

Методика процессного моделирования операционных систем разделения времени

77-48211/657378

12, декабрь 2013

Черненький В. М., Семкин П. С.

УДК 004.454.9

Россия, МГТУ им. Н.Э. Баумана

iu5vmch@rambler.ru

semkin@bmstu.ru

Введение

При изучении процессов функционирования систем разделения времени и их отдельных компонент возникает проблема построения адекватного и формализованного описания этих процессов. Такие описания необходимы для проведения анализа вариантов, для построения математических моделей, для проведения имитационного моделирования, для изложения этого материала в учебных целях. В работах [1-7] изложены принципы функционирования операционных систем разделения времени и их компонент, однако они не содержат формализованного описания процессов их функционирования. В связи с этим существенно затрудняется анализ характеристик этих систем методами имитационного моделирования. Поэтому излагаемые в настоящей статье предложения по методике описания процессов в системах разделения времени востребованы в практике моделирования.

Основной текст.

Существенную роль в формировании таких показателей, как пропускная способность, время выполнения заданий, коэффициент загрузки процессора в системах разделения времени играет эффективность

алгоритмов планирования параллельно выполняющихся процессов [6,7], особенно при использовании высокопроизводительных сетевых рабочих станций и серверов. Наиболее распространенным методом анализа в этом случае является имитационное моделирование.

Рассмотрим применение методики процессных описаний [8] при исследовании циклического алгоритма планирования, который является основным в современных системах разделения времени. Новые задания попадают во входную очередь системы, в которой они ждут освобождения адресного пространства оперативной памяти. Из входной очереди после получения доли оперативной памяти, задания попадают в очередь номер 1, готовых к выполнению. Задания в этой очереди ожидают освобождения процессора. Каждому процессу выделяется фиксированный квант времени q , в конце которого, если задание к этому времени не закончится, оно покидает процессор и помещается в очередь с номером, на единицу большую. Общее количество очередей N ограничено и является управляемым параметром. Если задание достигает очереди N , то оно остается в дальнейшем в этой же очереди. Диспетчер выбирает для обслуживания заявку из i -ой очереди, если пусты все предыдущие очереди.

Для проведения имитационного моделирования необходимо описать совокупность параллельных взаимосвязанных процессов, отражающих функционирование системы. В первую очередь, необходимо описать функциональную структуру. Используя вышеприведенное описание, можно предложить два варианта функциональной структуры, приведенных на рисунках 1 и 2.

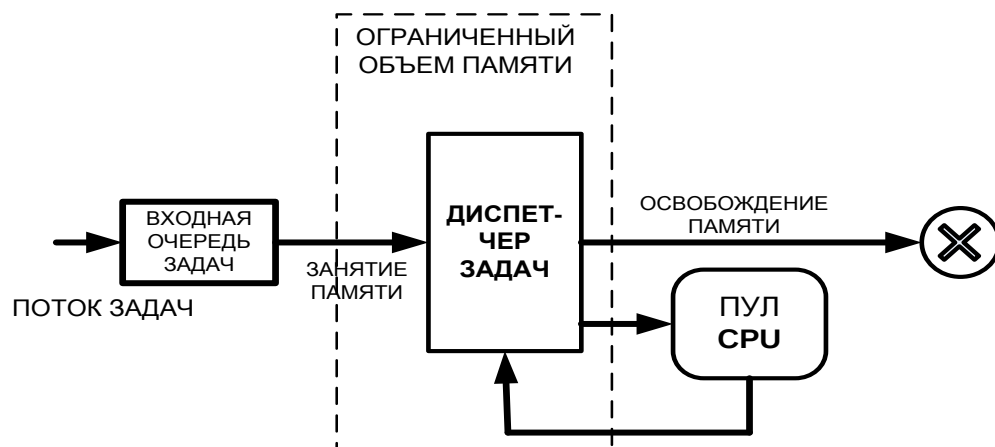


Рисунок 1. Функциональная схема с сосредоточенными функциями

На рисунке 1 представлена схема с максимальным сосредоточением функций. Она содержит три функциональных блока: генератор потока, диспетчер и блок процессоров. Большинство функций управления потоком сосредоточены в блоке «диспетчер». Остальные два блока выполняют простые функции: подача очередной задачи в систему и задержка задачи на некоторое время обработки, соответственно.

