

ОРГАНИЗАЦИЯ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА И ОТЛАДКИ ПРОГРАММ В МНОГОПУЛЬТОВОЙ ЦВМ ДЛЯ ИНЖЕНЕРНЫХ РАСЧЕТОВ

В. Д. Кожухов

Харьков

Проблема повышения эффективности использования цифровых вычислительных машин — одна из актуальных проблем, стоящих перед вычислительной техникой. Одним из путей повышения эффективности ЦВМ является внедрение систем мультипрограммного управления. Особый интерес представляют системы, в которых машинное время распределено полностью между различными вычислителями. Такие системы получили название систем с полным распределением времени.

Настоящая работа посвящена вопросам организации вычислительного процесса и отладки программ в многопультной ЦВМ для инженерных расчетов с полным распределением времени. Алгоритмы работы всех устройств даны в терминах алгоритмического адресного языка¹, вопросы о занесении входной информации, а также о передаче информации между блоками машины и организации печати результатов рассмотрены ранее в статьях автора².

Решению задачи предшествует ее математическая постановка и запись во входном языке. До занесения в память машины задачи необходимо произвести стирание памяти, относящейся к данному пульту вычислителя. Для этого вычислитель нажимает на своем пульте клавишу «Общее гашение». При этом устанавливается в единицу $T_{обрk}$, T_{rk} и код гашения заносится в РР k .

Поскольку в РР k записан код гашения, то когда БЗУ обратится к данному пульту, в работу будет запущен алгоритм гашения.

Алгоритм гашения «страницы» ленты:

- | | |
|----------|---|
| 1-й такт | $0 \rightarrow Ar; 0 \rightarrow PO;$ |
| 2-й " | $Ark \rightarrow Ara; Arkk \rightarrow Ark$ |
| 3-й " | $'Ara + 1 \rightarrow Ara$ |
| 4-й " | „Запись“ |
| 5-й " | $P\{'Ara \neq 'Ark\} \quad Gt = 3$ |
| 6-й " | $0 \rightarrow Ar$ |
| 7-й " | $Rk \rightarrow Ar$ |
| 8-й " | $0 \rightarrow Ar$ |
| 9-й " | „Запись“ |
| 10-й " | $Lk \rightarrow Ar$ |
| 11-й " | „Запись“ |
| 12-й " | Запуск СОП |

¹ Е. Л. Ющенко. Адресное программирование. Гостехиздат УССР, Киев, 1963.

² В. Д. Кожухов. Об одном способе повышения эффективности вычислительной машины для инженерных расчетов. Сб. «Приборы и системы автоматизации», вып. 5. Изд-во Харьковск. ун-та, 1967.

Как только НМЛ окажется свободным по T_k или по $T_1 \dots T_7$, а РПК равным единице, номер пульта заносится в Арл и устанавливается в единичное состояние $T_{имл}$, т. е. начинается подвод ленты. Когда подвод ленты заканчивается, устанавливается в нулевое состояние триггер записи кода и по нужной дорожке на ленту начинают записываться пули, что продолжается до тех пор, пока блок головок не пройдет все зоны данного пульта. После этого лента будет остановлена, а РПК и T_k погашены.

По окончании гашения алгоритмы решения задач и исходные данные заносятся в память машины. Если при вводе входной информации нажимались кнопки операций Π_1 , Π_2 или $P\{\}$, то устанавливается в единичное состояние триггер задачи второго класса данного пульта T_{3k} . Если вводилась задача первого класса, то при этом не могли быть нажаты кнопки операций циклирования или предикат, поэтому T_{3k} остается в нулевом состоянии. После подачи команды «Счет» в зависимости от состояния T_{3k} задача ставится в очередь задач первого или второго класса. Это происходит следующим образом. При нажатии клавиши «Счет» устанавливается в единичное состояние $T_{обрk}$, в РПК заносится код заявки. Как только БЗУ обратится к данному пульта, на выходе ДРП появится потенциал алгоритма постановки задачи на очередь.

