

## Общие указания для выполнения лабораторных работ по моделированию процессов вычислений в МВС

Цикл лабораторных работ посвящен одному из основных разделов курса, связанному с организацией вычислений в многопроцессорных вычислительных системах (**МВС**). Рассматриваемые системы относятся к классу "множественный поток команд, множественный поток данных" (**МКМД**).

Основными компонентами современных МВС являются:

- **решающее поле**, состоящее из однородных или разнородных процессоров, со своей локальной памятью (или без нее);
- **коммутационная сеть**, позволяющая вести обмен данными, как между процессорами, так и между процессорами и общей памятью;
- **общая память модульного типа**;
- **иерархическая система управления**, позволяющая реализовать параллелизм обработки данных на уровне независимых задач, независимых ветвей задач, команд и операций.

В качестве параметров компонент системы используются:

- **решающее поле** – число процессоров (в данном цикле лабораторных работ рассматриваются ***МВС с одинаковыми процессорами***), характеризующихся относительным временем выполнения независимых ветвей задачи, определяемым быстродействием процессоров;
- **локальная память (ЛП) процессора** - емкость (предполагается, что быстродействие ЛП таково, что ***не вносит задержку при обращении*** к ней процессора за данными, которые в ней находятся);
- **коммутационная сеть (КС)** – определяется типом (***общая шина или мультиплексная шина***), числом шин, пропускной способностью при передаче данных (характеризуется относительным временем занятия сформированного канала: «процессор – процессор», «процессор - общая память», «общая память – процессор»);
- **общая память** - число модулей памяти, быстродействие, косвенно отображаемое временем занятия канала процессор - общая память, общая память - процессор.

***Затраты на управление в моделях не учитываются.***

На вход такой МВС в общем случае может поступать:

1. ***одна сложная задача***, декомпозированная на подзадачи (***однозадачный режим*** функционирования МВС) или
2. набор независимых задач, каждая из которых может иметь свой приоритет (***многозадачный режим*** функционирования МВС).

**В качестве модели задачи используется взвешенный ориентированный (направленный) граф, узлы которого отображают подзадачи. Каждой подзадаче соответствует часть (ветвь) программы, начав выполнение которой процессор заканчивает ее без прерываний. Дуги графа моделируют связи по данным между соответствующими подзадачами (ветвями программы).**

Узлы графа взвешиваются целыми числами, соответствующими временам их выполнения в условных единицах (например, тактах) на процессорах. Дуги графа взвешиваются тоже целыми числами, соответствующими временам занятия канала связи (шины) при передаче данных от одного узла графа к другому.

В настоящем цикле лабораторных работ рассматриваются графы задач без обратных связей, с различным соотношением времен выполнения узлов графа ( $t_{\text{вып}}$ ) и передачи данных ( $t_{\text{пер}}$ ). В соответствие с этим соотношением различают

**слабосвязные задачи ( $t_{\text{вып}} \gg t_{\text{пер}}$ ),**

**среднесвязные ( $t_{\text{вып}} \approx t_{\text{пер}}$ ),**

**сильносвязные ( $t_{\text{вып}} \ll t_{\text{пер}}$ ).**

Каждая задача может характеризоваться максимальным временем выполнения  $T_{\text{max}}$  и минимальным временем  $T_{\text{min}}$ .