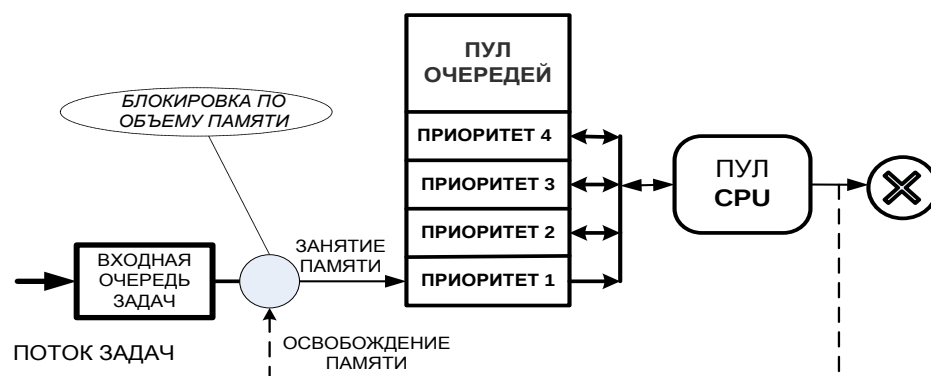


Рисунок 2. Функциональная схема с рассредоточенными функциями

На рисунке 2 представлена схема с рассредоточенными функциями вдоль потока задач. Как видно, здесь блок «диспетчер» преобразовался в блок управления очередями, а функции других блоков усложнены.

Далее, на основе построенных функциональных схем необходимо построить блочно - параметрические схемы, содержащие блоки,

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

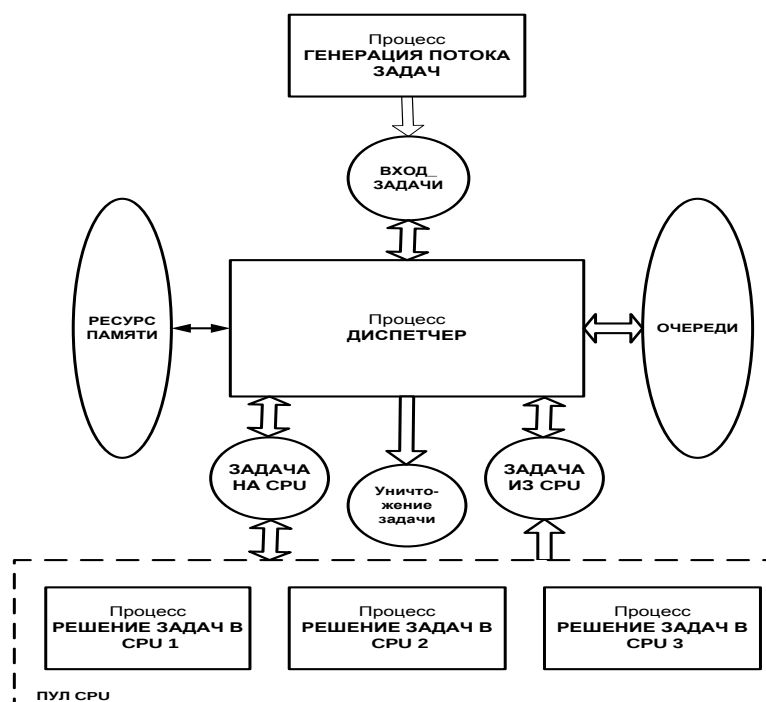


Рисунок 3. Блочно - параметрическая схема с сосредоточенными функциями

На рисунке 3 показана блочно - параметрическая схема, отражающая схему с сосредоточенными функциями. Видно, что генератор потока выдает поток задач, помещая каждую задачу в параметр ВХОД_ЗАДАЧИ. Блок процессоров CPU взаимодействует с блоком диспетчера через параметры ЗАДАЧА_НА_CPU и ЗАДАЧА_ИЗ_CPU.

