Московский государственный технический университет
гражданской авиации
Факультет прикладной математики и вычислительной техники
Кафедра прикладной математики

Коновалов В.М.

Математические модели и алгоритмы управления информационными системами

ПОСОБИЕ

к выполнению лабораторных работ

для студентов IV курса
специальности 230401
дневного обучения

Москва – 2010

Данное пособие издается в соответствии с рабочей программой учебной дисциплины «Математические модели и алгоритмы управления информационными системами» по Учебному плану для студентов IV курса специальности 230401 дневного обучения, утвержденному в 2007 г.

Рассмотрено и рекомендовано к изданию на заседании кафедры и методического совета 17.11.2009 г.

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ

Выполнение лабораторных работ по дисциплине “Математические модели и алгоритмы управления информационными системами” направлено на приобретение студентами практических навыков в реализации математических моделей, используемых в решении проблем анализа и синтеза объектов информационных технологий.

Реализация математической модели представляет собой программное отображение метода исследования на определенной совокупности исходных данных, обрабатываемых на ЭВМ. В зависимости от характера задачи, ставящейся на лабораторном занятии, студент должен освоить основные методы решения и вычислительные приемы, связанные с прогнозированием поведения системы, процесса или с выбором оптимальных значений системных параметров. Соответственно этому, учебный материал лабораторной работы можно отнести к решению задачи анализа, связанному с расчетом значений характеристик системы, т.е. с оценкой ее поведения, либо к решению оптимизационной задачи, связанной с расчетом (поиском) таких значений структурных параметров системы, которые удовлетворяют используемому системному критерию эффективности.

Выполнение двух лабораторных работ, методическое описание которых является содержанием данного пособия, связано с решением ***задачи анализа*** характеристик вычислительного процесса, как объекта моделирования (лабораторная работа 1), и с решением ***задачи синтеза*** оптимального управления ресурсами информационной системы на основе как точного, так и эвристического методов теории планирования (лабораторная работа 2).

Лабораторная работа 1. МОДЕЛИ ВЫЧИСЛИТЕЛЬНЫХ АЛГОРИТМОВ

Цель работы:

- изучение технологии математического моделирования вычислительных алгоритмов;
- моделирование вычислительного алгоритма для оценки его трудоемкости;
- реализация математической модели на ЭВМ.

Продолжительность выполнения работы - 8 часов.

1.Теоретическая часть.

Вычислительный процесс, реализуемый в информационной системе, представляется в виде алгоритма - правила, определяющего последовательность действий над исходными данными, приводящего к получению искомых результатов. Реализация алгоритма состоит из последовательности периодов обработки информации и обращения к файлам, которые следуют в порядке, определяемом программой. Длительность вышеназванных периодов определяется трудоемкостью работ - количеством операций, выполняемых процессором; количеством данных, передаваемых при обращении к файлам. При этом должно быть оговорено быстродействие устройств системы, используемых для выполнения соответствующих работ. Таким образом, трудоемкость - одна из основных характеристик алгоритма, обусловленная его вычислительной сложностью и определяющая его потребность в ресурсах системы.

Точная оценка трудоемкости алгоритма обычно апостериорна и возможна после получения машинной программы. Потому, в целях прогноза требуемых ресурсов обычно используют априорные приближенные оценки, которые находятся путем анализа задачи, порождающей вычислительный процесс.

Будем под трудоемкостью алгоритма понимать количество вычислительной работы, требуемой для реализации алгоритма. В таком случае оценкой трудоемкости может быть количество операций, выполняемых во время процессорной обработки и вводе/выводе данных в процессе реализации алгоритма. Если при этом известна производительность устройств обработки и ввода/вывода, то можно получить интегрированную временную оценку трудоемкости.

Каждая реализация алгоритма случайна вследствие того, что исходные данные представляют собой, в общем случае, случайную выборку из множества исходных данных, к которым применим алгоритм. Поэтому полная характеристика трудоемкости предполагает описание количества операций, выполняемых за одну реализацию алгоритма, случайными величинами в форме закона распределения числа операций в реализации. Получение таких сведений - чрезвычайно сложный процесс. В практических целях трудоемкость алгоритма обычно оценивают на уровне средних оценок, например, математическими ожиданиями числа выполняемых операций.

Определим следующую совокупность параметров, используемых для определения трудоемкости алгоритма:

θ - среднее количество процессорных операций, выполняемых за одну реализацию алгоритма (при одном прогоне программы);

$N_1,\dots,N_H$ - среднее количество обращений к файлам $F_1,\dots,F_H$, соответственно, за одну реализацию алгоритма;

$\theta_1,\dots,\theta_H$ - среднее количество данных (байтов), передаваемое за одно обращение к файлам $F_1,\dots,F_H$, соответственно.

Значение θ характеризует трудоемкость обработки данных (счета), а значения $N_1,\dots,N_H$, $\theta_1,\dots,\theta_H$ - трудоемкость процесса ввода/вывода данных.

Математическая модель вычислительного процесса должна отражать его свойства, существенные, в данном случае, для определения трудоемкости.

Будем рассматривать вычислительный процесс как последовательность этапов счета и ввода/вывода данных при обращении к файлам $F_1,\dots,F_H$. Состояние процесса, соответствующее этапу счета, обозначим S_0, а состояния, соответствующие обращениям к файлам $F_1,...,F_H$ - символами $S_1,...,S_H$. Кроме того, окончание вычислительного процесса рассматривается как переход процесса в состояние S_{H+1}, поглощающее вычислительный процесс.

Переходя во временное пространство, вычислительный процесс можно описать последовательностью состояний $S_{t_0}, S_{t_1},...,S_{t_M}$, в которые процесс попадает в моменты времени $t_0, t_1,...,t_M$, причем $S_{t_i} \in \{S_0, S_1,...,S_H\}$, $i = \overline{0, M-1}$; а $S_{t_M} = S_{H+1}$.

Марковская модель вычислительного процесса

Принимая допущение об отсутствии последействия, получают наиболее простую модель вычислительного процесса - марковскую модель, описывающую его как марковский случайный процесс, определяемый множеством состояний $\{S_0,...,S_{H+1}\}$, матрицей вероятностей переходов P = $\left[p_{ij}\right]$, i,j = $\overline{0,H+1}$, и распределением вероятностей $(a_0,...,a_{H+1})$ состояний $S_1,...,S_H,S_{H+1}$ в момент времени t = 0.

Элементы p_{ij} матрицы P определяют вероятности перехода процесса из состояния S_i в состояние S_j. Матрица P - стохастическая, для которой $\sum_j p_{ij} = 1$.

Вероятности a_j определяют первое возможное состояние процесса - S_{t_0}.

Примем следующую концептуальную модель вычислительного процесса: процесс начинается с состояния S_0, т.е. программа начинает выполняться с этапа счета. Этап ввода/вывода может быть инициирован только процессором, т.е. следовать только за этапом счета. Это одновременно означает, что после

каждого этапа ввода/вывода следует этап счета. В таком случае вероятности начальных состояний и матрица вероятностей переходов примут вид:

$$(a_0, a_1, a_2, \ldots, a_{H+1}) = (1\ ,\ 0\ ,\ 0\ ,\ \ldots\ ,\ 0\ )\ ;$$

$$P = \| p_{ij} \| = \begin{vmatrix} 0 & p_{0,1} & p_{0,2} & \cdots & p_{0,H} & p_{0,H+1} \\ 1 & 0 & 0 & \ldots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 \end{vmatrix}\ , \quad i, j = \overline{0,H+1}\ .$$

Значения вероятностей $p_{0,1}, \ldots, p_{0,H+1}$ предопределяют ход вычислительного процесса и зависят от параметров трудоемкости алгоритма. Эти значения находятся из следующих рассуждений.

Трудоемкость алгоритма определяется, в частности, средним числом $N_1, \ldots, N_H$ обращений к файлам $F_1, \ldots, F_H$. Тогда среднее число переходов из состояния S_0 в состояния $S_1, \ldots, S_H$ должно быть $N_1 + N_2 + \cdots + N_H$. Один раз процесс переходит из состояния S_0 в поглощающее состояние S_{H+1}. Поэтому среднее число этапов счета в реализации алгоритма будет равно

$$N = \sum_{h=1}^{H} N_h + 1\ .$$

Значение $p_{0,h}$ определяет долю переходов в состояние S_h по отношению ко всевозможным переходам из состояния S_0 в состояния $S_1, \ldots, S_H, S_{H+1}$. Следовательно,

$$p_{0,h} = \frac{N_h}{N} \quad \text{и} \quad p_{0,H+1} = \frac{1}{N}.$$

Трудоемкость каждого этапа будем рассматривать как случайную величину T_h с математическим ожиданием θ_h , $h = 0\ ,\ 1\ ,\ \ldots\ ,\ H$. В частности, <u>средняя трудоемкость этапа счета</u> есть величина

$$\theta_0 = \frac{\theta}{N}\ ,$$

где θ - среднее количество процессорных операций, выполняемых за одну реализацию алгоритма.

Таким образом, моделью вычислительного процесса является марковская цепь с (H+2) состояниями, начальным состоянием S_0 и матрицей вероятностей переходов - P. Реализация вычислительного процесса - случайная последовательность состояний $S_{t_0}, S_{t_1}, S_{t_2}, ..., S_{t_M}$, изменение которых происходит в соответствии с матрицей вероятностей переходов.

С состояниями $S_0, S_1, ..., S_H$ связано определенное количество работы $\theta_0, \theta_1, ..., \theta_H$, характеризуемое значениями случайных величин $T_0, T_1, \ldots, T_H$, соответственно. Значение $T_h^{(j)}$ можно рассматривать как j -ое значение случайной величины T_h (h = 0 , ... , H), математическим ожиданием которой является θ_h.

Оценка трудоемкости алгоритмов методом теории марковских цепей

Все операторы алгоритма подразделяются на функциональные, перехода, ввода/вывода.

Функциональный оператор - задает совокупность вычислительных операций.

Оператор перехода - задает правило выбора одного из возможных путей развития вычислительного процесса, соответствующего текущим значениям данных, отношения между которыми представляются предикатами.

Оператор ввода/вывода - задает обращение к определенному файлу с целью передачи некоторого количества данных.

Первые два типа операторов задают совокупность вычислительных операций над данными и относятся к классу операторов, называемых основными.

Совокупность операторов алгоритма и связей между ними наглядно представляются графом алгоритма, вершины которого соответствуют операторам алгоритма, а дуги отображают связи между операторами. Среди вершин графа выделяют начальную, конечную и операторные.

Граф алгоритма является корректным, если:

- имеются только одна начальная и только одна конечная вершины;
- для каждой вершины (кроме начальной) существует по крайней мере один путь, ведущий в эту вершину из начальной;
- для каждой вершины (кроме конечной) существует по крайней мере один путь, ведущий из этой вершины в конечную;
- разные выходы из одной вершины ведут к разным вершинам;
- при любых значениях логических условий (предикатов) существует путь из начальной вершины в конечную, причем любому фиксированному набору значений условий соответствует только один такой путь.

Номера вершин графа обозначим 0,1, ... ,k , где 0 - начальная, k - конечная вершина графа. Номера 1,2, ... ,k-1 идентифицируют операторы алгоритма.

Таким образом, граф алгоритма дает наглядное представление структуры алгоритма, определяя множество операторов $V = \{v_1, \dots, v_{k-1}\}$ и дуг $D = \{(i, j)\}$; $i = 0,\dots,k-1$; $j = 1,\dots,k$, связывающих операторы.

