

И Ф В Э  
СВМ/СПК/СЭФ 74-89

Ю.А. Белокопытов, Л.Г. Каминский,  
С.В. Клименко, А.А. Лебедев

СИСТЕМА РАТСНУ  
ДЛЯ РАМ-ФАЙЛОВ ПРЯМОГО ДОСТУПА

*E42; F51*

**Ю.А. Белокопытов, Л.Г. Каминский,  
С.В. Клименко, А.А. Лебедев**

**СИСТЕМА РАТСНУ  
ДЛЯ РАМ-ФАЙЛОВ ПРЯМОГО ДОСТУПА**

Аннотация

Белокопытов Ю.А., Каминский Л.Г., Клименко С.В., Лебедев А.А.

Система PATCHY для PAM-файлов прямого доступа. Серпухов, 1974.  
15 стр. (ИФВЭ 74-89).

Библиогр. 6.

Описана используемая в ИФВЭ на ЭВМ ICL 1906A система хранения и редактирования текстов больших программ, работающая с файлами прямого доступа. ] + ИФВЭ р. А.

Abstract

Belokopytov Yu.A., Kaminsky L.G., Klimentko S.V., Lebedev A.A.

PATCHY System for Direct Access PAM-Files. Serpukhov, 1974.

p. 15. (INEP 74-89).

Ref. 6.

The PATCHY program designed for the maintenance, updating and development of large programs used in INEP on ICL 1906A computers and intended for direct access PAM-files is described. It is fully compatible with CERN PATCHY version adapted for ICL 1906A and runs 7-10 times faster. Direct access PAM-files are supplied with catalogue of main structure elements (patches and decks). The algorithm used is the same one as for serial PAM-files, apart from selection of following deck (patch) is made by means of scanning the catalogue rather than that of serial PAM-file. The INPUT/OUTPUT routines have been optimized. Further possible improvements are reviewed.

## 1. В В Е Д Е Н И Е

В ИФВЭ для хранения и редактирования больших текстов программ обработки снимков с пузырьковых камер используется система RATCHU<sup>/4/</sup>. Эта система была разработана в ЦЕРНе и является весьма удобным и эффективным средством развития больших программ в условиях коллективного труда. Она позволяет пользователям иметь текст программы на одном накопителе ЭВМ потенциально в нескольких вариантах.

Система RATCHU предполагает хранение текстовой информации вместе с исправлениями и добавлениями в одном файле (называемом RAM - Ratchu Master) , а специальная программа-сборщик RATCHU<sup>/1/</sup> собирает пользователю требуемый вариант текста. Подготовку и редактирование такого файла осуществляет специальная программа-редактор URPA<sup>/2/</sup>. В принципе, для этих целей можно использовать произвольный редактор.

Основным элементом системы RATCHU является программа-сборщик. Режим её работы задается пользователем с помощью специальной колоды перфокарт, называемой колодой пользователя. Работой программы управляет набор директив, расположенных в колоде пользователя и в RAM-файле. Вся текстовая информация, содержащаяся в RAM-файле и колоде пользователя, подразделяется на отдельные элементы с иерархической структурой. Руководствуясь директивами, программа-сборщик выбирает указанные ей структурные элементы, выполняет требуемое редактирование и выдает

результат сборки в два выходных файла. Один из них (листинг) выдает-ся на печать, а другой (рабочий) записывается на внешнем накопителе для последующей передачи его на компиляцию.

Система RATSNU была разработана для закрытых вычислительных систем с пакетным режимом работы <sup>/3/</sup> при наличии для пользователей только личных магнитных лент. Программа-сборщик обрабатывает RAM-файл как последовательный файл.

В ИФВЭ система RATSNU используется на ЭВМ БЭСМ-6 и ICL 1906A. Закрытый режим работы <sup>/4/</sup> БЭСМ-6 и её конфигурация хорошо согласуются с идеологией этой системы. ЭВМ ICL 1906A работают в режиме мультимедиа <sup>/5/</sup> и имеют ограниченное количество магнитофонов, что затрудняет использование магнитных лент в качестве накопителей для оперативной работы с текстами. Однако пользователи имеют в своем распоряжении достаточно емкое хранилище файлов прямого доступа на сменных дисках.