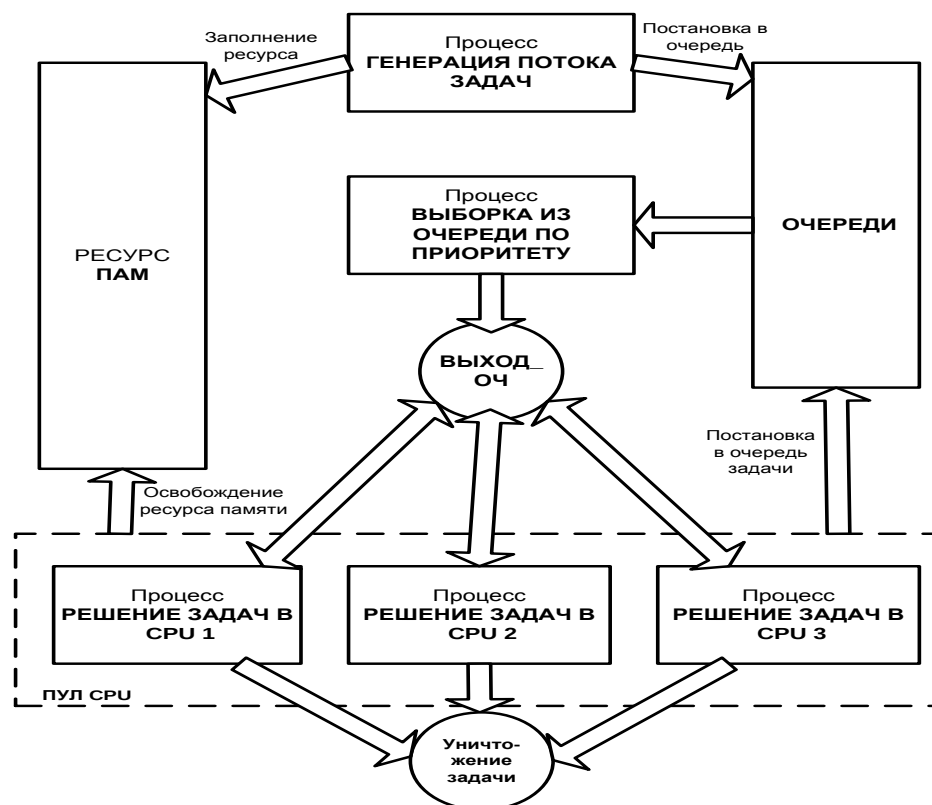


Рисунок 4. Блочно - параметрическая схема с рассредоточенными функциями

На рисунке 4 показана блочно - параметрическая схема, отражающая схему с рассредоточенными функциями. Как видно, распределение функций привело и к изменению состава параметров схемы.

Далее блочно – параметрические схемы должны быть реализованы с помощью языковых средств, в качестве которых предлагается использовать метаязык, изложенный в [9]. Понятие «агрегат» соответствует концепции [8]. Поскольку в предлагаемых ниже описаниях использованы лишь агрегаты, то понятие «инициатор» опущено.

Ниже приведена программа на псевдоязыке, реализующая блочно - параметрическую схему с сосредоточенными функциями. В этом описании служебные слова выделены жирным шрифтом, имена параметров заданы заглавными обычными буквами, комментарии подчеркнуты.