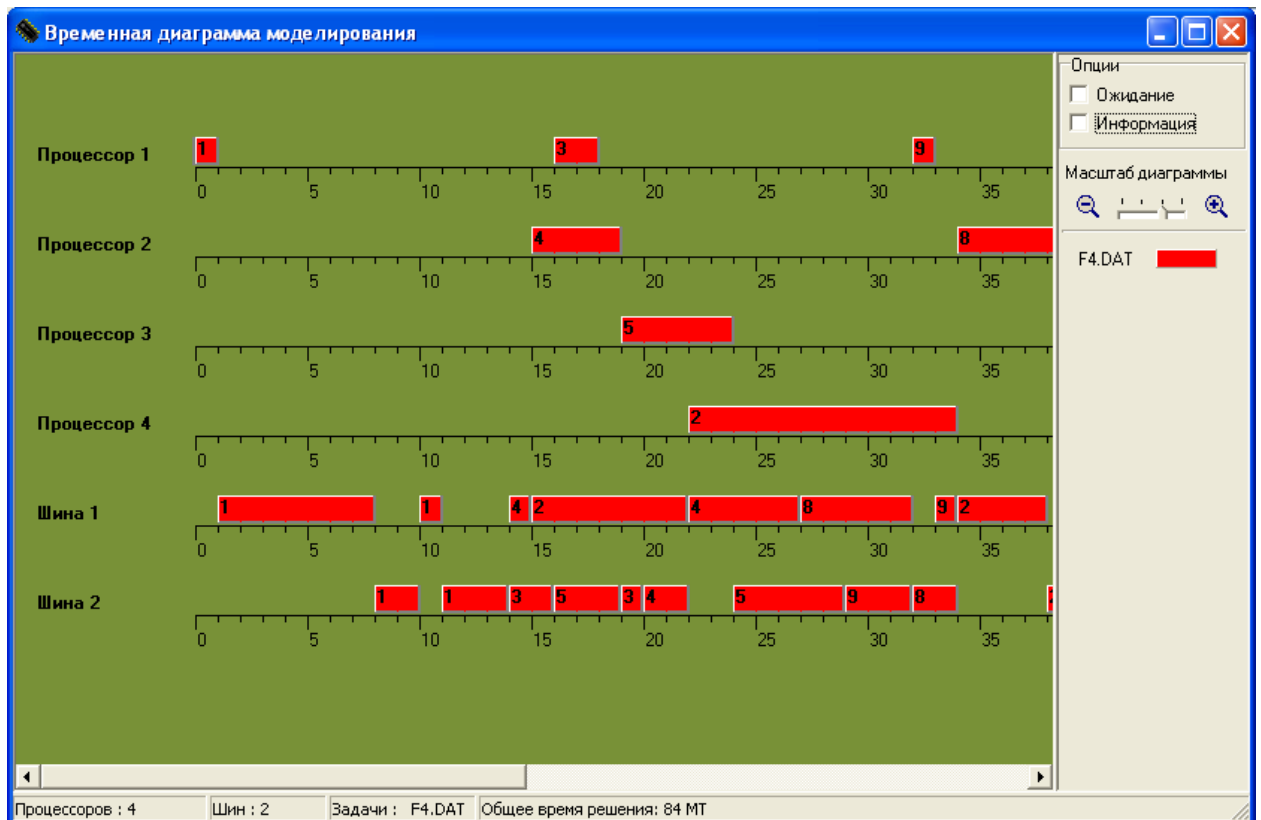
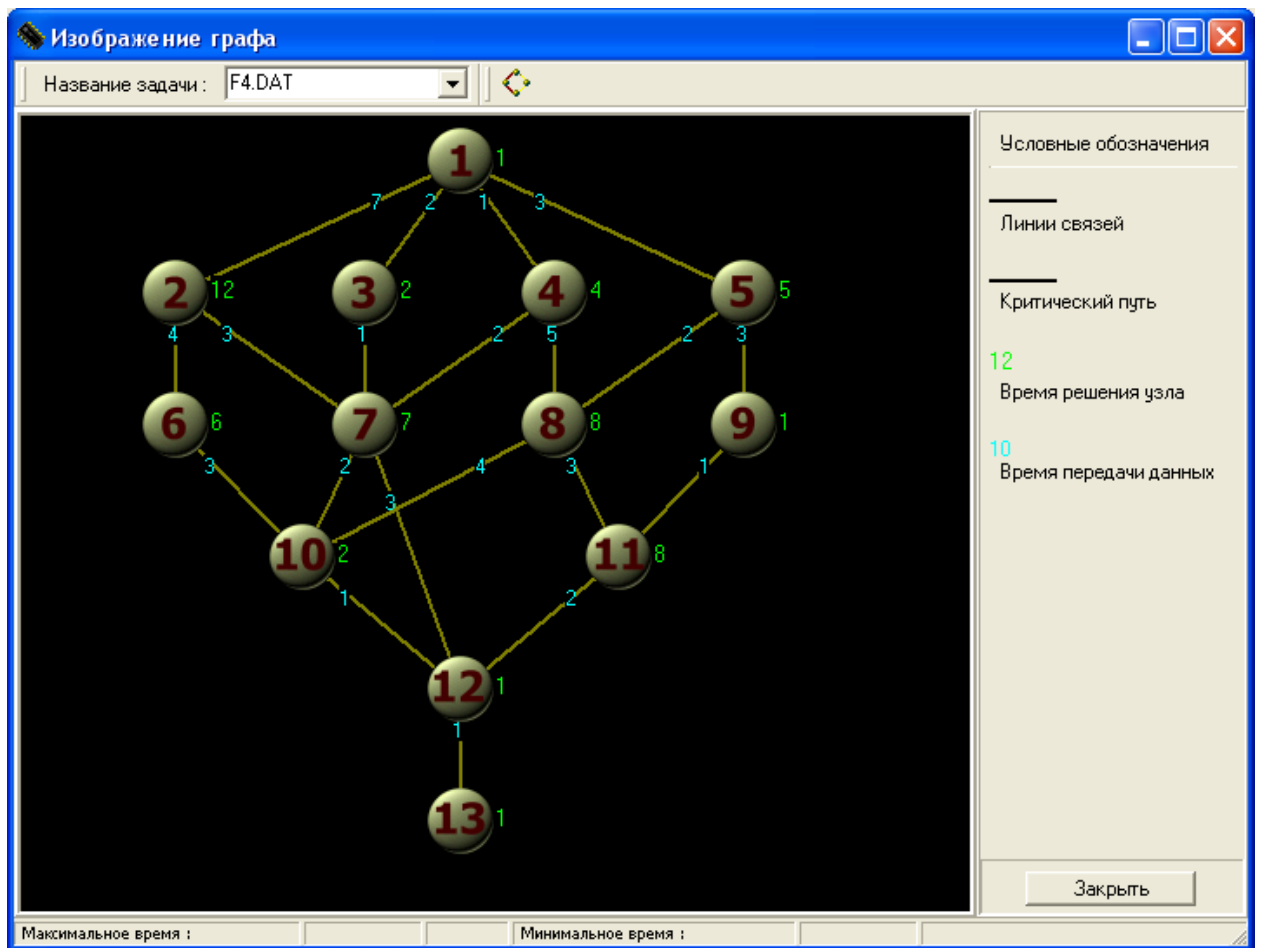
Например, для задач, у которых для всех узлов  $t_{\text{пер}}=0$ ,

$T_{\text{max}}$  вычисляется как **сумма времен выполнения всех узлов графа** (следует иметь в виду, что это справедливо, если все узлы графа выполняются на одном и том же процессоре), а

$T_{\text{min}}$  - как **сумма времен выполнения узлов графа, принадлежащих его критическому пути.**

Приведем пример графа слабосвязной задачи и результаты процесса его моделирования в МВС с общей памятью с использованием **4-х** процессоров и **2-х** шин.