Для оценки трудоемкости алгоритма обозначим множество основных операторов: $S_0 = \{v_{\alpha_1}, \dots, v_{\alpha_{m_0}}\}$, где $\alpha \in \{1,2,\dots,k-1\}$; $m_0 \in I = \overline{1,k-1}$.

Множество операторов ввода/вывода: $S_h = \{v_{\beta_1}, \dots, v_{\beta_{m_h}}\}$, где $\beta \in \{1,2,\dots,k-1\}$; $m_h \in I = \overline{1,H}$. Каждый из операторов $v_{\beta_{m_h}}$ задает обращение к файлу F_h.

Обозначим k_α - среднее количество операций, составляющих оператор v_α;

ℓ_β - среднее количество данных, передаваемых при выполнении оператора v_β.

Переходы между операторами v_i и v_j рассматриваем как случайные события и характеризуем вероятностями p_{ij}, т.е. каждая дуга (i, j) графа алгоритма помечается числом p_{ij}.

Так как вычислительный процесс не может приостановиться в вершине v_i, то с вероятностью, равной 1, произойдет переход к какой-либо вершине графа алгоритма. Поэтому вероятности переходов должны отвечать условию:

$$\sum_{j=1}^{k} p_{ij} = 1, \qquad i = 0,1, \dots ,k\text{-}1 .$$

Если за оператором i непременно выполняется оператор j, то p_{ij}=1.

Пусть $n_1, \dots, n_{k-1}$ - среднее число обращений к операторам $v_1, \dots, v_{k-1}$ за один прогон алгоритма. Тогда трудоемкость алгоритма характеризуется следующим набором соотношений.

- Среднее число процессорных операций, выполняемых при одном прогоне алгоритма (трудоемкость алгоритма по основным операторам):

$$\theta_{\text{OCH}} = \sum_{v_i \in S_0} n_i \cdot k_i \ . \tag{1}$$

- Среднее число обращений к файлу F_h при одном прогоне алгоритма:

$$N_h = \sum_{v_i \in S_h} n_i , \qquad h = 1, \dots ,H . \tag{2}$$

- Среднее количество данных, передаваемое при одном обращении к файлу F_h:

$$\theta_h = \frac{1}{N_h} \sum_{v_i \in S_h} n_i \cdot \ell_i \ , \qquad h =1, \dots ,H . \tag{3}$$

- Среднее количество данных, передаваемых при обращениях к файлу F_h за один прогон алгоритма (трудоемкость алгоритма по h - вводу/выводу):

$$\theta^{(h)}_{B/B} = \theta_h \cdot N_h = \sum_{v_i \in S_h} n_i \cdot \ell_i \ , \qquad h =1, \dots ,H . \tag{4}$$

- Среднее количество данных, передаваемых при обращениях к файлам за один прогон алгоритма (трудоемкость алгоритма по вводу/выводу):

$$\theta_{В/В} = \sum_h \theta_{В/В}^{(h)} , \qquad h = 1, \ldots, H . \qquad (5)$$

В приведенных выражениях суммирование выполняется по всем вершинам графа, относящимся к классу основных операторов - S_0 или к классу операторов ввода/вывода - S_h, обращающихся к файлу F_h.

Таким образом, для оценки трудоемкости алгоритма необходимо определить среднее число обращений $n_1, \ldots, n_{k-1}$ к операторам. Для этого примем ряд допущений, идеализирующих модель.

Будем считать, что p_{ij} постоянны и после выполнения оператора v_i ($i = 1,...,k-1$) переход к следующему оператору определяется только распределением вероятностей p_{ij}, т.е. не зависит от хода вычислительного процесса в прошлом - до перехода к оператору v_i. В таком случае процесс выполнения алгоритма можно считать марковским процессом с k состояниями $S_1,...,S_k$, соответствующими пребыванию процесса в вершинах $v_1,...,v_k$ графа алгоритма.

Состояния $S_1,...,S_{k-1}$ - невозвратные. Состояние S_k - поглощающее. Начальным является состояние S_i, определенное дугой (0, i), выходящей из вершины v_0 графа. Для упрощения обозначений примем i = 1, т.е. начальным состоянием процесса является состояние S_1, соответствующее вершине v_1 (всегда можно перенумеровать вершины графа так, чтобы дуга (0, i) стала дугой (0, 1)). В результате принятых условий, граф алгоритма можно рассматривать в качестве графа марковской цепи.

Среднее число $n_1, \ldots, n_{k-1}$ пребываний марковского процесса в невозвратных состояниях $S_1,...,S_{k-1}$ (т.е. искомое среднее число $n_1, \ldots, n_{k-1}$

обращений к операторам $v_1,...,v_{k-1}$) определяется корнями системы линейных алгебраических уравнений:

$$n_i = \delta_{1i} + \sum_{j=1}^{k-1} p_{ji} \cdot n_j \ , \qquad i = 1, ... ,k\text{-}1 .$$

Здесь δ_{1i} - символ Кронекера : $\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}.$

К вышеприведенной записи приводят следующие рассуждения. Значение n_i будет равно, по крайней мере, 1, если процесс начинается из состояния с номером i = 1, что определяется значением $\delta_{11} = 1$. В остальных случаях процесс попадает в состояние S_i только из какого-либо другого состояния S_j, j = 1,...,k-1 с вероятностью p_{ji}.

Если процесс находится в состоянии j n_j раз и $p_{ji} \neq 0$, то из этого состояния он попадает в состояние i в среднем $p_{ji} \cdot n_j$ раз. Суммированием значений $p_{ji} \cdot n_j$ по всем j находится число попаданий процесса в состояние i из всех других состояний j.

Каноническая запись системы уравнений имеет вид:

$$\begin{cases} n_1 = 1 & +p_{11}n_1 & +p_{21}n_2 & +\dots & +p_{k-1,1}n_{k-1} \\ n_2 = & p_{12}n_1 & +p_{22}n_2 & +\dots & +p_{k-1,2}n_{k-1} \\ \dots & & \dots & & \\ n_{k-1} = & p_{1,k-1}n_1 & +p_{2,k-1}n_2 & +\dots & +p_{k-1,k-1}n_{k-1} \end{cases}.$$

После преобразований получим:

$$\begin{cases} (p_{11}-1)n_1 + p_{21}n_2 + \ldots + p_{k-1,1}n_{k-1} = -1 \\ p_{12}n_1 + (p_{22}-1)n_2 + \ldots + p_{k-1,2}n_{k-1} = 0 \\ \ldots \\ p_{1,k-1}n_1 + p_{2,k-1}n_2 + \ldots + (p_{k-1,k-1}-1)n_{k-1} = 0 \end{cases}.$$

Пример оценки трудоемкости алгоритма по методу марковских цепей

Пусть для некоторой конкретной реализации граф алгоритма имеет вид:

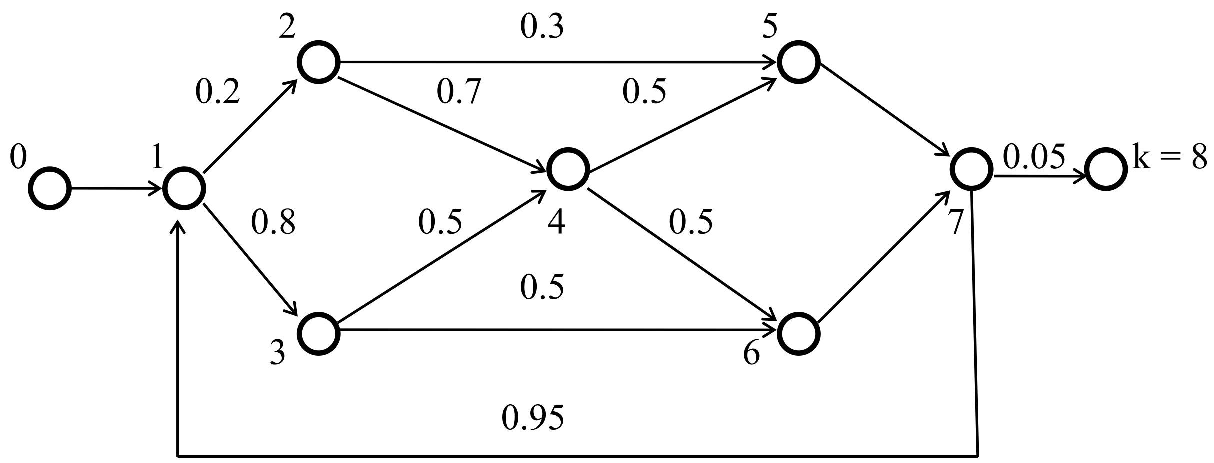

Рис.1.

Имеем систему из семи линейных алгебраических уравнений:

$$\begin{cases} -n_1 & & & & & & +0.95n_7 & = -1 \\ 0.2n_1 & -n_2 & & & & & & = 0 \\ 0.8n_1 & & -n_3 & & & & & = 0 \\ & 0.7n_2 & +0.5n_3 & -n_4 & & & & = 0 \\ & 0.3n_2 & & +0.5n_4 & -n_5 & & & = 0 \\ & & 0.5n_3 & +0.5n_4 & & -n_6 & & = 0 \\ & & & & n_5 & +n_6 & -n_7 & = 0 \end{cases}.$$

Решение системы уравнений определяет среднее число попаданий вычислительного процесса в состояния $S_1, \ldots, S_7$:

$n_1 = 20, \quad n_3 = 16, \quad n_5 = 6.6, \quad n_7 = 20.$

$n_2 = 4, \quad n_4 = 10.8, \quad n_6 = 13.4,$

• Пусть <u>все</u> операторы алгоритма - <u>основные</u>, а количество операций - k_i, порождаемых оператором v_i, постоянно и равно 1. Тогда трудоемкость алгоритма будет равна $\theta_{OCH} = \sum_{i=1}^{7} k_i n_i = 20 + 4 + ... + 20 = 90.8$ операций.

• • Если, к примеру, <u>операторы</u> v_4 и v_7 являются операторами <u>ввода/вывода</u>, а количество операций, порождаемых каждым из основных операторов, равно $k_1 = 500$, $k_2 = 500$, $k_3 = 50$, $k_5 = 300$, $k_6 = 20$, соответственно, то <u>трудоемкость по основным операторам</u> составит величину

$\theta_{OCH} =$ 500*20 + 500*4 + 50*16 + 300*6.6 + 20*13.4 = 15 048 операций.

Пусть при обращении из оператора v_4 к файлу F_4 передается $\ell_4 = 1800$ байт данных, а при обращении из оператора v_7 к файлу F_7 передается $\ell_7 =$ 2540 байт. Тогда <u>трудоемкость по вводу/выводу</u> определяется, как $\theta_{B/B} = \sum_{4,7} \theta_{B/B}^{(h)} =$ $\theta_{B/B}^{(4)} + \theta_{B/B}^{(7)}$. В свою очередь: $\theta_{B/B}^{(4)} = \sum_{v_4 \in S_4} n_i \ell_i = n_4 \ell_4 = 10.8*1800$ = 19440 байт. Аналогично, $\theta_{B/B}^{(7)} = n_7 \ell_7 = 20*2540$ = 50800 байт. Тогда $\theta_{B/B} =$ 19440 + 50800 = 70240 байт.