**ПРОГРАММА РЕАЛИЗАЦИИ БЛОЧНО - ПАРАМЕТРИЧЕСКОЙ СХЕМЫ С
СОСРЕДОТОЧЕННЫМИ ФУНКЦИЯМИ**

ОБЩИЕ ОПИСАНИЯ

тип W : вектор (ОБЪЕМ, НОМЕР_ОЧЕРЕДИ, ЧИСЛО_КВАНТОВ);

ресурс – ПАМ(Q);

тип : массив очередей ОЧЕРЕДИ (1-N);

КВАНТ := K;

ВХОД_ЗАДАЧА, ЗАДАЧА_ИЗ_CPU, ЗАДАЧА_НА_CPU := 0;

перед моделированием определить значения Q, K, N

процесс-агрегат ГЕНЕРАЦИЯ ПОТОКА ЗАДАЧ;

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

ЗАДАЧА(НОМЕР_ОЧЕРЕДИ) := 1;

определить остальные параметры вектора ЗАДАЧА,

ВХОД_ЗАДАЧА := ЗАДАЧА;

уничтожить ЗАДАЧА;

ждать условие ВХОД_ЗАДАЧА = «пусто»;

перейти на метку НАЧАЛО;

конец описания процесса-агрегата ГЕНЕРАЦИЯ ПОТОКА ЗАДАЧ;

процесс-агрегат ДИСПЕТЧЕР;

НАЧАЛО: **ждать условие1** (ВХОД_ЗАДАЧА ≠ 0) **и** (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):

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

ВХОД_ЗАДАЧА := «пусто»;

перейти на метку НАЧАЛО;

ОБРАБ: ЗАДАЧА_ИЗ_CPU (ЧИСЛО_КВАНТОВ) := ЗАДАЧА_ИЗ_CPU (ЧИСЛО_КВАНТОВ) – 1;

если ЗАДАЧА_ИЗ_CPU (НОМЕР_ОЧЕРЕДИ) = N **то перейти на** М1;

ЗАДАЧА_ИЗ_CPU (НОМЕР_ОЧЕРЕДИ) := ЗАДАЧА_ИЗ_CPU (НОМЕР_ОЧЕРЕДИ) + 1;

М1: **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(ЗАДАЧА_ИЗ_CPU (НОМЕР_ОЧЕРЕДИ));

ЗАДАЧА_ИЗ_CPU := «пусто»;

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 ≠ 0 **то перейти на метку** ПРОД;

конец цикла Ц1;

перейти на метку НАЧАЛО;

ПРОД: ЗАДАЧА_НА_CPU := РАБ_ДИСП;

перейти на метку НАЧАЛО;

конец описания процесса-агрегата ДИСПЕТЧЕР;

процесс-агрегат CPU;

запустить агрегат в нулевой момент времени из ОЧЕРЕДЬ(1);

НАЧАЛО: **ждать условие** ЗАДАЧА_НА_CPU ≠ 0;

РАБ_CPU := ЗАДАЧА_НА_CPU;

ЗАДАЧА_НА_CPU := «пусто»;
задержать процесс на время КВАНТ;
ЗАДАЧА_ИЗ_CPU := РАБ_CPU;
перейти на метку НАЧАЛО;
конец описания процесса-агрегата CPU;

Ниже приведена программа на псевдоязыке, реализующая блочно - параметрическую схему с распределенными функциями. В этом описании служебные слова выделены жирным шрифтом, имена параметров заданы заглавными обычными буквами, комментарии подчеркнуты.

ПРОГРАММА РЕАЛИЗАЦИИ БЛОЧНО - ПАРАМЕТРИЧЕСКОЙ СХЕМЫ С РАСПРЕДЕЛЕННЫМИ ФУНКЦИЯМИ

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;

 ЗАДАЧА(НОМЕР_ОЧЕРЕДИ) := 1;

определить остальные параметры вектора ЗАДАЧА,

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

занять ресурс ПАМ объемом ЗАДАЧА(ОБЪЕМ);

записать ЗАДАЧА в очередь ОЧЕРЕДЬ(1);

уничтожить ЗАДАЧА;

перейти на метку НАЧАЛО;

конец описания процесса-агрегата ГЕНЕРАЦИЯ 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 ≠ 0 **то перейти на метку** ПРОД;

конец цикла Ц1;

перейти на метку НАЧАЛО;