Задача слабосвязная, т.к. среднее значение трудоемкости передачи данных < средней трудоемкости выполнения узлов



Лабораторные работы по моделированию включают:

## Лабораторная работа №2

### Исследование функционирования многопроцессорной системы в однозадачном режиме

**Цель работы:** изучение принципов распределения узлов вычислительного процесса (ВП) в *однозадачном режиме* с использованием различных стратегий назначения готовых к исполнению узлов (подзадач).

## Лабораторная работа № 3

### Организация многозадачного режима выполнения вычислительного процесса в МВС

**Цель работы:** исследование задачи составления расписания выполнения ВП в многозадачном режиме в МВС с использованием различных стратегий назначения готовых узлов, с учетом приоритета выполняемой задачи и без учета приоритета; изучение методики выбора характеристик МВС (количество процессоров, стратегия назначения, приоритетность задачи) для достижения заданного времени выполнения набора задач.

**Общее:** 1. В обеих работах рассматривается некоторая абстрактная МВС, в которой при выполнении задачи не учитывается время передачи данных между подзадачами.

2. Объекты исследования – а) граф с числом вершин порядка 22-23

(Вариантов 30) и

б) граф с числом вершин не более 13-14  
(Вариантов 30)

3. Оба графа необходимо описать в матричной форме для представления в памяти \*)

3. Для обоих графов необходимо **самостоятельно рассчитать** критический путь для определения минимально возможного времени решения задачи (рассчитанный критический путь можно сравнить с найденным в программе)

4. В работах исследуется применение различных стратегий назначения готовых узлов графа для выполнения в МВС:

в работе №2 анализируются 2 стратегии выбора узла: «с максимальным временем выполнения» и «с минимальным временем выполнения узла»

в работе №3 анализируются 3 стратегии выбора узла: «с максимальным временем выполнения», «с минимальным временем выполнения узла» и «принадлежащего критическому пути».

## Лабораторная работа № 4

### Исследование принципов организации вычислительного процесса в МВС с общей памятью

**Цель работы:** изучение способов организации ВП при выполнении набора задач различных типов на МВС с общей памятью (рис. 1) и шинной организацией коммутации.

Результатами должны являться определение параметров МВС (количество процессоров, шин), позволяющих выполнить набор задач конкретного типа за заданное время.

## Лабораторная работа № 5

### Исследование принципов организации вычислительного процесса в МВС с распределенной памятью

**Цель работы:** изучение способов организации вычислительного процесса в МВС с распределенной памятью для определения параметров МВС (количество процессоров и шин), позволяющих выполнить задачу за заданное время; сравнение полученных характеристик с аналогичными характеристиками для МВС с общей памятью.

**Общее:** 1. В обеих работах при выполнении задач **у ч и т ы в а е т с я** время передачи данных между подзадачами.

2. В каждой работы сначала исследуется **однозадачный режим** функционирования МВС (для одного **графа**, заданного в текстовом файле, для которого необходимо определить самостоятельно КП)
3. Затем исследуется **многозадачный режим**, где объекты исследования – **три различных наборы из двух однотипных задач**:
  - **слабосвязные задачи**, в которых время выполнения узлов задачи много больше времени передач между узлами  $t_p \gg t_n$ ;
  - **среднесвязные задачи**, в которых  $t_p \approx t_n$ ;
  - **сильносвязные задачи**, в которых  $t_p \ll t_n$ .

**Рассмотрим пример** моделирования процессов вычислений в МВС для графа задачи, приведенного на рис.1.

Для этого определим **критический путь на графе и минимальное время выполнения графа  $T_{\min}$**  задачи, которое соответствует значению критического пути.