Алгоритм постановки задачи на очередь:

1-й такт	$0 \rightarrow Ar; 0 \rightarrow PO$
2-й "	$P\{T_{3k} = 0\} 1 \rightarrow T_{24}$ $1 \rightarrow T_{23}$
3-й "	$Ард\ 1 \rightarrow Ар$
4-й "	„Чтение“
5-й "	$0 \rightarrow Ар$
6-й "	$'PO \rightarrow Ар; Ардк\ 1 \rightarrow Арк$
7-й "	$P\{'Ара \neq 'Арк\} 1 \rightarrow T_{26}$ $1 \rightarrow T_{27}$
9-й "	$'Ара + 1 \rightarrow Ара; 0 \rightarrow PO$
10-й "	„k“ $\rightarrow PO$
11-й "	„Запись“
12-й "	$0 \rightarrow T_{3k}; 0 \rightarrow РПК$
13-й "	Запуск СОП
T_{24} 3-й такт	$Ард_2 \rightarrow Ар$
4-й "	„Чтение“
5-й "	$0 \rightarrow Ар$
6-й "	$'PO \rightarrow Ар; Ардк_2 \rightarrow Арк$
7-й "	$P\{'Ара \neq 'Арк\} 1 \rightarrow T_{28}$ $1 \rightarrow T_{29}$
9-й "	$'Ара + 1 \rightarrow Ара; 0 \rightarrow PO$
10-й "	„k“ $\rightarrow PO$
11-й "	„Запись“
12-й "	$0 \rightarrow T_{3k}; 0 \rightarrow РПК$
13-й "	Запуск СОП
T_{26} 8-й такт	$'Ардн_1 \rightarrow Ара; 1 \rightarrow T_{27}$
T_{28} 9-й "	$'Ардн_2 \rightarrow Ара; 1 \rightarrow T_{29}$

Примечание: 1. Триггера $T_{23} + T_{29}$ гасятся в первом такте.

2. $Ард_1$ — номер ячейки, сохраняющей номер ячейки, в которую был записан номер пульта, с этого пульта поступила предыдущая заявка на решение задачи первого класса;

Затем он подает команду «Счет», по которой машина производит перезапись содержимого «страницы» пульта в НМЛ и ставит задачу на очередь.

Алгоритм сдвига очереди задач второго класса:

1-й такт	$0 \rightarrow \text{Ар}$ $0 \rightarrow \text{РО}$
2-й "	$\text{Аро}_2 \rightarrow \text{Ар}$
3-й "	„Чтение“
4-й "	$0 \rightarrow \text{Ар}$
5-й "	$\text{РО} \rightarrow \text{Ар}; \text{Арди}_2 \rightarrow \text{Арк}$
6-й "	$\text{Р} \{ \text{Ара} \neq \text{Арк} \} \quad 1 \rightarrow \text{T}_{33}$
	$1 \rightarrow \text{T}_{33}$
8-й "	$\text{Ара} \rightarrow 1 \rightarrow \text{Ара}; \quad 0 \rightarrow \text{РО}$
9-й "	$\text{Ар} \rightarrow \text{РО}$
10-й "	$0 \rightarrow \text{Ар}$
11-й "	$\text{Аро}_2 \rightarrow \text{Ар}$
12-й "	„Запись“
13-й "	Запуск СОП.
T_{32} 7-й "	$\text{Арди}_1 \rightarrow \text{Ара}; \quad 1 \rightarrow \text{T}_{33}$

Примечание: 1. Аро_2 — номер ячейки, сохраняющей номер ячейки, в которой хранится номер пульта, задача этого пульта стояла первой на очереди до начала работы алгоритма сдвига очереди.

2. Триггера T_{32} и T_{33} гасятся в первом такте.