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 ≠ 0;

```

ЗАДАЧА_CPU := ВЫХОД_ОЧ;
ВЫХОД_ОЧ := 0;
задержать процесс на время КВАНТ;
ЗАДАЧА_CPU (ЧИСЛО_КВАНТОВ) := ЗАДАЧА_CPU(ЧИСЛО_КВАНТОВ) – 1;
если ЗАДАЧА_CPU(НОМЕР_ОЧЕРЕДИ) = N то перейти на М1;
ЗАДАЧА_CPU(НОМЕР_ОЧЕРЕДИ) := ЗАДАЧА_CPU(НОМЕР_ОЧЕРЕДИ) + 1;
М1: если ЗАДАЧА_CPU(ЧИСЛО_КВАНТОВ) = 0 то освободить ресурс ПАМ на объем
    ЗАДАЧА_CPU (ОБЪЕМ) иначе записать ЗАДАЧА_CPU в очередь
    ОЧЕРЕДЬ(ЗАДАЧА_CPU(НОМЕР_ОЧЕРЕДИ));

перейти на метку НАЧАЛО;
конец описания процесса-агрегата CPU;

```

Анализируя эти программы, можно сделать следующие выводы:

- 1) видно, что параметрическое окружение для схемы с сосредоточенными функциями собрано около блока «диспетчер», тогда как остальные блоки имеют более слабую параметризацию;
- 2) логика программы «диспетчер» в схеме с сосредоточенными функциями достаточно сложная;
- 3) программа с распределенными функциями короче и проще для описания по сравнению с программой с сосредоточенными функциями;

Было выполнено моделирование процессов выполнения заданий в системе разделения времени. В качестве формализованного описания было выбрано описание с распределенными функциями. Модель составлена на языке имитационного моделирования GPSS. Модель функционирует в среде свободно распространяемой системы имитационного моделирования GPSS World [10]. С помощью разработанной модели определены временные характеристики пребывания заданий в системе в зависимости от изменения параметров. Ниже приведён текст модели на языке GPSS .

```

RAM_Proc FUNCTION RN1,C2
0,40/1,301 ;      Объём адресного пространства процесса(Мб)
N_KVANT FUNCTION P$RAM_Proc,C2
40,1/300,10 ;    Количество квантов времени для процесса
CLASS FUNCTION P$N_KVANT,D3

```

3,101/6,102/9,103 ; Класс задания в зависимости от количества квантов (интерактивный, программный, фоновый)

RAM STORAGE 512 ; Объем оперативной памяти системы

CPU STORAGE 1 ; Число процессоров системы

T_KVANT EQU 20; Величина кванта(мс)

N_PRIOR EQU 1 ; Число 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_MULTI

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_MULTI

TERMINATE

met1 TEST G PR,1,met2

PRIORITY (PR-1)

TRANSFER ,met2

GENERATE 5000000

TERMINATE 1

start 1;

Исходные данные для имитационной модели.

- Размер адресного пространства задания (Мбайт).
- Количество квантов времени для выполнения задания, пропорциональное размеру адресного пространства, необходимого заданию.
- Классов заданий (в модели: интерактивные, программные, фоновые)
- Среднее время появления новых заданий в системе (Мс).
- Объем оперативной памяти системы (Мбайт).

- Размер кванта процессорного времени (Мс).
- Число очередей (приоритетов).
- Число процессоров

Разработанная модель была использована для исследования зависимости времени выполнения заданий каждого класса от числа приоритетов, объёма оперативной памяти и числа процессоров. На рисунках 5-8 представлены результаты моделирования.