<u>Интегрированная оценка трудоемкости</u> в этом случае составляет величину:

$$\theta = \theta_{OCH} / B_{OCH} + \theta_{B/B} / B_{B/B} \quad [\text{ед.времени}] . \qquad (6)$$

Здесь B_{OCH} - быстродействие процессора по основным операциям; $B_{B/B}$ - быстродействие соответствующих устройств ввода/вывода.

• • • Если операторы v_4 и v_7 обращаются к одному и тому же файлу F_{47}, то в этом случае <u>трудоемкость по вводу/выводу</u> определяется, как $\theta_{B/B} = \theta_{B/B}^{(47)}$.

Сразу находим: $\theta_{B/B}^{(47)} = \sum_{v_4, v_7 \in S_{47}} n_i \ell_i = n_4 \ell_4 + n_7 \ell_7 = 19440 + 50800$ = 70240 байт. Как видно, результат не отличается от предыдущего.

Оценка трудоемкости алгоритмов сетевым методом

Количество вычислений, проводимых при расчете трудоемкости алгоритмов, можно значительно сократить, если использовать сетевой подход к анализу трудоемкости. Однако, он применим к графам алгоритмов, несодержащих циклы. Поэтому, вначале рассматривается граф алгоритма без циклов, после чего метод обобщается на алгоритм с циклами. Это производится путем преобразования графа алгоритма с циклами в граф с эквивалентной трудоемкостью, но без циклов.

Для применения сетевого метода к оценке трудоемкости алгоритма вершины графа должны быть перенумерованы в порядке их следования: любая вершина должна иметь номер, больший любого номера предшествующих ей вершин. Нумерация вершин производится следующим образом.

Начальной вершине присваивается номер 0. Очередной номер i = 1 ,2 , ... присваивается вершине, в которую входят дуги от уже пронумерованных вершин с номерами, меньшими i. При этом, любым двум вершинам должны соответствовать разные номера. Такой порядок нумерации является результативным для любого графа без циклов.

Поскольку граф не содержит циклов, то $p_{ji} = 0$ для всех $j \geq i$. Тогда, при прогоне алгоритма, вершина 1 будет выполнена точно один раз, т.е. $n_1 = 1$.

Среднее число попаданий вычислительного процесса в вершину i будет равно:

$$n_i = \sum_{j=1}^{k-1} p_{ji} n_j \ , \quad i = 2,3, \dots ,k\text{-}1 \ .$$

При описанном выше порядке нумерации вершин графа, на момент вычисления n_i значения $n_1, \dots, n_{i-1}$ будут уже определены. Очевидно, суммирование следует проводить только для $j < i$, поскольку $p_{ji} = 0$ для $j \geq i$.

В качестве примера рассмотрим граф на рис.1, но исключим из рассмотрения дугу (7,1). В таком случае имеем граф без циклов, удовлетворяющий порядку нумерации вершин. Среднее число обращений $n_1, n_2, ..., n_7$ к операторам алгоритма будет равно:

$n_1 = 1;$ $n_2 = p_{12}n_1 = 0.2*1 = 0.2;$ $n_3 = p_{13}n_1 = 0.8*1 = 0.8;$

$n_4 = p_{24}n_2 + p_{34}n_3 = 0.7*0.2 + 0.5*0.8 = 0.54;$

$n_5 = p_{25}n_2 + p_{45}n_4 = 0.3*0.2 + 0.5*0.54 = 0.33;$

$n_6 = p_{36}n_3 + p_{46}n_4 = 0.5*0.8 + 0.5*0.54 = 0.67;$

$n_7 = n_5 + n_6 = 0.33 + 0.67 = 1.$

Теперь рассмотрим случай алгоритма, содержащего циклы. Задача состоит в том, чтобы исключить циклы, заменив их операторами с эквивалентной трудоемкостью.

Все циклы делятся на ранги. К рангу 1 относятся циклы, несодержащие внутри себя ни одного цикла, к рангу 2 - циклы, содержащие в себе циклы ранга 1 и т.д. Для графа на рис.1 с учетом дуги (7,1) имеем цикл ранга 1.

Совокупность операторов, входящих в цикл, и связывающих их дуг, за исключением дуги, замыкающей цикл, называют <u>телом цикла</u>. Тело цикла ранга 1 является графом без циклов. Применяя к этому графу вышеописанную методику, можно определить значения n_i для операторов тела цикла. Тогда трудоемкость тела цикла C может быть определена как $\sum_{v_j \in C} k_j n_j$, где k_j - трудоемкость j-ого оператора тела цикла; суммирование ведется по всем вершинам v_j, содержащимся в цикле C.

Обозначим n_C - <u>среднее число повторений цикла</u>, равное числу выполнений тела цикла при одном прогоне алгоритма. Если вероятность перехода по дуге, замыкающей цикл, равна p_{kl}, тогда

$$n_C = 1 + p_{kl} n_C .$$

В этом выражении второе слагаемое представляет собой среднее число повторных выполнений тела цикла, которое можно трактовать как долю от общего числа выполнений цикла - n_C, обусловленную вероятностью p_{kl}.

Отсюда получим, что

$$n_C = \frac{1}{1 - p_{kl}} \quad . \tag{7}$$

Тогда средняя трудоемкость цикла равна

$$k_C = n_C \sum_{v_j \in C} k_j n_j \, . \tag{8}$$

Теперь цикл С можно заменить оператором, имеющим трудоемкость k_C. Если граф содержит циклы, ранг которых выше 1, то последовательное применение процедуры эквивалентной замены приводит к графу без циклов, трудоемкость которого находится вышеописанным способом.

Отметим, что структура формулы (8) является обобщенной в том смысле, что дает возможность рассчитывать трудоемкость как по основным операторам тела цикла, так и по операторам ввода/вывода, используя соотношения (1), (4) и (5) в “циклической” форме:

$$k_{C\ \text{OCH}} = n_C \sum_{v_i \in S_0} n_i \cdot k_i \quad ; \tag{1a}$$

$$\theta_{C\ \ \text{B/B}}^{(h)} = n_C \sum_{v_i \in S_h} n_i \cdot \ell_i \quad ; \tag{4a}$$

$$\theta_{C\ \ \text{B/B}} = \sum_h \theta_{C\ \ \text{B/B}}^{(h)} \quad . \tag{5a}$$

Для рассмотренного выше примера, используя средние числа $n_1, \ldots, n_7$ обращений к операторам, полученные на основе сетевого метода, находим:

- $k_{C\ \ \text{OCH}} = \frac{1}{1 - 0.95} \sum_{i=1}^{7} n_i k_i = 20(1 + 0.2 + 0.8 + 0.54 + 0.33 + 0.67 + 1) = 90.8$ опер.

• • $k_{C\ \ OCH} = \frac{1}{1-0.95}\sum_{v_i \in S_0} n_i k_i = 20(1*500\ +\ 0.2*500\ +\ 0.8*50\ +\ 0.33*300\ +$ $0.67*20) = 15048$ опер.

$$\theta_{C\ \ B/B}^{(4)} = \frac{1}{1-0.95}\sum_{v_4 \in S_4} n_i \cdot \ell_i = 20*0.54*1800 = 19440 \text{ байт.}$$

$$\theta_{C\ \ B/B}^{(7)} = \frac{1}{1-0.95}\sum_{v_7 \in S_7} n_i \cdot \ell_i = 20*1*2540 = 50800 \text{ байт.}$$

$$\theta_{C\ \ B/B} = \sum_{4,7} \theta_{C\ \ B/B}^{(h)} = 19440 + 50800 = 70240 \text{ байт.}$$

• • • $\theta_{C\ \ B/B} = \theta_{C\ \ B/B}^{(47)} = \frac{1}{1-0.95}\sum_{v_{4,}v_7 \in S_{47}} n_i \ell_i = 20(0.54*1800 + 1*2540) = 70240$ байт.

Таким образом, трудоемкость цикла в данном примере одновременно определяет трудоемкость алгоритма, поскольку граф алгоритма, по сути, является телом единственного цикла. Сравнивая полученные результаты с соответствующими результатами, полученными при использовании метода теории марковских цепей, видно, что результаты тождественны.

Литература.

1. Основы теории вычислительных систем. Под ред. С.А. Майорова. - М.: Высшая школа, 1987, с. 34-47.

2. Порядок выполнения лабораторной работы 1.

• Ознакомиться с содержанием лабораторной работы.

• Построить по таблице 1, в соответствии с вариантом задания, граф алгоритма.

• Построить математическую модель вычислительного процесса для оценки трудоемкости алгоритма по методу теории марковских цепей.

• Построить математическую модель вычислительного процесса для оценки трудоемкости алгоритма сетевым методом.

• Подготовить программу для расчета модельных характеристик трудоемкости на одном из языков высокого уровня (например, на Паскале).

• Отладить программу и получить результаты расчетов.

• Подготовить в объектно-ориентированной среде разработки интерактивную форму для управления работой программы и визуализации полученных результатов.

• Подготовить отчет о выполненной работе. Форма отчета - по стандарту.

3. Варианты заданий на выполнение лабораторной работы 1.

А. Вариант графа алгоритма. Таблица 1.

Вариант	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
P_{12}	0.2	0.1	1	0.5	1	1	0.3	1	0.2	0.3	0.1	1	1	0.2
P_{13}	0.2	0.3		0.5			0.7		0.4	0.7	0.1			0.8
P_{14}	0.6	0.6							0.4		0.8			
P_{23}			0.1		0.3	0.6		0.2				0.2	0.5	
P_{24}			0.3	0.4	0.7	0.4	0.5	0.3		0.7		0.8	0.5	0.3
P_{25}	1	1	0.6	0.6			0.5	0.5	1	0.3	1			07
P_{34}					1							1		
P_{35}	0.1	0.3		0.3		0.1	0.1		0.1	0.6	0.2		0.2	0.2
P_{36}	0.9	0.3	1			0.9	0.2	1	0.2		0.8		0.8	0.2
P_{37}		0.4		0.7			0.7		0.7	0.4				0.6
P_{45}					0.5							0.3		
P_{46}	1		1	1	0.5	0.3	1	1		1	1	0.7	0.2	1
P_{47}		1				0.7			1				0.8	
P_{56}		1	1	1			1	1	1	1				1
P_{57}	1				1	1					1	1	1	
P_{61}			0.9					0.8						
P_{67}	1	1	0.1		1	1	1	0.2	1		1	1	1	1
P_{68}				1						1				
P_{71}	0.9	0.8					0.8		0.8		0.9			0.9
P_{72}					0.8							0.5		

P_{78}				1		1				1			1	
P_{81}				0.9		0.7				0.8			0.6	

Продолжение таблицы 1

Вариант	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**
P_{12}	0.2	0.1	1	0.5	1	1	0.3	1	0.2	0.3	0.1	1	1	0.2
P_{13}	0.2	0.3		0.5			0.7		0.4	0.7	0.1			0.8
P_{14}	0.6	0.6							0.4		0.8			
P_{23}			0.1		0.3	0.6		0.2				0.2	0.5	
P_{24}			0.3	0.4	0.7	0.4	0.5	0.3		0.7		0.8	0.5	0.3
P_{25}	1	1	0.6	0.6			0.5	0.5	1	0.3	1			07
P_{34}					1							1		
P_{35}	0.1	0.3		0.3		0.1	0.1		0.1	0.6	0.2		0.2	0.2
P_{36}	0.9	0.3	1			0.9	0.2	1	0.2		0.8		0.8	0.2
P_{37}		0.4		0.7			0.7		0.7	0.4				0.6
P_{45}					0.5							0.3		
P_{46}	1		1	1	0.5	0.3	1	1		1	1	0.7	0.2	1
P_{47}		1				0.7			1				0.8	
P_{56}		1	1	1			1	1	1	1				1
P_{57}	1				1	1					1	1	1	
P_{61}			0.9				0.1	0.8			0.3			0.6
P_{67}	1	1	0.1		1	1	0.9	0.2	1		0.7	1	1	0.4
P_{68}				1						1				
P_{71}	0.8	0.6	0.7		0.3		0.8		0.8		0.9	0.2		0.9
P_{72}	0.1	0.2		0.3	0.5	0.2		0.7	0.1	0.6		0.5	0.2	
P_{78}				0.7		0.8				0.4			0.8	
P_{81}				0.9		0.7				0.8			0.6	

В. Тип и трудоемкость операторов. Таблица 2.