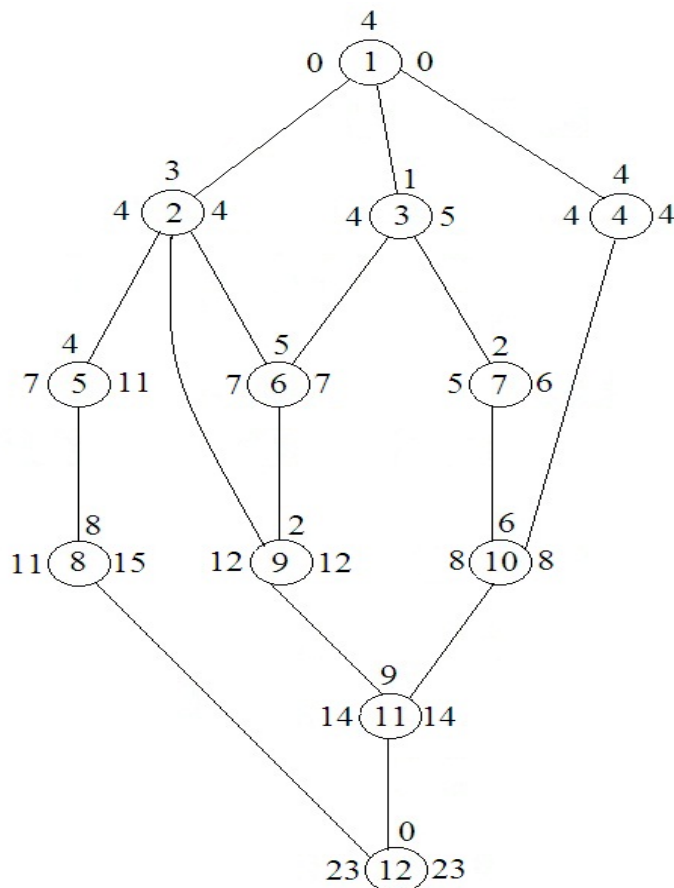


Рис. 1. Граф анализируемой задачи.

Для этого графа задачи  $T_{\max}=48$ ,  $T_{\min}=23$ , следовательно, первое приближение числа процессоров МВС, с помощью которых можно выполнить задачу за минимальное время  $n=3$  (ближайшее большее целое от значения  $T_{\max}/T_{\min}$ ).

Рассмотрим подробно алгоритм поиска критического пути на графе.

Суть алгоритма заключается в **определении минимально возможного и максимально возможного времени начала выполнения узлов графа**. В предлагаемом комплексе лабораторных работ используется следующий алгоритм определения критических путей (КП) в графе со взвешенными узлами и дугами, то есть с учетом времен обработки узлов графа задачи и передачи данных между узлами графа. Данный алгоритм, аналогично соответствующему алгоритму для графа без учета времени передачи, основан на последовательном проходе по дереву графа от начальной вершины к конечной и возврата к начальной вершине. При начальном проходе по графу

определяется минимально возможное время  $T_{\min}$  начала выполнения каждого узла по формуле:

$$T_{\min i} = \max_{j=1}^s (T_{\min j} + t_j + T_{ij}) \quad (1)$$

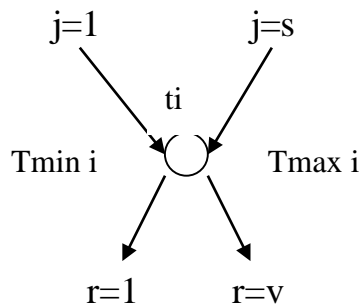
где  $s$  - число узлов-предшественников  $i$ -го узла;

$T_{\min i(j)}$  - минимально возможное время начала выполнения  $i$ -го ( $j$ -го) узла;

$t_j$  - время выполнения  $j$ -го узла;

$T_{ji}$  - время передачи данных между узлами  $j$  и  $i$ , которому присваивается одно из значений множества  $\{0, \tau_{ji}, 2\tau_{ji}\}$  в зависимости от способа организации памяти, где  $\tau_{ji}$  - время передачи данных между узлами  $j$  и  $i$ , задаваемое на исходном графе задачи как вес ребра между  $j$  и  $i$  вершинами.

Алгоритм поиска критических узлов можно проиллюстрировать схемой:



При повторном анализе графа при проходе от конечной вершины к начальной определяется максимально возможное время начала выполнения узла по формуле:

$$T_{\max i} = \min_{r=1}^v (T_{\max r} - t_i - T_{ir}) \quad (2)$$

где  $v$  - число узлов-последователей  $i$ -го узла;

$T_{\max i(r)}$  - максимально возможное время начала выполнения  $i$ -го ( $r$ -го) узла;

$t_i$  - время выполнения  $i$ -го узла;

$T_{ir}$  - время передачи данных от  $i$ -го узла к узлу  $r$ , значение которому присваивается аналогично  $T_{ij}$

При этом для начальной вершины  $T_{\min i} = 0$ , а для конечной вершины минимально возможное время начала ее выполнения совпадает с максимально возможным временем начала выполнения -  $T_{\max i} = T_{\min i}$ .

Узел  $i$ -ый является *критическим*, если выполняется равенство  $T_{\max i} = T_{\min i}$

**Критическим путем графа** является множество последовательных узлов, начинающихся с входной вершиной и заканчивающихся выходной вершиной, удовлетворяющих условию  $T_{\max i} = T_{\min i}$ , причем для каждой пары

узлов принадлежащих КП соотношения (1) и (2) определяются соседней вершиной в паре.

**Длина критического пути определяет минимальное возможное время выполнения графа задачи на МВС.**

Граф задачи при выполнении ее на МВС с различной организацией может иметь различные критические пути вследствие изменения времени передач между узлами. Последнее определяется способом организации памяти.

Например, в МВС только с общей памятью (без локальной памяти процессоров) при определении критического пути время передачи данных от  $j$ -го узла  $i$ -му и от  $i$ -го узла  $j$ -му необходимо удваивать. Двойной учет этого времени происходит потому, что при передаче данных, во-первых, необходимо время  $\tau_{ji}$ , чтобы передать данные  $j$ -го узла в общую память, и, во-вторых, время  $\tau_{ji}$ , чтобы забрать данные из общей памяти и передать  $i$ -му узлу, даже в случае выполнения  $i$ -го и  $j$ -го узлов на одном процессоре.