Когда задача второго класса оказывается первой на очереди, решение предыдущей задачи заканчивается, а результаты его выведены на ленту, гасится часть ОЗУ, предназначенная для решения задач второго класса. Затем задача переписывается с ленты в ОЗУ. По окончании перезаписи, так как на пульте вычислителя стоит режим отладки, в пусковой регистр машины заносится номер ячейки, с которой начинается запись первого пункта отладки, и машина запускается на выполнение этого пункта. Пункт плана отладки — это предписание выполнить некоторую часть алгоритма задачи. При этом время выполнения данного алгоритма вычислитель ограничивает меткой времени T_i . В процессе выполнения алгоритма формируется массив чисел, состоящий из отладочных меток, расположенных в том порядке, в котором они встретились машине во время счета программы. Если часть алгоритма, предусмотренная к выполнению планом отладки, была пройдена за время, меньшее T_i , то по достижении метки M_j машина прекращает выполнение алгоритма, переписывает в НМЛ полученный массив отладочных меток, дает команду начать перезапись его в «страницу» пульта БЗУ и вывести затем на печать печатной машинкой пульта вычислителя. Если часть алгоритма, ограниченная метками M_i и M_j , за время T_i не была пройдена, машина прерывает счет и также выдает массив отладочных меток на печать. После этого машина сдвигает очередь второго класса и начинает счет следующей задачи.

Вычислитель, получив результаты отладки на пульт, анализирует их и, если он находит, что первый пункт плана отладки прошел успешно, с помощью команд «Удалить» и «Вставить» заносит новый пункт плана отладки и вновь подает команду «Счет». Машина вновь ставит задачу на очередь и т. д.

Аналогично производится и счет задачи, только в этом случае в команде «Пуск $\text{M}_i \text{ M}_j \text{ T}_i$ » M_i — пусковая метка алгоритма, M_j — метка конца алгоритма, а $\text{T}_i = \infty$.

Команда «Удалить/ $a_1^*a_2/a_3^*a_4$ » выполняется следующим образом. Вначале в Арла заносится a_2 , а в старшие разряды АРЛк — a_1 (так как символ / ставится через каждые восемь ячеек). Затем устанавливается в единицу $T_{имл}$, начинается подвод ленты. Как только нужное место на ленте найдено, устанавливается в ноль триггер записи кода на ленту по нужной дорожке и в Арла заносится a_4 , а в Арлк — a_3 . Начинается запись нулей на ленту, что продолжается до тех пор, пока под блоком головок не окажется участок ленты, записанный в Арл.

По команде «Вставить/ $b_1^*b_2$ » b_1 и b_2 записываются в ячейку Lk . Затем вычислитель начинает заносить в «страницу» пульта вставляемый массив. Как только «страница» пульта окажется заполненной, содержимое ячейки Lk заносится в Арл, запускается подвод ленты и гасится Арл₂. Если подвод ленты окончен, начинается чтение в РЛ информации с ленты. По 38-му импульсу сдвига РЛ добавляется единица в Арл₂ и подается команда сравнения с нулем содержимого РЛ. Если РЛ = 0, добавляется единица в Арлк₂. После этого регистр ленты гасится и в РЛ записывается новое число до тех пор, пока содержимое Арла не станет равным $n + 1$. После этого лента реверсируется, вновь подводится к месту/ $b_1^*b_2$ (по Арл) и, если Арлк₂ = $n + 1$, содержимое «страницы» пульта переписывается на ленту начиная с ячейки, следующей за/ $b_1^*b_2$. После окончания перезаписи продолжается занесение вставляемого массива.

Если Арлк₂ $\neq n + 1$, то после реверса и подвода ленты к/ $b_1^*b_2$ пропускается Арлк₂ ячеек, а остальные $n + 1 - \text{Арлк}_2$ записываются в «страницу» ленты начиная с $n + 1$ ячейки. Таким образом, место для занесения содержимого первой части «страницы» ленты оказывается свободным. Тогда лента вновь реверсируется и производится перезапись содержимого «страницы» пульта в «страницу» ленты, а оттуда на ленту. После этого содержимое $n + 1, n + 2, \dots, 2(n + 1) - \text{Арлк}_2$ ячеек «страницы» ленты переписываются в 1, 2, ..., $n + 1 - \text{Арлк}_2$ ячейки этой же «страницы», лента вновь реверсируется и первая часть ее «страницы» заполняется полностью, а начиная с $n + 1$ и кончая $2(n + 1) - \text{Арлк}_2$ ячейкой «страницы» ленты заносится новая партия информации до тех пор, пока все содержимое ленты не будет сдвинуто на $n + 1 - \text{Арлк}_2$ ячеек.

Аналогично происходит занесение остальной части вставляемого массива и сдвиг информации на ленте.