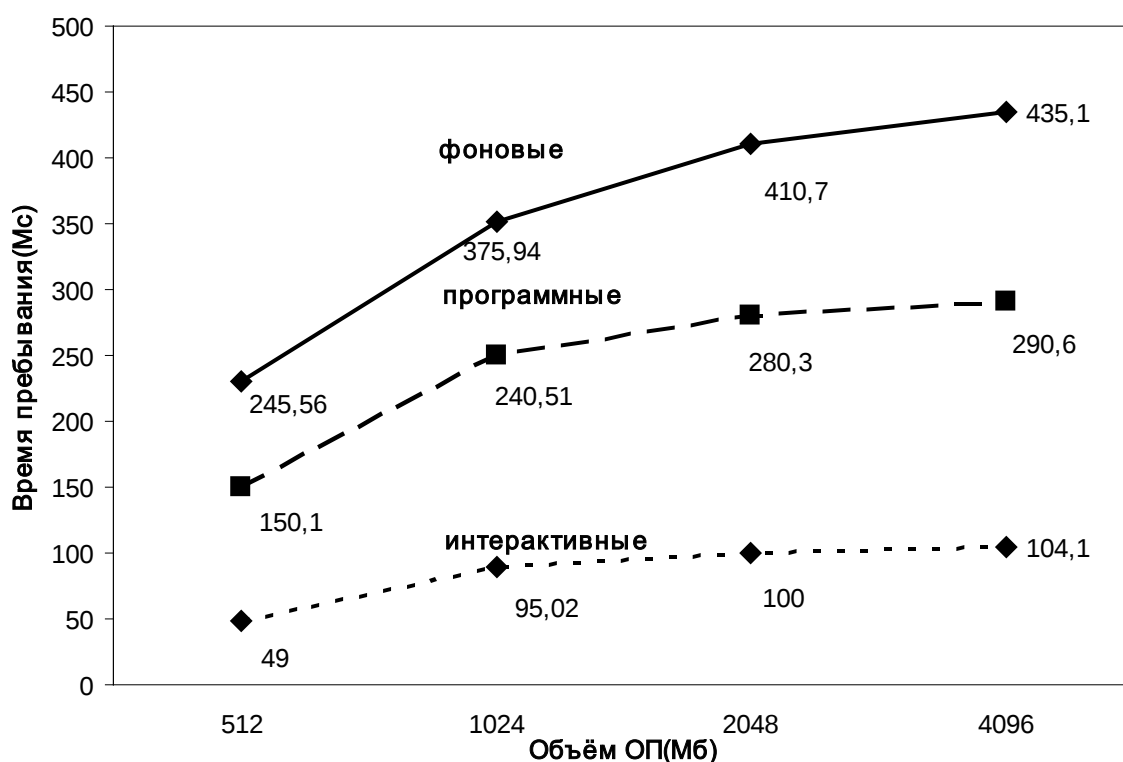


Рисунок 5. Время пребывания заданий в зависимости от объёма оперативной памяти для беспriorитетного алгоритма планирования и однопроцессорной конфигурации

Из результатов эксперимента видно, что с увеличением размера оперативной памяти время пребывания в системе заданий всех классов растёт. Этот рост обусловлен увеличением коэффициента мультипрограммирования (количеством одновременно выполняющихся процессов) (рисунок 6), что приводит к увеличению времени ожидания

обслуживания для всех заданий. Время пребывания фоновых заданий растёт быстрее, чем интерактивных, так как при данной дисциплине планирования приоритет неявно отдаётся интерактивным (коротким) заданиям.

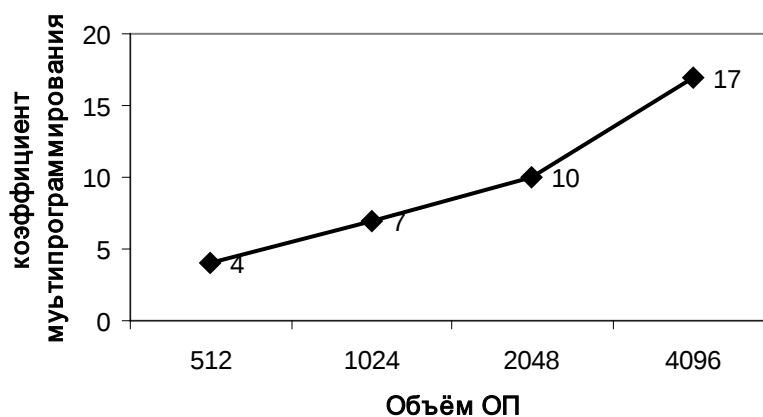


Рисунок 6. Коэффициент мультипрограммирования в зависимости от объема оперативной памяти для беспriorитетного алгоритма планирования и однопроцессорной конфигурации