вариант	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	100	200	120	320	250	300	100	100	200	150	100	100	300	100
2	200	150	120	200	300	200	400	200	300	150	200	120	200	200
3	120	300	150	200	200	300	500	100	200	200	100	200	300	300
4	300	250	200	150	150	800	400	200	100	200	250	300	200	300
5	100	200	300	150	300	300	300	300	250	400	500	300	250	250
6	300	300	600	300	400	200	250	300	250	300	300	150	200	150
7	100	800	300	100	100	200	200	400	300	200	200	100	150	200
8	-	-	-	800	-	900	-	-	-	200	-	-	200	-

Продолжение таблицы 2

вариант	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**
1	200	400	240	640	500	600	200	200	400	300	200	200	600	200
2	400	300	240	400	600	400	800	400	600	300	400	240	400	400
3	240	60	300	400	400	600	900	200	400	400	200	400	600	600
4	600	500	400	300	300	900	800	400	200	400	500	600	400	600
5	200	400	600	300	600	600	600	600	500	800	900	600	500	500
6	600	600	900	600	400	400	500	600	500	600	600	300	400	300
7	200	700	600	200	100	400	400	800	300	400	400	100	150	400
8	-	-	-	900	-	800	-	-	-	700	-	-	600	-

Примечание: Затемненная ячейка в таблице означает, что соответствующий оператор (строка таблицы) является оператором ввода/вывода.

4. Контрольные вопросы.

1. Как определяется трудоемкость вычислительного алгоритма?
2. Дайте определение интегрированной оценки трудоемкости алгоритма.
3. С помощью каких параметров модели вычислительного процесса оценивают трудоемкость алгоритма?
4. Какие упрощения в описании вычислительного процесса используются для построения его математической модели?
5. Что определяет марковскую модель вычислительного процесса?
6. Назовите типы операторов алгоритма, определяемые для оценки его трудоемкости.
7. Какие типы операторов алгоритма относятся к классу основных операторов при оценке трудоемкости алгоритма?
8. Как определяется граф алгоритма?
9. В каком случае граф алгоритма является корректным?
10. Перечислите используемые характеристики трудоемкости алгоритма.
11. В чем сущность сетевого подхода к оценке трудоемкости алгоритма?
12. Дайте определение цикла ранга К.
13. Как определяется трудоемкость алгоритмов, содержащих циклы ранга К, если К>1?
14. Дайте сравнительную оценку метода марковских цепей и сетевого метода для определения трудоемкости алгоритмов.

Лабораторная работа 2. МОДЕЛИ ПРОЦЕССОВ ПЛАНИРОВАНИЯ РАБОТ В ИНФОРМАЦИОННОЙ СИСТЕМЕ

Цель работы:

- ознакомиться с концептуальной и формальной моделью процесса планирования работ в информационной системе;
- определить план запуска заданий на обработку для 2-х и 3-х фазной моделей, используя точный и приближенный (эвристический) методы планирования;
- произвести оценку эффективности эвристического метода планирования.

Продолжительность выполнения работы - 8 часов.

1.Теоретическая часть.

Важнейшими понятиями при изучении процессов планирования являются “работа” и “задание”, которые на уровне формализованной (математической) модели информационной системы воспринимаются вне связи с конкретным назначением системы.

Работа - это объект, состоящий из определенной программы и определенного набора данных. Работа - это также процесс выполнения конкретной программы с целью обработки определенного набора данных. Как объект работа существует, как процесс работа выполняется. Одна программа может использоваться для обработки многих наборов данных и одни и те же данные могут обрабатываться различными программами. Пользуясь термином “работа”, из множества сочетаний “программа + данные” выделяют одно конкретное сочетание.

Задание - это причина создания работы. Задание определяет наименование программы, данных и, возможно, некоторые параметры работы, например, ее приоритет. Задания формируются вне системы и поступают на ее входные устройства.

Для того , чтобы выполнить задание, которое система интерпретирует как работу, последней требуется выделить совокупность ресурсов - необходимый объем ОЗУ, ВЗУ, требуемые устройства ввода/вывода, процессорное время и время каналов ввода вывода. Из-за ограниченности ресурсов все задания не могут быть одновременно обеспечены ими. Параллельно во времени может выполняться только часть работ, обеспечиваемых ресурсами системы. Потому работы выполняются в основном последовательно. В каждый момент времени в системе может обрабатываться ограниченное число работ, возможно - одна.

Различный порядок в последовательности выполнения работ приводит к различным эффектам. Например, процессор должен выполнить работы J_1, J_2, J_3 с трудоемкостью 4, 3 и 1 единиц процессорного времени, соответственно. Возможны 6 различных расписаний (планов) запуска работ на выполнение: $P_1 = (J_1, J_2, J_3)$; $P_2 = (J_2, J_1, J_3)$; ... ; $P_6 = (J_3, J_2, J_1)$, различающихся порядком запуска работ. Для планов P_1 и P_6 диаграмма выполнения работ изображена на рис.1.

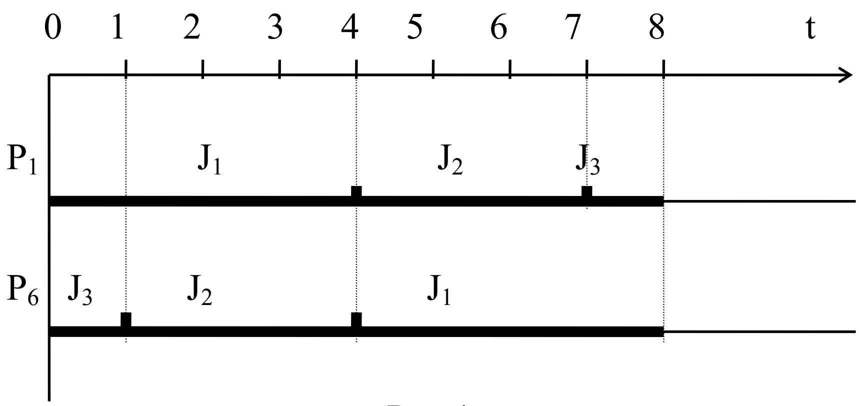

Рис.1.

Сравним планы P_1 и P_6 по величине среднего времени ожидания и среднего времени пребывания работ в системе.

$$w_1 = \frac{0+4+7}{3} = 3\tfrac{2}{3}, \qquad u_1 = \frac{4+7+8}{3} = 6\tfrac{1}{3};$$

$$w_6 = \frac{4+1+0}{3} = 1\tfrac{2}{3}\ , \qquad u_6 = \frac{8+4+1}{3} = 4\tfrac{1}{3}$$

Очевидно, изменение плана выполнения работ существенно повлияли на характеристики процесса обслуживания заданий.

Рассмотрим обслуживание задания в информационной системе несколькими устройствами. Например, работы J_1 и J_2 выполняются вначале на устройстве $У_1$, а затем на устройстве $У_2$, как показано на рис.2.

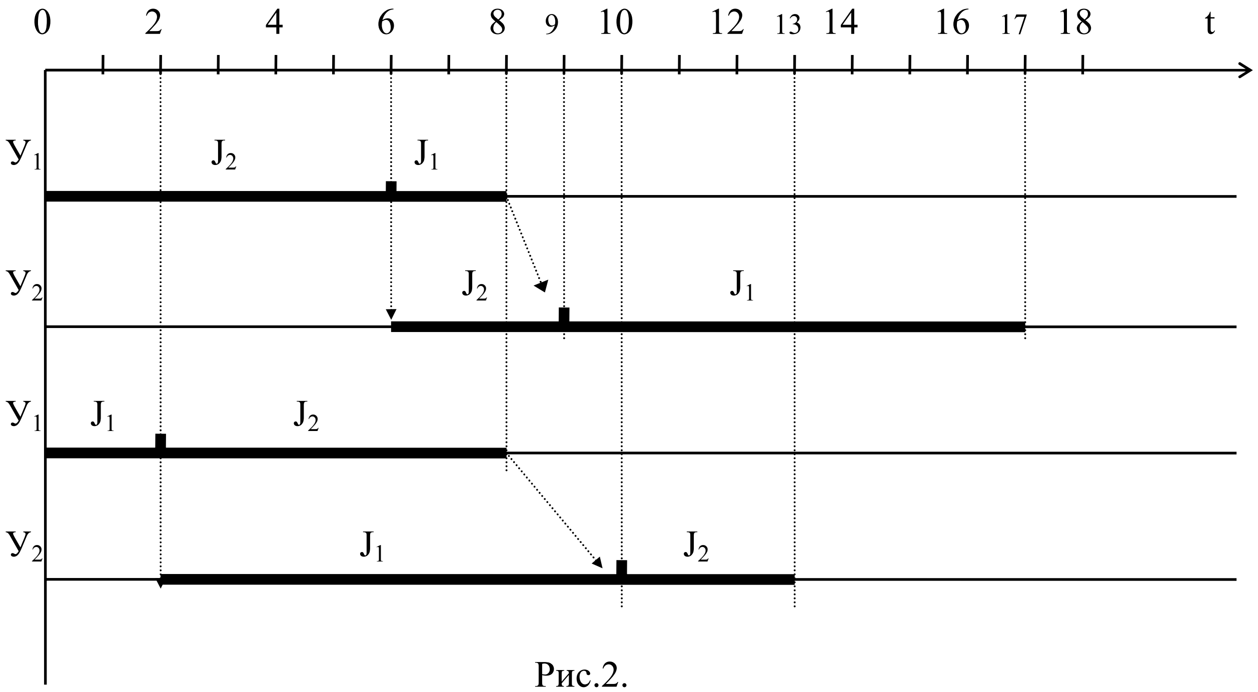

Рис.2.

Трудоемкость работ J1 и J2 равна (2, 8) и (6, 3), соответственно, где первая компонента вектора трудоемкости представляет число единиц времени первого устройства, а вторая - второго устройства для выполнения работы.

Возможны два плана запуска работ: $P_1 = (J_2, J_1)$ и $P_2 = (J_1, J_2)$. Сравнение этих планов можно осуществить по времени выполнения работ в системе. В первом случае оно равно 17 ед., во втором - 13 ед. Следовательно, план P_2 более предпочтителен, так как обеспечивает минимум времени выполнения работ за счет высокой степени совмещения их во времени.

Рассмотренные примеры показывают, что порядок запуска работ на выполнение влияет на характеристики обслуживания пользователей информационной системы. Следовательно, последняя должна иметь встроенные средства, обеспечивающие такой порядок запуска работ, который, к примеру, минимизировал бы время пребывания работ в системе.