В МВС с распределенной памятью каждый процессор имеет локальную память, в которой могут храниться промежуточные результаты. При этом, если узлы  $j$  и  $i$  обрабатываются одним процессором, то время обмена между ними считается равным 0 (то есть  $T_{ij} = 0$ ), если разными, то время обмена между узлами равно  $\tau_{ji}$ . Такое допущение объясняется тем фактом, что время передачи данных из процессора в свою оперативную память существенно меньше, нежели время пересылки данных в память удаленного процессора.

Выполнение этих дополнительных условий должно учитываться при решении задачи назначения.

Рассмотрим пример поиска критического пути для графа задачи без учета времен передач, представленного на рис. 1. Цифра внутри каждого кружка (узел графа) - номер узла, цифра над кружком - время выполнения узла  $t_j$ . Дуги графа не взвешены, следовательно,  $T_{ji} = T_{ir} = 0$ .

*Основным допущением при поиске КП на графе является следующее: граф задачи реализуется на системе с неограниченными ресурсами, в данном случае на МВС с неограниченным числом процессоров.*

При движении по графу назад от конечной вершины к первой, справа от узла записывается максимально возможное время начала выполнения этого узла  $T_{\max i}$ . Например, для узла 2 по формуле (2) находим:

$$T_{\max 5} - t_2 = 11 - 3 = 8;$$

$$T_{\max 6} - t_2 = 7 - 3 = 4.$$

$$T_{\max 12} - t_2 = 12 - 3 = 9.$$

$$\text{Следовательно, } T_{\max 2} = \min (T_{\max 5}, T_{\max 6}, T_{\max 12}) = 4.$$



Таким образом, для графа, изображенного на рис. 2, имеется один критический путь **1→2→6→9→11→12**, содержащий узлы, удовлетворяющие условию  $T_{\max i} = T_{\min i}$ , причем для каждой пары узлов соотношения (1) и (2) определяются соседней вершиной в паре.

Пути **1→2→9→11→12** и **1→4→10→11→12** не являются критическими, в частности, потому что для пары узлов 2 и 9 соотношения (1) и (2) не определяются соседним в паре узлом. Например,  $T_{\min 9}$  определяется не узлом 2, а узлом 6, аналогично и  $T_{\max 2}$  не определяется узлом 9, т.е.:

$$T_{\min 9} \neq T_{\min 2} + t_2 = 4 + 3 = 7;$$

$$T_{\max 2} \neq T_{\max 9} - t_2 = 12 - 3 = 9.$$

Следовательно, для данной прикладной задачи  $T_{\min} = 23$ ,  $T_{\max} = 48$ .

Примем в качестве стратегии назначения - **выбор готового к исполнению узла ВП с максимальным временем выполнения**. Временная диаграмма выполнения задачи, соответствующая данной стратегии, представлена на рис. 2. В случае нескольких свободных процессоров назначение готового к исполнению узла осуществляется на процессор с меньшим номером. По вертикальной оси - 1,2,3 - номера процессоров, по горизонтальной - время, заданное в условных единицах. В интервалах занятости процессоров проставлены номера узлов задачи.

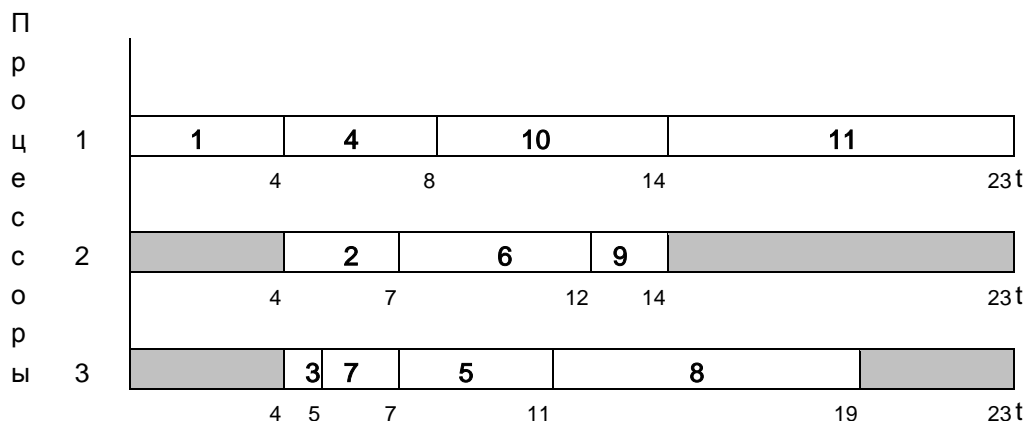


Рис. 2. Диаграмма выполнения задачи без учета времени передач

Анализируя временную диаграмму, можно сделать следующие выводы:

- Минимальное число процессоров, с помощью которых задача выполняется за  $T_{\min}=23$ , составляет  $n_{\min}=3$ , один из процессоров (первый) работает без простоев, два других заняты с 5 по 14 такт и с 5 по 19 такт соответственно.
- Коэффициенты загрузки процессоров соответственно равны  $K_{31} = 1$ ,  $K_{32} = 10/23=0,43$ ,  $K_{33} = 15/23=0,65$ .
- Коэффициент ускорения (максимально возможный для данной задачи)

$$K_y = T_{\max}/T_{\min} = 48/23 = 2,09.$$

- Заштрихованные интервалы времени ("пустоты") для второго и третьего процессоров можно использовать для выполнения копий узлов ВП, готовых к исполнению на начало интервала простоя процессора. Следует обратить внимание на то, что при заполнении "пустот", с точки зрения контроля всех процессоров, целесообразно, чтобы копии одинаковых узлов выполнялись на разных процессорах. Соответствующая диаграмма представлена на рис. 3.

