVANT,(INT(FN$N_KVANT))
ASSIGN Class,FN$CLASS
QUEUE SYS
QUEUE P$Class
QUEUE RAM
ENTER RAM,P$RAM_Proc
DEPART RAM
QUEUE K_MULT1
met2 QUEUE PR
ENTER CPU
DEPART PR
ADVANCE T_KVANT
LEAVE CPU
ASSIGN N_KVANT-,1
TEST E P$N_KVANT,0,met1
LEAVE RAM,P$RAM_Proc
DEPART SYS
DEPART P$Class
DEPART K_MULT1
TERMINATE
met1 TEST G PR,1,met2
PRIORITY (PR-1)
TRANSFER ,met2
GENERATE 5000000
TERMINATE 1
start 1 ;

```

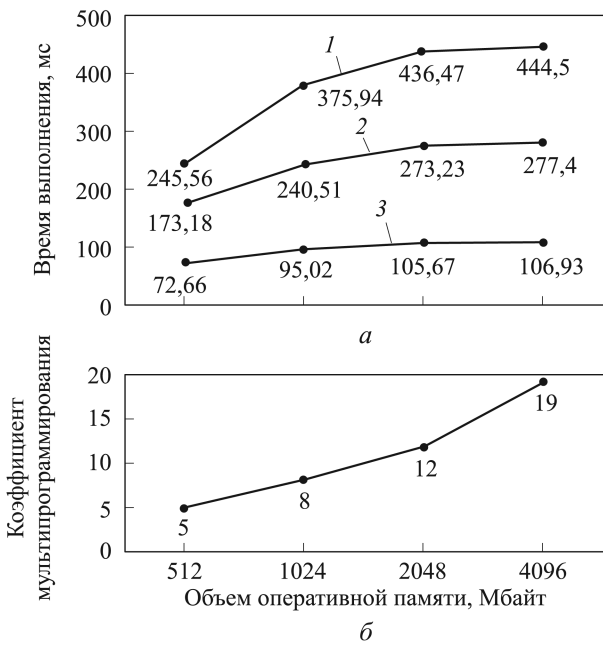
Перечислим исходные данные для имитационной модели:

- объем оперативной памяти для каждого задания, Мбайт;
- число квантов процессорного времени для каждого задания, которое определяется объемом оперативной памяти, востребованным заданием;
- число классов заданий (короткие, средние и длинные);
- среднее время выполнения задания, мс;
- общий объем оперативной памяти, Мбайт;
- длительность одного кванта процессорного времени, мс;
- число очередей;
- число процессоров.

С помощью разработанной модели исследованы зависимости времени выполнения заданий от числа очередей, объема оперативной памяти и числа процессоров. Результаты моделирования приведены на рис. 3 и 4.

Согласно зависимостям, приведенным на рис. 3, *а*, среднее время выполнения заданий с увеличением объема оперативной памяти возрастает. Это связано с повышением коэффициента мультипрограммирования (рис. 3, *б*), что приводит к увеличению времени ожидания процессорного времени для каждого задания. Время выполнения длинных





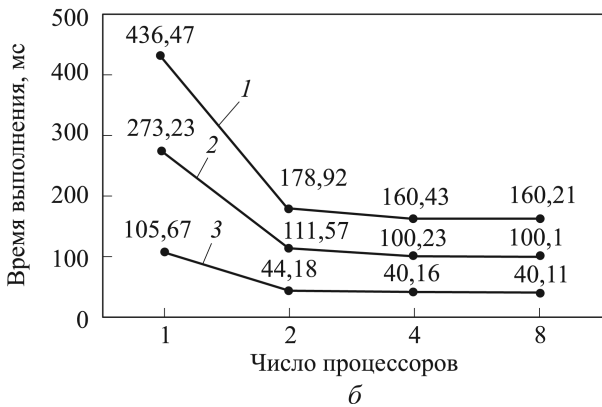
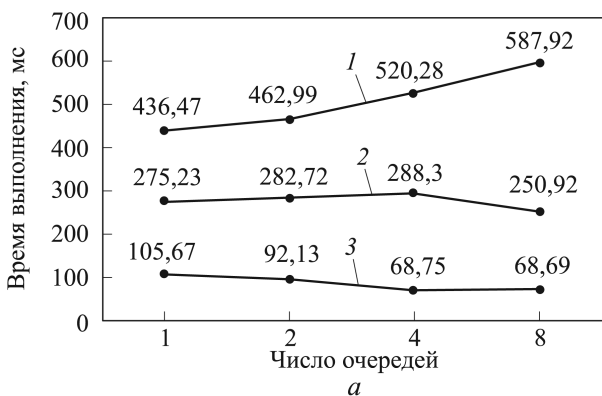
**Рис. 3.** Зависимость времени выполнения заданий от объема оперативной памяти (а) при длинных (1), средних (2) и коротких (3) заданиях, зависимость коэффициента мультипрограммирования (число одновременно выполняющихся заданий) от объема оперативной памяти (б) для одноуровневого кругового алгоритма диспетчеризации и однопроцессорной конфигурации

заданий растет быстрее, так как при данной дисциплине диспетчеризации приоритет неявно отдается коротким заданиям.