Процесс определения порядка выполнения работ во времени (т.е. плана или расписания) называется планированием работ. Его целью является улучшение определенных характеристик системы. Такое улучшение достигается за счет соответствующего распределения работ в пространстве (среди устройств системы) и во времени.

Планирование работ производится, исходя из имеющихся в системе ресурсов, потребностей работ в ресурсах, приоритетов работ и критерия, определяющего качество планов.

Ресурсы информационной системы.

Ресурсы информационной системы - это ресурсы памяти и ресурсы устройств.

Ресурс памяти - это емкость памяти, которая может быть разделена частями между несколькими работами. Ресурс памяти системы слагается из емкости оперативной памяти и емкости внешней памяти.

Ресурс устройства - это количество работы, которую может выполнить устройство, в свою очередь, определяемое производительностью устройства (его быстродействием) и пропорциональное времени, отведенному для работы. Можно говорить, что ресурс устройства - это время.

Устройство в каждый момент времени может выполнять только одну работу. В этом смысле внешнее ЗУ является не только памятью, но и устройством.

В отличие от ВЗУ оперативная память в этом смысле устройством не является, поскольку ее работа во времени организуется схемными средствами и не поддается системному планированию.

Системные средства планирования.

Выделяют два основных этапа выполнения работы: создание работы по заявке; собственно выполнение работы. Совокупность средств, обеспечивающих порядок выполнения работ, иллюстрируется схемой на рис.3.

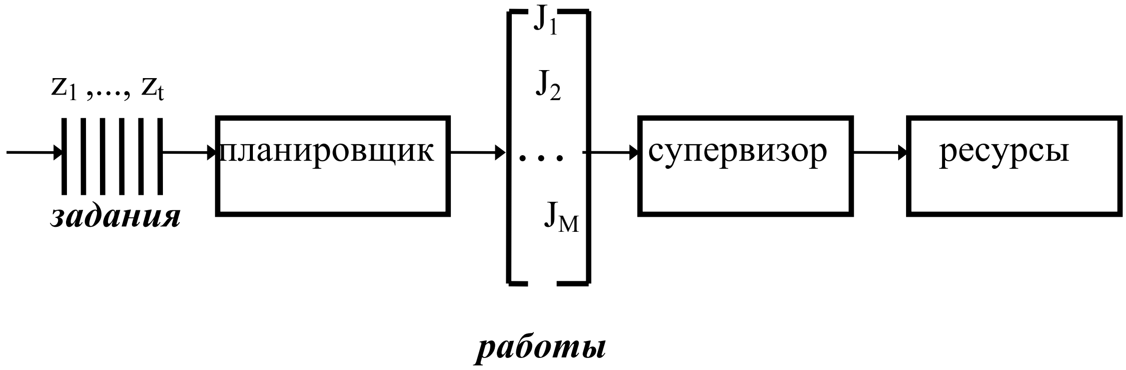

Рис.3.

Ввод заданий Z_1, ... ,Z_t и создание на их основе работ выполняется управляющей программой - планировщиком. Планировщик выполняет общее (стратегическое) планирование, выбирая из массы заданий те, которые следует и можно выполнить в первую очередь. Порядок выполнения заданий может определяться их приоритетом или целью планирования, например, стремлением уменьшить среднее время пребывания работ в системе.

Выполнение работ J_1,J_2, ... ,J_M, созданных планировщиком, сводится к предоставлению этим работам ресурсов системы и протекает под управлением другой системной программы - супервизора. В момент окончания некоторой работы часть ресурсов освобождается и супервизор передает управление планировщику, который пытается создать новую работу.

Супервизор планирует выполнение работ, созданных планировщиком, занимаясь распределением ресурсов, потребность в которых возникает уже в процессе выполнения работ. Сфера планирования, охватываемая супервизором, носит более частный (тактический) характер по сравнению со сферой, контролируемой планировщиком.

Различие в назначении информационных систем предопределяет необходимость в использовании разных стратегий планирования. Однако, в любом случае алгоритмы планирования, реализуемые планировщиком, должны быть достаточно простыми и малотрудоемкими. В противном случае, эффект, достигаемый за счет рационального планирования, понижается из-за больших расходов процессорного времени на реализацию собственно алгоритма планирования.

Постановка задачи планирования.

Задача планирования сводится к нахождению такого порядка выполнения определенного множества работ с использованием заданного набора ресурсов, при котором некоторый критерий эффективности принимает экстремальное значение. Имеем постановку оптимизационной задачи.

Будем считать, что информационная система обладает совокупностью ресурсов $F_1,\ldots,F_i,\ldots,F_N$, где F_i - ресурс памяти или ресурс устройства.

Потребность работ $J_1, \ldots ,J_M$ в ресурсах $F_1, \ldots ,F_N$ характеризуется матрицей:

$$T = \left[\tau_{ij}\right] = \begin{bmatrix} \tau_{11} & \tau_{12} & \cdots & \tau_{1N} \\ \tau_{21} & \tau_{22} & \cdots & \tau_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ \tau_{M1} & \tau_{M2} & \cdots & \tau_{MN} \end{bmatrix}.$$

Элемент τ_{ij} $(i = \overline{1,M};\ j = \overline{1,N})$ характеризует потребность работы J_i в ресурсе F_j. Если работа J_i выполняется без использования ресурса F_j, то $\tau_{ij} = 0$. Если ресурс F_j является памятью, то значение τ_{ij} задает количество единиц данных, которые должны быть размещены в памяти F_j при выполнении работы J_i. Если F_j - устройство, то τ_{ij} определяет число единиц времени, необходимого для выполнения работы J_i на этом устройстве. Матрицу T называют матрицей трудоемкости работ.

Критерий эффективности плана дает возможность количественной оценки его качества, устанавливая способ измерения последнего. Критерий эффективности выводится из критерия эффективности системы в целом и поэтому отображает степень соответствия плана назначению системы.

Для оценки качества планирования используются различные критерии. В целях данной лабораторной работы будем использовать один из них, который называется критерием минимума суммарного времени выполнения работ. Данный критерий используется в многозадачных системах, для которых характерно стремление достичь максимальной производительности на имеющихся ресурсах, что равносильно обработке пакета заданий за минимальное время. Исходя из этого, эффективность планирования в таких системах следует оценивать суммарным временем выполнения работ, равным промежутку времени от момента загрузки пакета заданий в систему до момента завершения последней работы.

Можно предположить, что суммарное время обработки пакета будет тем лучше, чем больше загружены во времени устройства системы. В таком случае, величина $R = \sum_{i=1}^{N} \rho_i$, равная сумме коэффициентов загрузки отдельных устройств, может служить приемлемой оценкой качества плана. Тогда план считается оптимальным, если суммарная загрузка устройств системы - R принимает максимальное значение (соответственно, суммарное время выполнения работ минимизируется).

Теперь можно сформулировать задачу планирования: исходя из заданных ресурсов $F_1, \ldots ,F_N$ системы, матрицы трудоемкости работ - T, известного порядка использования ресурсов работой J_i, требуется построить план (расписание), при котором используемый критерий оценки качества принимает экстремальное значение.

Методы планирования работ.

Задачи построения наилучших планов относятся к типу задач распределения и упорядочения. Особенностью их является комбинаторный характер. Точное решение этих задач предполагает использование методов математического программирования - линейного, целочисленного, динамического. Трудоемкость этих методов ограничивает возможность их применения для оперативного планирования работ, в связи с чем используют эвристические (приближенные) методы планирования. Эвристический подход позволяет при умеренном объеме вычислений получать решения, которые, возможно, не являются оптимальными, но значительно лучше произвольных.

В случае использования критерия минимума суммарного времени выполнения работ, минимизация критерия достигается путем совмещения работы устройств системы во времени: чем больше во времени совмещена работа отдельных устройств, тем меньше времени будет затрачено на выполнение пакета задач. Чтобы решить задачу планирования в такой постановке, необходимо иметь данные о трудоемкости каждой работы и порядке использования ресурсов системы каждой работой.

Трудоемкость работ будем представлять матрицей T, элементы τ_{ij} которой определяются априори известными детерминированными значениями. Простые алгоритмы планирования по критерию минимума суммарного времени выполнения работ могут быть построены в предположении, что все работы используют ресурсы системы в одинаковом порядке.

В дальнейшем будем использовать следующий порядок прохождения фаз выполнения работ в системе:

- ввод - обработка;
- ввод - обработка - вывод.

Кроме того, будем считать, что на каждой фазе выполнения работы используется лишь один ресурс. Так, канал ввода - единственный для всех работ, и тогда фаза ввода для различных работ будет выполняться только последовательно. Аналогично, ресурс процессора и канала вывода - также единственные. Тогда процесс выполнения работ разделяется на ВВОД - ОБРАБОТКУ или на ВВОД - ОБРАБОТКУ - ВЫВОД, выполняемые последовательно. Параллелизм возможен за счет совмещения ввода, обработки и вывода, причем только для различных работ.

Планирование на основе двухфазной модели

Двухфазная модель системы, используемая в целях решения задачи планирования работ, показана на рис.4.

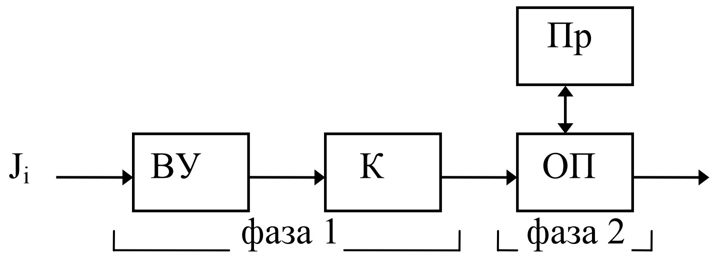

Рис.4

Первая фаза - фаза ввода, представлена устройством ввода - ВУ и каналом - К. Вторая фаза - фаза обработки, реализуется подсистемой "процессор-оперативная память". Продолжительность выполнения работ J_1, ... ,J_M на первой фазе равна $\tau_{11},...,\tau_{M1}$ и на второй фазе - $\tau_{12},...,\tau_{M2}$, соответственно.

Алгоритм оптимального планирования в такой постановке задачи состоит из следующих этапов.

1. Отметить начало очереди работ позицией A := 1, конец очереди - позицией B := M.

2. В матрице трудоемкости найти минимальное значение

$$\tau = \min(\tau_{11}, \tau_{12}, ..., \tau_{M1}, \tau_{M2}) .$$

3. Выделить работы $J_\alpha,...,J_\omega$, $(\alpha,\omega = 1,2,...,M)$, для которых

$$\tau_{ij} = \tau\ , \qquad i = \alpha,...,\omega\,;\, j = 1,2\ .$$

4. Если выделена единственная работа J_α, то при $\tau_{\alpha 1} = \tau$, она ставится в начало очереди, определяемое позицией A, а при $\tau_{\alpha 2} = \tau$ - в конец очереди, определяемый позицией B. После этого перейти к п.7 алгоритма.

5. Если выделено несколько работ $J_\alpha,...,J_\omega$, то они разделяются на две группы:

1) с одинаковыми значениями $\tau_{\alpha 1},...,\tau_{\omega 1} = \tau$;

2) с одинаковыми значениями $\tau_{\alpha 2},...,\tau_{\omega 2} = \tau$.