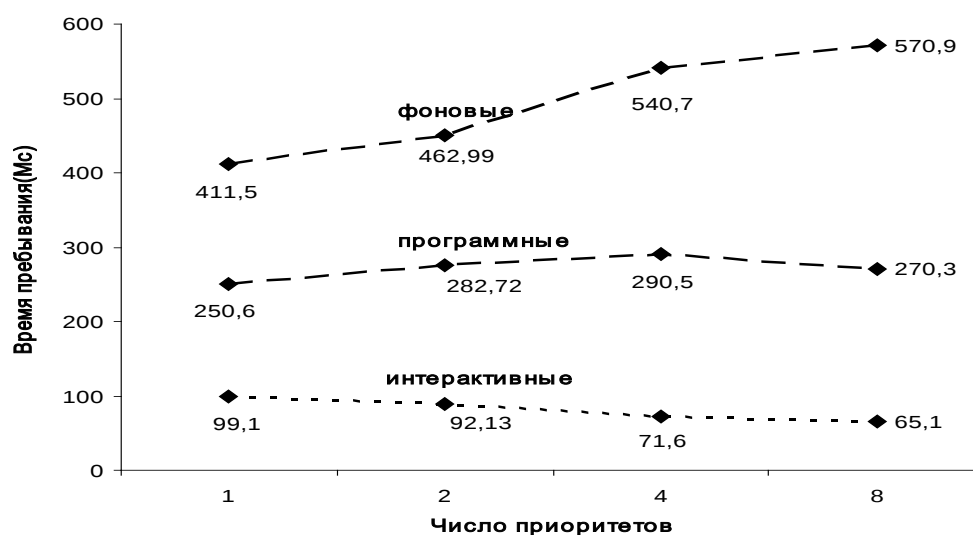


Рисунок 7. Время пребывания заданий в зависимости от числа приоритетов для многоуровневого алгоритма планирования и однопроцессорной конфигурации.

Из результатов эксперимента, представленных на рисунке 7 видно, что с увеличением числа приоритетов время выполнения интерактивных заданий уменьшается за счёт увеличения времени выполнения остальных заданий.