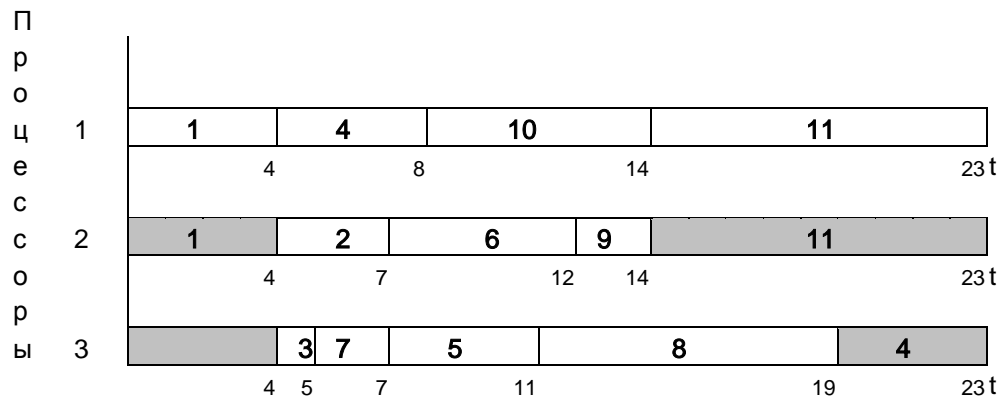


Рис. 3. Пример выполнения копий узлов в "пустотах" временной диаграммы

На основании результатов ее анализа определим вероятность обнаружения ошибки

$$P_{\text{ош}} = (4+9+4) / 48 = 17/48=0,35$$

Перед началом каждой лабораторной работы необходимо подготовить исходные данные для ее выполнения.

В работах 2 и 3 исследуются **графы задач №1** (число вершин 20-23) и **графы задач №2** (число вершин 10-13). Варианты графов задач представлены в соответствующих файлах: *Графы №1\_var\_30\_2021.pdf* и *Графы №2\_var\_30\_2021.pdf*

При выполнении лабораторных работ используются программные модели: *Lab\_Work2\_однозадач режим.exe*, *Lab\_Work3\_многозадач режим.exe* и др.

При запуске программ на экране появляется стандартное меню, включающее

1. блок ввода исходных данных,
2. блок моделирования и
3. блок просмотра результатов.

При вводе исходных данных необходимо последовательно ввести следующие данные:

- режим работы МВС (*однозадачный* или *многозадачный*);
- граф задачи (или графы задач), который вводится из файла данных;
- характеристики системы: *количество процессоров*,

*шин, модулей памяти;  
стратегию выбора готовых узлов.*

В блоке просмотра результатов приводятся временные характеристики вычислительного процесса (ВП), распределение узлов графа по процессорам, временные диаграммы выполнения ВП, коэффициенты загрузки процессоров, шин и т.д.