6. Работы из первой группы заносятся в начало очереди, в позиции A, A+1, ... в порядке увеличения значений $\tau_{\alpha 2},...,\tau_{\omega 2}$. Работы из второй группы заносятся в конец очереди в позиции B, B-1, ... в порядке увеличения значений $\tau_{\alpha 1},...,\tau_{\omega 1}$.

7. После включения работы в очередь (в позицию A или B) работа удаляется из матрицы трудоемкости. Переменной A или B присваивается новое значение A :=A+1 или B :=B-1.

8. Процесс продолжается от п.2 до распределения всех работ по позициям очереди.

Пример расчета плана для двухфазной модели.

Упорядочим работы по критерию минимума суммарного времени выполнения работ в двухфазной системе, матрица трудоемкости которой имеет вид:

$$T = \begin{array}{c} J_1 \\ J_2 \\ J_3 \\ J_4 \\ J_5 \\ J_6 \end{array}\begin{bmatrix} 2 & 3 \\ 3 & 5 \\ 4 & 1 \\ 1 & 2 \\ 1 & 4 \\ 3 & 1 \end{bmatrix}.$$

В соответствии с вышеизложенным алгоритмом, выполним следующее.

1. Присвоим A:=1 и B:=6.

2. Найдем минимальное значение τ=1.

3. Выберем работы, для которых $\tau_{ij} = \tau = 1$:

$$\begin{array}{c} J_3 \\ J_4 \\ J_5 \\ J_6 \end{array} \begin{bmatrix} 4 & 1 \\ 1 & 2 \\ 1 & 4 \\ 3 & 1 \end{bmatrix}.$$

4. Разделим работы на группы с одинаковыми значениями $\tau_{i1} = \tau = 1$ и $\tau_{i2} = \tau = 1$:

$$\begin{array}{c} J_4 \\ J_5 \end{array} \begin{bmatrix} 1 & 2 \\ 1 & 4 \end{bmatrix}, \qquad \begin{array}{c} J_3 \\ J_6 \end{array} \begin{bmatrix} 4 & 1 \\ 3 & 1 \end{bmatrix}.$$

5. Работы J_4 и J_5 включаем в начало очереди, в позиции A и A+1, т.е. в позиции 1 и 2. Работы J3 и J6 включаем в конец очереди в позиции B-1 и B, т.е. в позиции 5 и 6. Тогда план работ на данном этапе планирования примет вид :

1	2	3	4	5	6
J_4	J_5	-	-	J_3	J_6

6. Начало и конец очереди отмечаем позициями A:=A+2 и B:=B-2, в результате чего они примут значения 3 и 4, соответственно. После удаления распределенных работ J_4, J_5, J_3, J_6 вид матрицы трудоемкости следующий:

$$T = \begin{array}{c} J_1 \\ J_2 \end{array} \begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix}.$$

7. В последней матрице минимальное значение τ=2. Тогда работу J_1 включаем в начало очереди в позицию A = 3, а оставшуюся работу J_2 - в позицию B = 4. Теперь окончательный план имеет вид:

1	2	3	4	5	6
J_4	J_5	J_1	J_2	J_3	J_6

Диаграмма выполнения этих работ в системе показана на рис. 5.

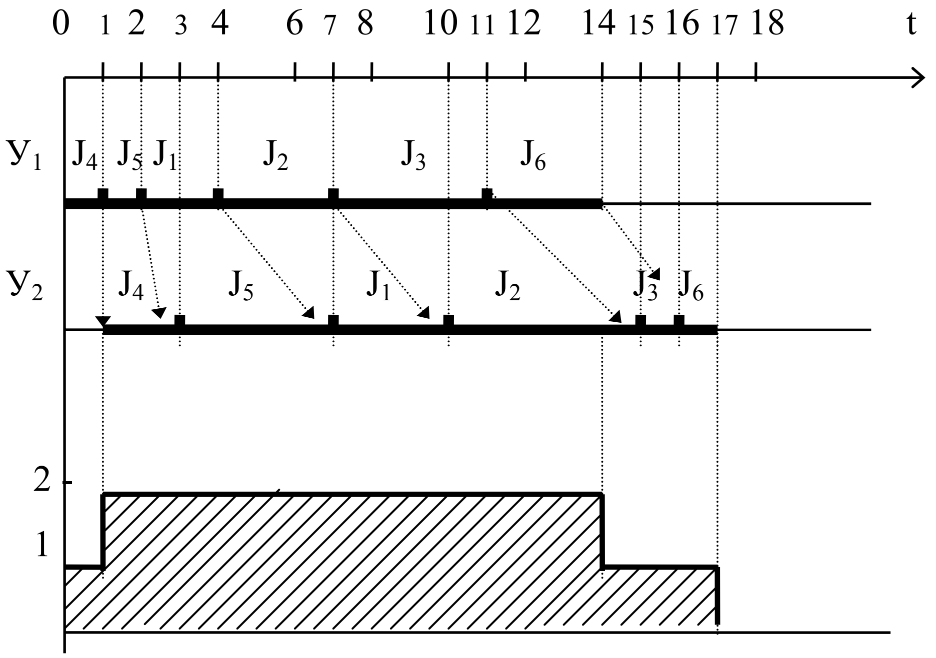

Рис. 5.

Заштрихованная область является диаграммой (диаграмма Ганта), наглядно отражающей степень совмещения работы устройств $У_1$ и $У_2$ во времени при данной последовательности запуска работ на выполнение. При любой другой последовательности запуска работ суммарное время их выполнения будет не меньше 17 ед.

Планирование на основе трехфазной модели

Трехфазная модель работает по схеме, приведенной на рис.6.

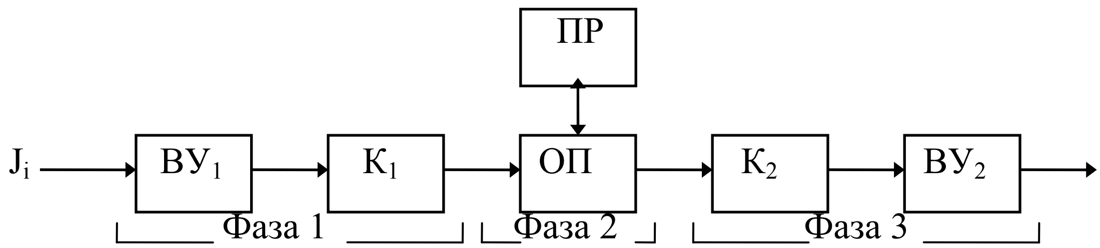

Рис.6.

Здесь фаза ввода реализуется устройством ввода $ВУ_1$ и каналом $К_1$. Фаза обработки реализуется подсистемой “процессор-оперативная память”. Фаза вывода - каналом $К_2$ и устройством $ВУ_2$.

Работа J_i , (i = 1,...,M) на первой фазе выполняется в течение времени τ_{i1}, на второй фазе - τ_{i2}, на третьей фазе τ_{i3}. Одновременно могут выполняться три работы, одна из которых находится в фазе ввода, вторая - в фазе обработки, третья - в фазе вывода.

Алгоритм оптимального планирования работ в трехфазной модели сводится к вышерассмотренному алгоритму для двухфазной модели, если выполняется хотя бы одно из следующих ограничений:

1) $\min(\tau_{i1}) \geq \max(\tau_{i2})$,
2) $\min(\tau_{i3}) \geq \max(\tau_{i2})$, $i = 1,...,M$.

Как установлено С. Джонсоном, в этом случае трехстолбцовая матрица трудоемкости преобразуется в двухстолбцовую путем попарного сложения элементов первого и второго столбцов, а также элементов второго и третьего столбцов, т.е. элементы ϑ_{i1} и ϑ_{i2}, (i = 1,...,M) новой матрицы определяются так:

$$\vartheta_{i1} = \tau_{i1} + \tau_{i2} , \quad \vartheta_{i2} = \tau_{i2} + \tau_{i3} .$$

К полученной таким способом матрице применяется ранее рассмотренный алгоритм планирования работ.

Пример расчета плана для трехфазной модели.

Упорядочим работы по критерию минимума суммарного времени выполнения работ в трехфазной системе, матрица трудоемкости которой имеет вид:

$$T = \begin{array}{c} J_1 \\ J_2 \\ J_3 \\ J_4 \\ J_5 \end{array} \begin{bmatrix} 2 & 3 & 4 \\ 1 & 1 & 5 \\ 4 & 2 & 5 \\ 5 & 3 & 7 \\ 3 & 4 & 6 \end{bmatrix}$$

Данная матрица имеет следующие характеристики:

$$\min(\tau_{i1}) = 1; \quad \max(\tau_{i2}) = 4 \ ; \quad \min(\tau_{i3}) = 4 .$$

Следовательно, одно из условий Джонсона выполнено и можно перейти к двухфазной модели, используя двухстолбцовую матрицу:

$$T = \begin{array}{c} J_1 \\ J_2 \\ J_3 \\ J_4 \\ J_5 \end{array} \begin{bmatrix} 5 & 7 \\ 2 & 6 \\ 6 & 7 \\ 8 & 10 \\ 7 & 10 \end{bmatrix} .$$

Применяя вышерассмотренный алгоритм планирования для двухфазной модели, можно построить оптимальный план запуска работ на выполнение для трехфазной модели:

1	2	3	4	5
J_2	J_1	J_3	J_5	J_4

Диаграмма выполнения этих работ в системе показана на рис. 7.

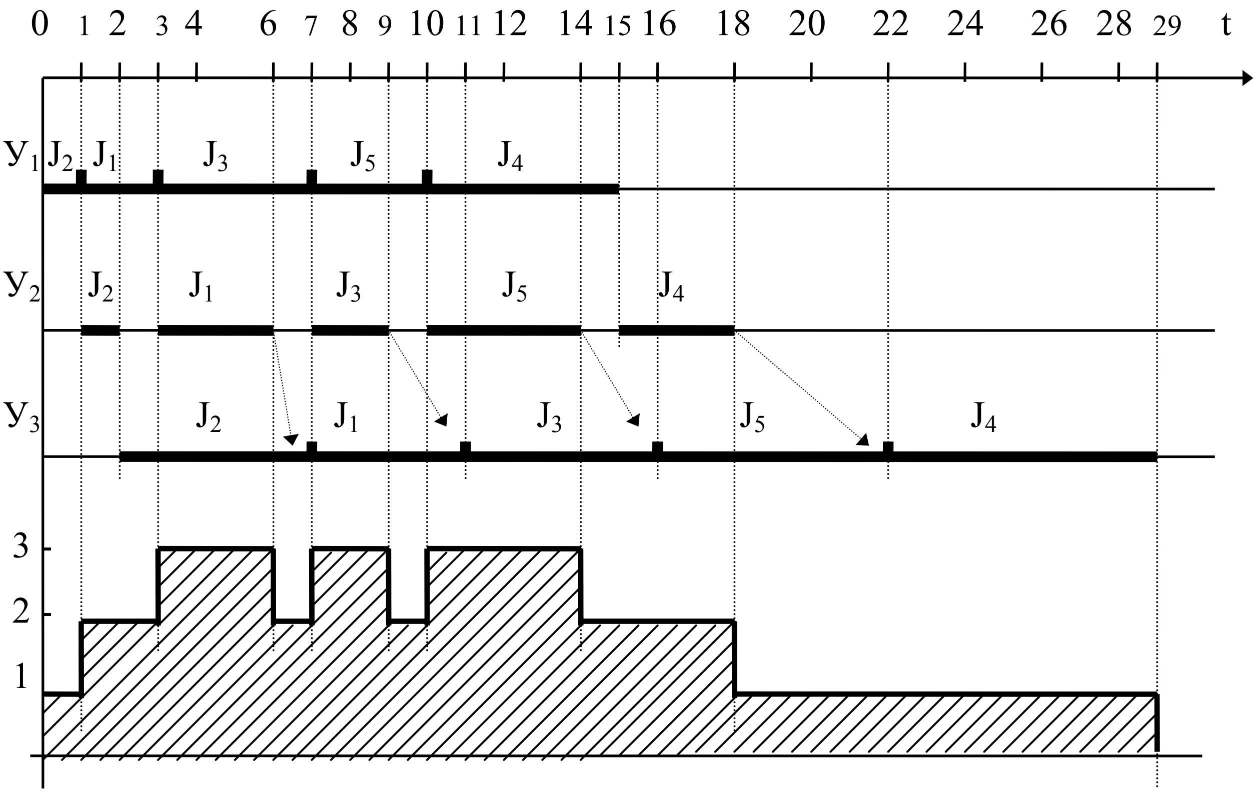

Рис. 7.