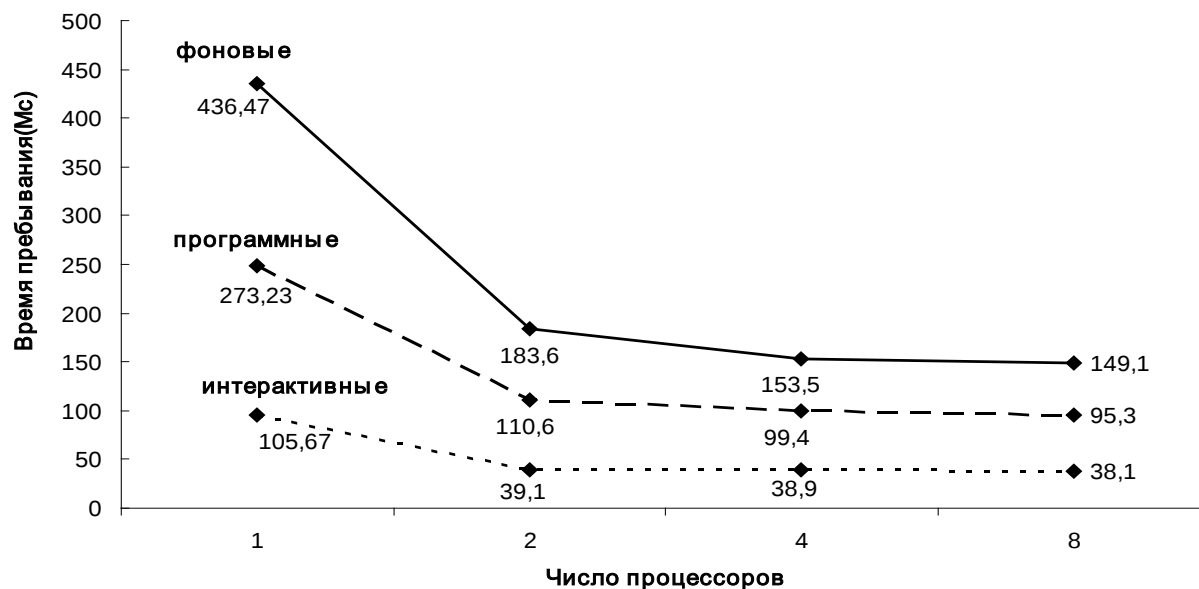


Рисунок 8. Время пребывания заданий в зависимости от числа процессоров для беспriorитетного алгоритма планирования и многопроцессорной конфигурации.

Из результатов экспериментов, представленных на рисунке 8, видно, что при заданной интенсивности поступления новых заданий увеличение числа процессоров приводит к уменьшению времени пребывания только для двух и четырёхпроцессорных конфигураций. Дальнейший рост числа процессоров не даёт желаемого эффекта, так как время пребывания заданий определяется только временем обработки в процессоре и отсутствием ожидания.

Заклучение

В статье излагается методика моделирования процессов функционирования систем разделения времени. За основу начальной

формализации выбраны принципы процессных описаний, изложенные в [8]. В качестве языкового средства использован псевдоязык описания сцепленных процессов [5]. Для простоты описания в язык введен ряд макрооператоров. В результате сформированы два варианта описания: первое, опирающееся на распределение функций вдоль трека обработки задачи, и второе, опирающееся на сосредоточение функций в одном блоке «диспетчер». Описание функционирования системы на псевдоязыке позволяет достаточно просто выполнить имитационное моделирование на языке имитационного моделирования. В статье было выполнено имитационное моделирование на языке GPSS. По результатам моделирования построены графики, позволяющие оценить временные характеристики пребывания заданий в системах разделения времени.

Список литературы

1. Э Столлингс, Вильям. Операционные системы, 4-е издание. : Пер. с англ. Издательский дом «Вильямс», 2004. – 848 с.
2. Таненбаум Э. Современные операционные системы. 3-е изд. – СПб.: Питер, 2010. 1116 с.: ил.
3. Джин Бэкон, Тим Харрис. Операционные системы Параллельные и распределенные системы. СПб.: Питер, 2004. 800 с.: ил.
4. Х.М. Дейтел, П. Дж. Дейтел, Д.Р. Чофнес. Операционные системы. Часть 1. Основы и принципы: Третье издание. Пер. с англ. – М.: ООО «Бином-Пресс», 2011 г. – 1024 с.: ил.
5. Соломон Д. и Руссинович М. Внутреннее устройство Microsoft Windows 2000. Мастер-класс / Пер. с англ.- СПб.: Питер; М.: Издательско-торговый дом «Русская Редакция», 2004. – 746 стр.: ил.
6. Таненбаум Э, А. Вудхалл Операционные системы. Разработка и реализация . 3-е издание СПб.: Питер, 2007. 703 с.: ил.
7. Таненбаум Э, М. ванн Стен. Распределённые системы Принципы и парадигмы. СПб.: Питер, 2003. 877 с. :ил.
8. Черненький В.М. Теоретические основы описания процессов функционирования дискретных систем // <http://www.inforeg.ru> ФГУП «Информрегистр». 2011 URL <http://iu5.bmstu/nir.php> (Электронное учебное издание № 0321100676) , (дата обращения 28.10.13)

9. Черненький В.М. Псевдоязык описания сцепленных процессов (ПОСП) // <http://www.inforeg.ru> ФГУП «Информрегистр». 2011 URL <http://iu5.bmstu/nir.php> (Электронное учебное издание № 0321100674) , (дата обращения 20.9.13)
10. Черненький В.М. Адаптированное описание системы имитационного моделирования GPSS // <http://www.inforeg.ru> ФГУП «Информрегистр». 2011 URL <http://iu5.bmstu/nir.php> (Электронное учебное издание № 0321100673) , (дата обращения 18.10.13)