В соответствии с результатами эксперимента (рис. 4, а), с увеличением числа очередей блока диспетчера время выполнения коротких заданий уменьшается за счет увеличения времени выполнения длинных заданий. Оптимальной для представленных исходных данных является четырехпроцессорная конфигурация, так как дальнейшее увеличение числа процессоров не приводит к снижению времени выполнения (рис. 4, б). Это связано с тем, что время выполнения определяется в основном временем обработки в процессоре при отсутствии очередей.

**Заключение.** Рассмотрена методика моделирования процессов функционирования операционных систем, в котором в качестве языкового средства описания процессов функционирования использован метаязык, основанный на идее псевдоязыка описания сцепленных процессов.

Для удобства описания рассматриваемой предметной области предложены макрооператоры. Использование метаязыка позволило сравнить два варианта описания функционирования систем. Первый вариант опирается на рассредоточение функций вдоль потока обработки задачи, а второй — на сосредоточение функций в одном блоке диспетчера. Показано, что описание с рассредоточенными функциями



**Рис. 4.** Зависимости времени выполнения заданий от числа очередей для многоуровневого кругового алгоритма диспетчеризации и однопроцессорной конфигурации (а) и от числа процессоров для одноуровневого кругового алгоритма диспетчеризации и многопроцессорной конфигурации (б) при длинных (1), средних (2) и коротких (3) заданиях (объем оперативной памяти 2048 Мбайт)

короче и проще для реализации, чем описание с сосредоточенными функциями. С учетом этого можно сделать вывод, что для быстрого моделирования в целях оценки характеристики конкретного варианта архитектуры следует выбирать описание с рассредоточенными функциями вдоль потока обработки задач. В случае, когда осуществляется поиск решений в пространстве дисциплин и параметров только блока диспетчера, более эффективно выбирать описание с сосредоточенными функциями. Наличие описания функционирования системы на псевдоязыке позволяет использовать любой алгоритмический язык имитационного моделирования. Так, в работе было выполнено имитационное моделирование по блочно-параметрической схеме с рассредоточенными функциями одного из алгоритмов работы диспетчера операционной системы на языке GPSS. С помощью полученных по результатам моделирования зависимостей можно оценить эффективность алгоритмов диспетчеризации в мультипрограммных и мультипроцессорных системах.

## ЛИТЕРАТУРА

1. *Столлинс Э.В.* Операционные системы / пер. с англ. М.: Издательский дом “Вильямс”, 2004. 848 с.
2. *Таненбаум Э.* Современные операционные системы. СПб.: Питер, 2010. 1116 с.
3. *Бэкон Дж., Харрис Т.* Операционные системы. Параллельные и распределенные системы. СПб.: Питер, 2004. 800 с.
4. *Дейтел Х.М., Дейтел П. Дж., Чоффнес Д.Р.* Операционные системы. Ч. 1. Основы и принципы / пер. с англ. М.: ООО “Бином-Прессе”, 2011. 1024 с.
5. *Соломон Д., Руссинович М.* Внутреннее устройство Microsoft Windows 2000. Мастер-класс / пер. с англ. СПб.: Питер; М.: Издательско-торговый дом “Русская Редакция”, 2004. 746 с.
6. *Таненбаум Э., Вудхалл А.* Операционные системы. Разработка и реализация. СПб.: Питер, 2007. 703 с.
7. *Таненбаум Э., Ванн Стен М.* Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003. 877 с.
8. *Чёрный В.М.* Теоретические основы описания процессов функционирования дискретных систем // URL: <http://www.inforeg.ru> ФГУП “Информрегистр”. 2011. URL: <http://iu5.bmstu/nir.php> Электронное учебное издание № 0321100676. (дата обращения: 28.12.2011).
9. *Чёрный В.М.* Псевдоязык описания сцепленных процессов (ПОСП) / Электронное учебное издание № 0321100674. URL: <http://iu5.bmstu/nir.php> (дата обращения: 20.12.2011).
10. *Чёрный В.М.* Адаптированное описание системы имитационного моделирования GPSS / Электронное учебное издание № 0321100673. URL:<http://iu5.bmstu/nir.php> (дата обращения: 18.12.2011).