Суммарное время выполнения работ в системе составляет 29 ед., причем устройство $У_1$ работает 15 ед. времени, устройство $У_2$ - 13 ед., $У_3$ - 27 ед.

Коэффициенты загрузки устройств: $\rho_1 = 0.52$; $\rho_2 = 0.45$; $\rho_3 = 0.93$. Средняя загрузка устройства в системе составит величину:

$$R = \frac{1}{3}\sum_{i=1}^{3}\rho_i = 0.63 \ .$$

Планирование на основе эвристических алгоритмов.

В тех случаях, когда ограничения Джонсона не выполняются, точного решения задачи планирования не существует даже для трехфазной модели.

На практике, тем не менее, приходится решать задачи планирования работ, выполняемых более, чем тремя устройствами. В таких случаях прибегают к использованию эвристических алгоритмов планирования, позволяющих получить приемлемые для практических целей результаты. Один из алгоритмов данного класса, применительно к трехфазной системе, имеет следующую последовательность действий.

1. Выделяется номер фазы k с наибольшей суммарной продолжительностью работ:

$$\max_{k}\,(\sum_{i=1}^{M}\tau_{ik}) \ , \qquad k = 1,2,3.$$

2**. Если k = 1**, то работы запускаются в порядке убывания величин $(\tau_{i2} + \tau_{i3})$.

3. **Если k = 3**, то работы запускаются в порядке возрастания величин $(\tau_{i1} + \tau_{i2})$.

4. **Если k = 2**, то работы упорядочиваются на основе алгоритмов двухфазного планирования, причем из матрицы трудоемкости исключается второй столбец.

В качестве иллюстрации воспользуемся матрицей трудоемкости из предыдущего примера. Тогда суммарные трудоемкости фаз 1, 2, 3 равны, соответственно:

$$\sum_{i=1}^{5}\tau_{i1} = 2 + 1 + 4 + 5 + 3 = 15 \text{ ед. };$$

$$\sum_{i=1}^{5}\tau_{i2} = 3 + 1 + 2 + 3 + 4 = 13 \text{ ед.};$$

$$\sum_{i=1}^{5}\tau_{i3} = 4 + 5 + 5 + 7 + 6 = 27 \text{ ед.}$$

Следовательно, $\max\limits_{k}(\sum\limits_{i=1}^{5}\tau_{ik}) = 27$ ед., поэтому третья фаза имеет максимальную трудоемкость. В таком случае работы следует запускать в порядке возрастания величин $(\tau_{i1} + \tau_{i2})$, что дает следующий план:

1	2	3	4	5
J_2	J_1	J_3	J_5	J_4

В данном конкретном случае применение эвристического алгоритма привело к тому же результату, что и в случае точного алгоритма в предыдущем примере.

Оценка эффективности эвристических методов планирования.

Эффективность эвристических алгоритмов планирования принято оценивать степенью близости суммарного времени выполнения работ - T к минимально возможному времени T_0 . Значение T_0 оценивается следующим образом.

Анализируя матрицу трудоемкости работ, выделяют фазу k = 1,2,...,N, суммарная продолжительность работ в которой имеет максимальное значение. Обозначив через T_2 - продолжительность выполнения работ на наиболее трудоемкой фазе, можно записать:

$$T_2 = \max_{k}(\sum_{i=1}^{M}\tau_{ik}).$$

Очевидно, что в отрезок времени T_2 можно уложить большинство работ, выполняемых на менее трудоемких фазах $f = 1,...,k\text{-}1,k+1,...,N$, за исключением работы, которая должна быть выполнена сначала в фазах 1,2,...,k-1 до того момента, как она поступит на обработку в фазу k, начиная отсчет времени для этой фазы, а также работы, которая должна быть завершена после окончания отрезка времени T_2 на фазах k+1,...,N, заканчивая отсчет времени для выполнения работ в системе.

Естественно выбрать в качестве этих двух работ такие, чтобы продолжительность их до k-ой фазы и после k-ой фазы была минимальной.

Обозначив через T_1 - продолжительность выполнения первой работы до момента ее попадания в фазу k, можно записать:

$$T_1 = \begin{cases} 0 & ,k = 1, \\ \min\limits_i \left(\sum\limits_{j=1}^{k-1} \tau_{ij}\right) & ,k \neq 1; \quad i = \overline{1,M}. \end{cases}$$

Если k = 1, то начало любой работы совпадает с началом самой трудоемкой фазы, поэтому $T_1 = 0$.

Обозначив через T_3 - продолжительность выполнения работы на фазах k+1,...,N, можно записать:

$$T_3 = \begin{cases} 0 & ,k = N, \\ \min\limits_i \left(\sum\limits_{j=k+1}^{N} \tau_{ij}\right) & ,k \neq N; \quad i = \overline{1,M}. \end{cases}$$

Если k = N, т.е. самая трудоемкая фаза завершает процесс выполнения работ в системе, то $T_3 = 0$.

Теперь минимально возможное время выполнения работ в N-фазной системе будет иметь значение:

$$T_0 = T_1 + T_2 + T_3 = \min_i\left(\sum_{j=1}^{k-1}\tau_{ij}\right) + \max_k\left(\sum_{i=1}^{M}\tau_{ik}\right) + \min_i\left(\sum_{j=k+1}^{N}\tau_{ij}\right) .$$

Здесь k - номер фазы с максимальной суммарной продолжительностью выполнения работ. Формула имеет ненулевыми все компоненты, если самая трудоемкая фаза k находится в интервале $1 < k < N$.

Применим полученную оценку величины T_0 для вышерассмотренного примера трехфазной модели. Наиболее трудоемкой фазой в рассматриваемом примере является фаза 3. Тогда для данного частного случая получим:

$$T_0 = \min_i\left(\sum_{j=1}^{2}\tau_{ij}\right) + \max_k\left(\sum_{i=1}^{5}\tau_{ik}\right) + 0 = 2 + 27 = 29 \text{ ед.}$$

Полученная оценка для минимального суммарного времени выполнения работ в системе совпадает с результатом, который достигается планом, построенным на основе точного алгоритма. Однако, этого может не быть, если алгоритм эвристический. В последнем случае допустима погрешность эвристического метода, находящаяся в пределах 10 % - 15% значения T_0 .

В тех случаях, когда порядок использования устройств в процессе выполнения работ либо не одинаков для всех работ, либо вообще неизвестен, планирование с использованием критерия минимума суммарного времени выполнения работ невозможно. В этом случае используются другие критерии планирования.

Литература.

1. Основы теории вычислительных систем. Под ред. С.А. Майорова. - М.: Высшая школа, 1987, с. 131-142; 154-160.

2. Порядок выполнения лабораторной работы 2 .

- Изучить принципы и формальные модели процесса планирования работ в информационных системах.
- Подготовить процедуру программы, моделирующую алгоритм планирования в двухфазной системе.
- Подготовить процедуру программы, моделирующую алгоритм планирования в трехфазной системе.
- Получить решение, в ходе реализации разработанной программы на ЭВМ, в соответствии с вариантом задания.
- Построить диаграммы выполнения работ и диаграммы совмещения работ во времени на разных фазах (профиль системы или диаграмма Ганта) для каждой из реализуемых моделей.
- Определить коэффициенты загрузки устройств - ρ_j, среднюю загрузку устройства в системе - R.
- Подготовить в объектно-ориентированной среде разработки интерактивную форму для управления работой программы и визуализации полученных результатов.
- Подготовить отчет о выполненной работе. Форма отчета - по стандарту.

3. Варианты заданий на выполнение лабораторной работы 2.

Таблица 1.

Вариант	F_j / J_i	1	2	3	4	5	6	7	8	9	10	11	12
1	1	7.3	10.1	1.4	5.7	2.8	3.5	4.7	10.3	8.1	5.1	4.6	5.7
	2	5.4	3.2	7.7	11.2	5.3	6.4	10.8	6.2	2.6	6.2	8.9	9.4
	3	6.9	9.6	5.9	3.8	5.6	6.3	3.9	3.4	4.8	7.5	8.5	8.9
2	1	5.4	4.6	5.6	7.4	6.7	9.3	6.3	9.7	8.3	5.9	8.9	3.8
	2	7.5	10.1	10.3	2.4	1.8	10.8	2.3	7.5	7.2	3.5	4.6	5.8
	3	3.6	7.4	4.7	8.1	5.3	9.4	10.8	9.9	6.4	5.8	4.9	2.1
3	1	1.3	10.5	2.1	1.6	3.3	7.7	3.3	9.5	4.9	6.3	10.7	9.4
	2	4.2	8.4	3.6	5.4	7.8	2.8	6.5	7.9	7.6	5.2	8.8	5.2
	3	9.7	12.4	10.1	3.4	13.1	1.2	7.4	4.8	6.6	3.4	4.7	4.3
4	1	1.3	3.7	7.1	3.5	5.7	5.6	5.6	3.8	8.4	10.1	10.1	1.8
	2	5.4	6.8	6.7	3.4	3.4	7.3	8.4	10.3	5.4	9.3	1.5	3.4
	3	9.9	7.3	5.4	9.3	5.1	3.3	3.4	3.6	3.7	7.3	7.4	6.5
5	1	1.4	2.7	6.1	4.4	6.1	4.4	7.1	8.2	7.2	1.2	4.8	10.3
	2	5.1	2.5	6.3	7.3	5.5	7.9	5.7	8.1	8.9	8.3	2.2	9.1
	3	1.4	2.4	3.7	3.8	8.2	6.6	2.4	6.2	9.2	1.9	9.7	3.6
6	1	3.4	1.7	1.3	3.4	7.5	6.1	9.1	8.2	1.2	10.5	5.3	2.9
	2	4.6	4.4	2.8	6.2	5.2	8.1	7.2	10.8	9.4	2.6	1.6	4.8
	3	1.5	3.5	5.1	1.9	1.1	3.3	4.8	2.3	7.5	3.1	3.4	2.7
7	1	1.9	4.2	4.2	6.7	8.5	3.7	5.8	2.3	1.8	4.9	1.6	3.9
	2	5.3	2.8	3.1	7.6	3.4	9.4	4.1	1.2	3.9	3.7	4.0	1.5
	3	3.4	3.5	6.7	4.4	4.5	7.7	7.0	3.6	1.9	5.7	4.8	6.9