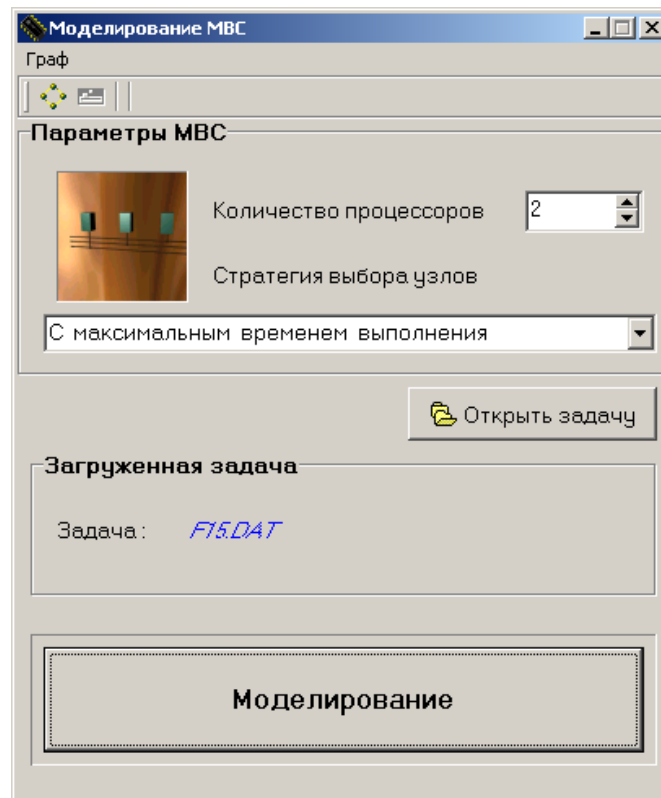


Рис.4. Окно моделирующей программы для лабораторной работы 2

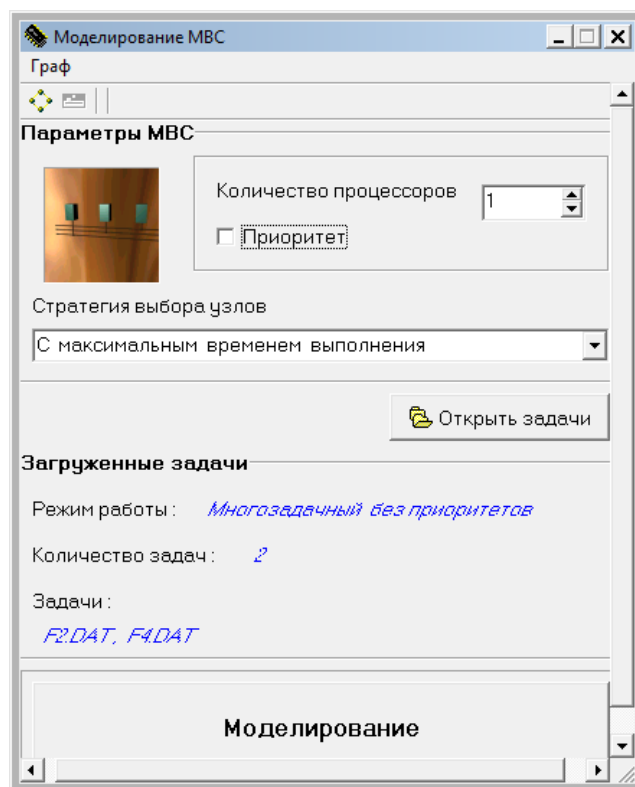


Рис.5. Окно моделирующей программы для лабораторной работы 3

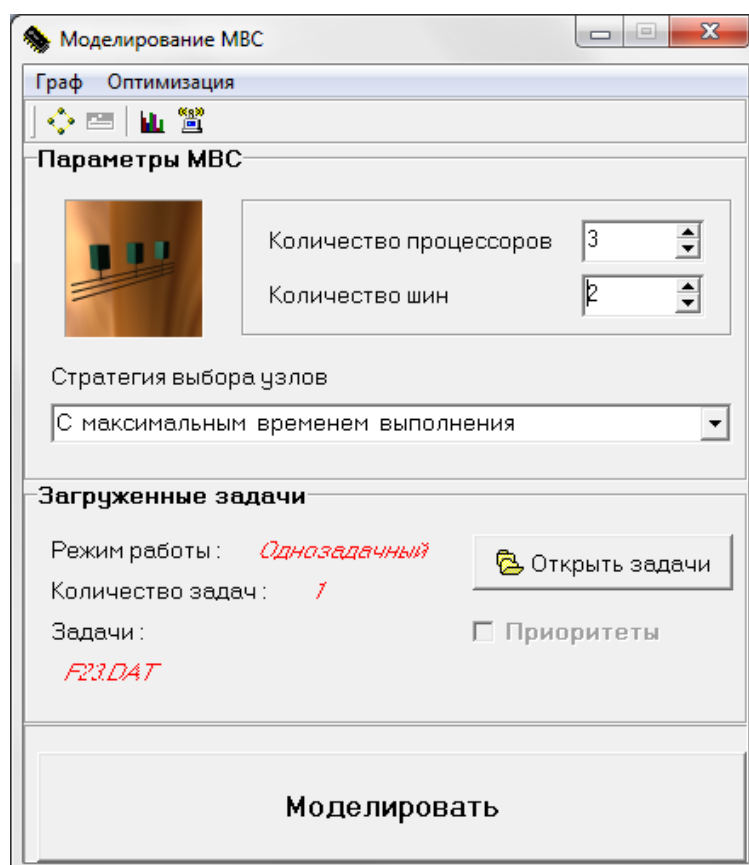


Рис.6. Окно моделирующей программы для лабораторных работ 4 и 5

Полученные результаты моделирования используются для исследований при выполнении лабораторных работ.

После анализа результатов необходимо **изобразить структуру МВС**, в которой наиболее эффективно решается анализируемая задача.

## Постановка задачи назначения

Прежде чем рассматривать существо задачи назначения, отметим, что задача назначения может решаться как в **статическом**, так и в **динамическом режиме**.

В первом случае задача назначения выполняется до начала реализации ВП в МВС (именно этот случай исследуется в данном цикле лабораторных работ), во втором она выполняется непосредственно в процессе реализации ВП.

Теоретической основой решения задачи назначения является теория расписаний.

*Под решением задачи назначения понимается процесс распределения узлов графа задачи (набора задач), выполняемой в МВС, между ее процессорами, при котором определяется время начала выполнения узла, его длительность и назначение процессора, который обеспечит это выполнение.*

Модель процесса распределения включает средства, описывающие ресурсы, систему узлов и дуг графа задачи (набора задач) и критерий оптимальности распределения. Под ресурсами понимаются: модули обработки (процессоры), модули памяти (она может быть распределенной, общей или смешанной), внутрисистемный интерфейс (общая шина, мультиплексная шина). При построении алгоритмов назначения в отказоустойчивых МВС в модель должны быть введены средства, описывающие систему обеспечения отказоустойчивости МВС. Например, средства, обеспечивающие введение дополнительных копий узлов графа задачи и дополнительных процессоров.

Рассмотрим наиболее простой случай.

Пусть в качестве ресурсов в модели используется только набор однотипных процессоров, имеющих равное быстродействие. На данном наборе процессоров выполняется вычислительный процесс, имеющий сетевую структуру и представляющий собой совокупность отдельных алгоритмов (сегментов задачи)  $Z = \{z_i\}$ .