## REFERENCES

- [1] Stallings W. Operating Systems. New Jersey, Prentice Hall, 2001. 822 p. (Russ. ed.: Stollings E.V. Operatsionnye sistemy. Moscow, Vil'yams Publ. 2004. 848 p.).
- [2] Tanenbaum A.S. Modern operating systems. Prentice Hall, 2007. 1104 p. (Russ. ed.: Tanenbaum E. Sovremennye operatsionnye sistemy. St. Petersburg, Piter Publ., 2010. 1116 p.).
- [3] Bacon J., Harris T. Operating systems. Concurrent and distributed software design. Pearson Educ. Ltd., 2003. 877 p. (Russ.ed.: Bekon Dzh., Kharris T. Operatsionnye sistemy. Parallel'nye i raspredelennye sistemy. St Petersburg, Piter Publ., 2004. 800 p.).
- [4] Deitel H.M., Deitel P.J., Choffnes D.R. Operating systems. Part 1. Prentice Hall, 2004. 1209 p. (Russ. ed.: Deytel Kh.M., Deytel P.Dzh., Chofnes D.R. Operatsionnye sistemy. Ch. 1. Osnovy i printsipy. Moscow, “Binom-Press” Publ., 2011. 1024 p.).
- [5] Solomon D.A., Russinovich M.E. Inside Microsoft Windows 2000, Redmond, WA, Microsoft Press, 2000. 800 p. (Russ. ed.: Solomon D., Russinovich M. Vnutrennee ustroystvo Microsoft Windows 2000. Moscow, Russkaya Redaktsiya Publ., 2004. 746 p.).
- [6] Tanenbaum A.S., Woodhull A.S. Operating Systems. Design and implementation. Prentice Hall, 2006, 1080 p. (Russ. ed.: Tanenbaum E., Vudkhall A. Operatsionnye sistemy. Razrabotka i realizatsiya. St. Petersburg, Piter Publ., 2007. 703 p.).
- [7] Tanenbaum A.S., Steen M. Distributed systems. Principles and paradigms. Prentice Hall, 2002. 686 p. (Russ. ed.: Tanenbaum E., Vann Sten M. Raspredelennye sistemy. Printsipy i paradigmy. St. Petersburg, Piter Publ., 2003. 877 p.).
- [8] Chernen'kiy V.M. Teoreticheskie osnovy opisaniya protsessov funktsionirovaniya diskretnykh system [Theoretical foundation of the description of discrete systems functioning]. Available at: <http://www.inforeg.ru>, <http://iu5.bmstu/nir.php> (accessed 28.12.2011).

- [9] Chernen'kiy V.M. Pseudoyazyk opisaniya stseplennykh protsessov (POSP) [A pseudolanguage for describing linked processes]. Available at: <http://iu5.bmstu/nir.php> (accessed 20.12.2011).
- [10] Chernen'kiy V.M. Adaptirovannoe opisanie sistemy imitatsionnogo modelirovaniya GPSS [The adapted description of a GPSS simulation system]. Available at: <http://iu5.bmstu/nir.php> (accessed 18.12.2011).

Статья поступила в редакцию 30.05.2013

Валерий Михайлович Чёрненко — д-р техн. наук, заведующий кафедрой “Системы обработки информации и управления” МГТУ им. Н.Э. Баумана. Автор более 200 научных работ в области моделирования автоматизированных систем управления, теории описания параллельных процессов, систем имитационного моделирования. МГТУ им. Н.Э. Баумана, Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5.

V.M. Chernen'kii — Dr. Sci. (Eng.), head of “Systems of Information Processing and Management” department of the Bauman Moscow State Technical University. Author of more than 200 publications in the field of simulation of automated control systems, theory of description of parallel processes, systems of imitating simulation. Bauman Moscow State Technical University, Vtoraya Baumanskaya ul. 5, Moscow, 105005 Russian Federation.

Петр Степанович Семкин — доцент кафедры “Системы обработки информации и управления” МГТУ им. Н.Э. Баумана. Автор более 100 работ в области процессов функционирования операционных систем, имитационного моделирования. МГТУ им. Н.Э. Баумана, Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5.

P.S. Semkin — assoc. professor of “Systems of Information Processing and Management” department of the Bauman Moscow State Technical University. Author of more than 100 publications in the field of processes of operating-system functioning, imitating simulation. Bauman Moscow State Technical University, Vtoraya Baumanskaya ul. 5, Moscow, 105005 Russian Federation.

---

Издательство МГТУ им. Н.Э. Баумана

---

Сдано в набор 15.12.2013

Формат 70 × 108/16

Заказ

Отпечатано в типографии МГТУ им. Н.Э. Баумана

Подписано в печать 25.01.2014

Усл.-печ. л. 11,55

Уч.-изд. л. 12,35