Продолжение таблицы 1

Вари ант	F_j / J_i	1	2	3	4	5	6	7	8	9	10	11	12
8	1	7.5	3.4	2.6	6.2	6.3	3.7	5.5	3.1	10.4	8.9	3.8	1.3
	2	9.4	9.1	2.4	2.4	3.3	7.4	4.7	9.6	3.1	11.1	2.9	1.0
	3	5.3	8.4	4.1	3.5	2.1	2.3	8.1	10.5	7.1	2.3	1.2	3.4
9	1	1.8	4.7	2.9	1.6	5.4	4.3	3.3	2.0	9.9	4.9	2.3	2.6
	2	1.3	2.8	1.5	4.3	1.7	6.2	1.9	8.4	2.1	10.6	2.5	12.7
	3	2.7	1.4	3.6	4.5	3.1	1.8	7.2	4.1	3.5	2.2	11.8	2.4
10	1	1.0	1.2	3.5	9.8	1.6	9.0	5.6	7.9	1.7	5.8	8.0	3.5
	2	1.3	2.8	4.9	5.1	1.5	9.7	2.7	1.8	5.7	10.2	2.6	5.9
	3	2.9	1.4	3.1	9.9	7.8	2.1	2.2	10.1	1.9	9.6	10.3	2.5
11	1	2.3	2.0	4.3	2.5	7.4	1.1	9.3	6.8	1.6	2.5	7.3	9.5
	2	1.1	6.1	8.4	6.2	7.2	2.4	10.4	1.3	7.5	9.4	1.8	7.6
	3	3.2	9.4	9.2	3.3	9.3	6.3	7.9	8.1	7.3	1.6	2.4	1.5
12	1	1.2	7.5	3.4	3.0	3.7	6.3	5.7	7.5	3.5	7.0	6.7	1.5
	2	7.7	3.7	5.3	6.5	6.5	5.7	1.3	2.6	7.3	7.7	7.4	7.7
	3	3.5	3.4	9.7	9.1	6.1	4.2	2.7	4.3	3.5	2.1	3.6	8.1
13	1	5.7	1.3	7.3	5.7	4.5	4.0	1.2	1.5	1.4	1.9	2.4	2.3
	2	8.8	3.1	1.6	9.8	8.9	1.5	5.4	8.5	1.6	2.1	8.4	2.3
	3	6.6	8.1	2.8	3.7	1.4	1.8	6.4	3.6	1.7	2.1	2.5	4.1
14	1	2.6	6.0	3.6	2.7	3.3	6.9	4.2	5.0	3.9	1.4	3.0	7.3
	2	3.5	8.7	1.7	6.8	3.7	2.9	1.9	7.1	2.2	8.2	4.0	8.3
	3	6.7	3.4	8.6	2.8	1.8	2.9	3.8	3.8	4.1	3.9	7.2	3.1

Продолжение таблицы 1

Вари ант	F_j / J_i	1	2	3	4	5	6	7	8	9	10	11	12
15	1	17.3	18.1	11.2	15.7	12.8	13.5	14.7	11.3	18.1	15.1	14.6	15.7
	2	5.4	3.2	7.7	11.2	5.3	6.4	10.8	6.2	2.6	6.2	8.9	9.4
	3	6.9	9.6	5.9	3.8	5.6	6.3	3.9	12.2	4.8	7.5	8.5	8.9
16	1	10.4	4.6	5.6	7.4	6.7	9.3	6.3	9.7	8.3	5.9	8.9	3.8
	2	7.5	10.1	10.3	12.4	1.8	10.8	2.3	7.5	7.2	3.5	4.6	5.8
	3	3.6	7.4	4.7	8.1	5.3	9.4	10.8	9.9	6.4	5.8	4.9	2.1
17	1	9.8	10.5	12.1	11.6	13.3	17.7	13.3	9.5	14.9	16.3	10.7	9.4
	2	4.2	8.4	3.6	5.4	7.8	2.8	6.5	7.9	7.6	5.2	8.8	5.2
	3	9.7	12.4	10.1	3.4	13.1	8.8	7.4	4.8	6.6	3.4	4.7	4.3
18	1	10.3	3.7	7.1	3.5	5.7	5.6	5.6	3.8	8.4	10.1	10.1	1.8
	2	5.4	6.8	6.7	3.4	3.4	17.3	8.4	10.3	5.4	9.3	1.5	3.4
	3	9.9	7.3	5.4	9.3	5.1	3.3	3.4	3.6	3.7	7.3	7.4	6.5
19	1	9.4	12.7	16.1	14.4	12.1	10.4	11.1	13.2	15.2	10.1	14.8	10.3
	2	5.1	2.5	6.3	7.3	5.5	7.9	5.7	8.1	8.9	8.3	2.2	9.1
	3	11.4	12.4	13.7	13.8	12.3	16.6	12.4	16.2	9.1	11.9	9.7	13.6
20	1	13.4	1.7	1.3	3.4	7.5	6.1	9.1	8.2	1.2	10.5	5.3	2.9
	2	4.6	4.4	2.8	6.2	5.2	8.1	7.2	15.8	9.4	2.6	1.6	4.8
	3	1.5	3.5	5.1	1.9	1.1	3.3	4.8	2.3	7.5	3.1	3.4	2.7
21	1	11.9	14.2	10.2	16.7	18.5	13.7	15.8	12.3	11.8	14.9	11.6	13.9
	2	5.3	2.8	3.1	7.6	3.4	9.4	4.1	1.2	3.9	3.7	4.0	1.5
	3	13.4	13.5	16.7	14.4	14.5	17.7	17.0	13.6	11.9	15.7	14.8	16.9

Продолжение таблицы 1

Вари ант	F_j / J_i	1	2	3	4	5	6	7	8	9	10	11	12
22	1	2.5	3.4	2.6	6.2	6.3	3.7	5.5	3.1	10.4	8.9	3.8	1.3
	2	9.4	9.1	2.4	2.4	3.3	7.4	4.7	9.6	3.1	1.1	2.9	1.0
	3	5.3	8.4	4.1	3.5	2.1	2.3	8.1	10.5	7.1	2.3	1.2	3.4
23	1	12.8	14.7	12.7	13.6	15.4	14.3	13.3	15.0	19.9	14.9	18.3	16.6
	2	1.3	2.8	1.5	4.3	1.7	6.2	1.9	8.4	2.1	10.6	2.5	12.7
	3	12.7	14.4	13.6	14.5	13.1	12.8	17.2	14.1	13.5	17.2	16.8	14.8
24	1	10.0	1.2	3.5	9.8	1.6	9.0	5.6	7.9	1.7	5.8	8.0	3.5
	2	1.3	2.8	4.9	5.1	1.5	9.7	2.7	1.8	5.7	10.2	2.6	15.9
	3	2.9	1.4	3.1	9.9	7.8	2.1	2.2	10.1	1.9	9.6	10.3	2.5
25	1	2.3	2.0	4.3	2.5	7.4	1.1	9.3	6.8	1.6	2.5	7.3	9.5
	2	1.1	6.1	8.4	6.2	7.2	2.4	10.4	1.3	7.5	9.4	1.8	7.6
	3	13.2	19.4	19.2	13.3	19.3	16.3	17.9	18.1	17.3	11.6	12.4	10.5
26	1	8.2	7.5	3.4	3.0	3.7	6.3	5.7	7.5	3.5	7.0	6.7	1.5
	2	7.7	3.7	5.3	6.5	6.5	5.7	1.3	2.6	7.3	7.7	7.4	7.7
	3	13.5	3.4	9.7	9.1	6.1	4.2	2.7	4.3	3.5	2.1	3.6	8.1
27	1	5.7	1.3	7.3	5.7	4.5	4.0	1.2	1.5	1.4	1.9	2.4	2.3
	2	8.8	3.1	1.6	9.8	8.9	1.5	5.4	8.5	1.6	2.1	8.4	2.3
	3	16.6	9.8	12.8	13.7	11.4	11.8	16.4	13.6	11.7	12.1	12.5	14.1
28	1	5.6	6.0	3.6	2.7	3.3	6.9	4.2	5.0	3.9	1.4	3.0	7.3
	2	3.5	8.7	1.7	6.8	3.7	2.9	1.9	7.1	2.2	8.2	4.0	8.3
	3	6.7	3.4	18.6	2.8	1.8	2.9	3.8	3.8	4.1	3.9	7.2	3.1

Примечание: в таблице для каждого варианта вначале рассматривается двухфазная модель, имеющая ресурсы F_1 и F_2 , и рассчитывается оптимальный план выполнения работ. После этого рассматривается трехфазная модель с ресурсами F_1 ,F_2 ,F_3 и рассчитывается план выполнения работ на основе алгоритмов для трехфазной модели.

4. Контрольные вопросы.

1. Дать определение терминам “работа”, “задание”, “ресурс” применительно к предметной области, связанной с планированием процессов обработки данных в информационных системах.
2. Как определяется процесс планирования в информационной системе, и к каким эффектам он приводит?
3. Какие системные средства планирования работ используются в информационной системе?
4. В чем принципиальное различие функций планировщика и супервизора при реализации планирования работ?
5. Какие критерии эффективности плана используются в практике?
6. Как определить оптимальный план запуска работ на выполнение?
7. Сформулируйте задачу планирования работ в информационной системе.
8. К какому классу методов планирования относятся эвристические методы?
9. Какие требования предъявляются к алгоритмам планирования?
10. Относится ли задача планирования к классу оптимизационных задач?
11. Какими данными необходимо располагать, чтобы решить задачу планирования по критерию минимума суммарного времени выполнения работ?
12. Как оценивается эффективность эвристического алгоритма планирования?
13. Как рассчитать среднюю загрузку устройств в системе по диаграмме выполнения работ?
14. Позволяет ли трехфазная модель всегда получать точное решение задачи планирования?

15. Что следует предпринять, если для трехфазной модели условия Джонсона не выполняются?

16. Какие преобразования выполняются над матрицей T в том случае, если возможен переход от трехфазной модели к двухфазной?

17. Допускает ли применение эвристических методов оптимальное решение задачи планирования?

18. Назовите технологические операции над данными, осуществляемые на первой, второй и третьей фазах модели информационной системы?

19. Какую характеристику работы J_i содержит матрица $T = \left[\tau_{ij}\right]$?

20. В чем состоит отрицательный эффект процесса планирования работ в информационной системе?

Дополнительная литература.

1. Олзоева С.И. Моделирование и расчет распределенных информационных систем. Учебное пособие. Изд-во ВСГТУ. г. Улан-Удэ, 2004 г.

2. Щеклеин В.С. Моделирование информационных систем. Конспект лекций. Изд-во УлГТУ. г. Ульяновск, 2002 г.

3. Клейнрок Л. Вычислительные системы с очередями. М.: Мир, 1979.

4 Моделирование систем с использованием теории массового обслуживания. Под ред. д.т.н. Д.Н.Колесникова: Учеб. Пособие /СПбГПУ. СПб, 2003.

5. Советов Б.Я., Яковлев С.А. Моделирование систем: Учебник для вузов. 3-е изд., переработанное и дополненное. М.: Высшая школа, 2001.

6. Советов Б.Я., Яковлев С.А. Моделирование систем. Практикум. Учебное пособие. М.: Высшая школа, 2003.

7. Вишневский В.М. Теоретические основы проектирования компьютерных сетей. - М.: Техносфера, 2003.