Формально модель выполнения задачи  $Z$  можно представить совокупностью

$$\{Z, <, T, W, \Theta\},$$

где  $Z = \{z_i, z_L\}$  - множество сегментов задачи, выполняемых в системе (узлы графа);

$<$  - означает задание в множестве  $Z$  отношения частичного порядка, которое определяет последовательность выполнения сегментов и информационные связи между ними (связность узлов);

$T = \{t_1, \dots, t_L\}$  - вектор времен выполнения сегментов на процессоре с заданным быстродействием;

$W = \{w_1, \dots, w_L\}$  - вектор коэффициентов важности сегментов, соответствующих коэффициентам относительных потерь эффективности из-за невыполнения сегментов задачи вследствие отказа процессора, на котором выполняется данный сегмент;

$\|\Theta\| = \|\tau_{iq}\|, i = 1 \dots L, q = 1 \dots L-1$  - матрица времен занятости шины с заданной пропускной способностью при передаче данных между узлами  $i$  и  $q$ .

Величина  $\tau_{iq}$  является характеристикой не только структуры графа задачи и пропускной способности шины, но и способа организации МВС.

Как было отмечено выше, в качестве модели задачи используется направленный граф, узлы которого отображают сегменты задачи. При построении алгоритмов, реализующих задачу назначения, направленный граф преобразуется в таблицу связности, элементы которой

$$A_{ij} = \begin{cases} 1, & \text{если выходная информация узла } z_i \text{ является входной} \\ & \text{для узла } z_j \\ 0 & \text{в противном случае.} \end{cases}$$

В качестве критериев оптимальности распределения узлов в МВС применяются обычно либо *минимум времени выполнения задачи (набора задач) при ограничении на число процессоров*, либо *минимум числа требуемых процессоров при ограничении на время решения задачи* (набора задач).

В качестве производных от этих критериев используются следующие:

- максимум загрузки каждого процессора ( $K_{\text{загр } i}$ ),
- минимум простоев каждого процессора ( $K_{\text{пр } i}$ ).

$$K_{\text{загр } i} = T_i / T_{\text{вып}},$$

где  $T_i$  – время, в течение которого  $i$ -й процессор занят обработкой задачи;  $T_{\text{вып}}$  - время выполнения задачи;

$$K_{\text{пр } i} = 1 - K_{\text{загр } i}.$$

Следует отметить при этом, что эффективность как алгоритмов распределения узлов задачи, так и выбранной структуры МВС можно оценить с помощью коэффициента ускорения ( $K_{\text{уск}}$ ), показывающего ускорение времени решения задачи на  $n$  процессорах в сравнении с временем решения этой же задачи на одном процессоре

$$K_{\text{уск}} = T_{\text{max}} / T_n,$$

где  $T_{\text{max}}$  – время решения задачи на одном процессоре;  $T_n$  – время решения задачи на  $n$  процессорах.

В теоретическом плане при построении алгоритмов оптимальных расписаний могут быть использованы математические методы, например, динамического программирования. Однако это достаточно сложная проблема, так как необходимо решать задачи большой размерности, при этом они относятся к NP -полным. На практике оптимальные расписания с использованием таких методов построены для простых типов прикладных задач сравнительно небольшой размерности, причем в основном для двухпроцессорных систем. Поэтому обычно применяются методы приближенной оптимизации, в частности, эвристические методы. Эвристические методы построения расписаний характеризуются тем, что они приспособляются как к информационно-логической структуре ВП, так и к структурной организации МВС, и относятся к классу приоритетного распределения. В таких расписаниях узлам задачи присваивается приоритет по тем или иным правилам (стратегии назначения), после чего узлы упорядочиваются в виде линейного списка по убыванию приоритетов. В процессе составления расписания осуществляется назначение узлов на процессоры в соответствии с их приоритетами для их выполнения. Наиболее исследованы и представлены в литературе различными моделями следующие стратегии назначения готовых к выполнению узлов вычислительного процесса:

1. равновероятный выбор;
2. выбор узла с минимальным временем выполнения;
3. выбор узла с максимальным временем выполнения;
4. выбор узла, принадлежащего критическому пути;
5. выбор узла, имеющего наибольшее число связей с последующими узлами;
6. выбор узла в порядке поступления в очередь на исполнение.

В данном цикле лабораторных работ используются стратегии **2-5**, при этом в ряде работ, где исследуются вопросы обработки набора задач, выбор готового к исполнению узла осуществляется **с учетом приоритета задачи**.

При выполнении лабораторных работ постановка задачи организации параллельных вычислений в МВС сводится к реализации следующей целевой функции:

Определить:

- минимальное число процессоров,
- шин связи коммутационной сети,
- модулей памяти,
- способ организации памяти
- тип стратегии назначения,



обеспечивающих выполнение прикладной задачи (набора задач) за заданное время.

Таким образом, при реализации выбранной стратегии назначения узлы ВП  $Z = \{z_i\}$ ,  $i=1 \dots L$  статически распределяются по процессорам МВС так, что каждому из выбранного числа процессоров  $M_j$  сопоставляется некоторое подмножество узлов  $z_j \in Z$ .

Результатом распределения сегментов задач по процессорам является матрица  $\|X\| = \|X_{ij}\|$ , ( $i=1 \dots L$ ,  $j=1 \dots n$ ) и временная диаграмма занятости процессоров и шин, число которых тоже выбрано при выполнении поставленной задачи.