Данная работа посвящена реализации на ЭВМ ICL 1906A системы хранения и редактирования текстов, внешне полностью совместимой с системой RATSNU и ориентированной на обработку RAM-файлов в режиме прямого доступа.

В настоящее время в коллаборациях по обработке снимков с пузырьковых камер стандартизирован обмен текстами программ в виде RAM-файлов, поэтому для лабораторий, занимающихся обработкой снимков, принятие системы RATSNU является очень важным.

## 2. РАБОТА ПРОГРАММЫ-СБОРЩИКА С ПОСЛЕДОВАТЕЛЬНЫМ ФАЙЛОМ

Программа-сборщик начинает свою работу с чтения колоды пользователя. Вся информация, расположенная после карт, определяющих режим работы сборщика, рассматривается как один сплошной массив независимо

от принадлежности либо к колоде пользователя, либо к RAM-файлу, причём сборщик обрабатывает этот массив последовательно за один проход. В любом месте колоды пользователя может встретиться карта переключения указателя чтения. После прочтения указанного на ней файла, происходит возврат чтения колоды пользователя под карту-переключатель, пока не встретится специальная директива, указывающая конец работы сборщика.

В этом массиве вся информация структурно делится на три уровня иерархии: петч, колода, карта. Соответственно имеется три уровня адресации. Массив делится на петчи, определяемые своими именами с помощью директив. Обычно петчами определяют части программ, выполняющих вполне законченные процедуры (это так называемые программные модули), пакеты подпрограмм целевого назначения, наборы директив, формирующих варианты программ для конкретных приложений, и т.п. Петчи состоят из колод, которые также определяются своими идентификаторами и обычно состоят из отдельных подпрограмм. Карты в колодах определяются порядковыми номерами.

В процессе просмотра массива сборщик анализирует встречающиеся директивы (для простоты опознавания все директивы снабжаются символом "+" в первой позиции карты). В основном директивы сводятся к указанию, какой петч или колоду следует рассматривать в процессе сборки. Информация, содержащаяся в колоде пользователя до первой директивы определения петча, автоматически приписывается к петчу со стандартным именем CRADLE, который рассматривается всегда. Те петчи и колоды, которые не выбраны для рассмотрения, пропускаются. Тем не менее, при этом приходится читать их карта за картой, чтобы зафиксировать появление выбранного элемента в последовательном файле. При опознавании очередной директивы на выбор петча или колоды происходит запись информации в специальный список программы-сборщика (Приложение 1). Когда появляется директива определения петча или колоды, происходит поиск по этому списку для выяснения, выбран данный элемент или нет.

В результате устанавливаются флаги, состояния которых переспределяются для каждого нового элемента (петча, колоды). Эти флаги имеют соответствующую иерархическую подчиненность, однако в дальнейшем это не существенно. Если элемент выбран, то в зависимости от установленного режима вывода он передается в соответствующий выходной файл (на листинг, в рабочий файл или в оба сразу). Режим вывода в процессе сборки может переопределяться специальными директивами для каждого отдельного петча или всех сразу.

Во вводимом массиве могут встречаться специальные элементы, так называемые последовательности, которые не входят в описанную иерархическую структуру. Это набор следующих одна за другой карт, снабженный идентификатором. Обычно последовательностью называют часто встречающийся набор операторов, как правило, декларативных. Последовательности также поступают в общий список сборщика, а при появлении во вводимом массиве специальной директивы текст соответствующей последовательности копируется в выходной файл. Сборщик рассматривает только такие последовательности, которые принадлежат выбранным петчам.

В процессе сборки возможно также элементарное редактирование собираемой информации. Директивы редактирования и информация к ним, которые в обычных редакторах образуют редактируемый файл, здесь находятся в одном массиве с редактируемыми файлами. При появлении директивы редактирования она запоминается в общем списке сборщика (Приложение 1), а при рассмотрении каждой карты текста производится проверка, не было ли на неё редакции. При этом редактирование может быть произвольной глубины, т.е. независимо от того, должна ли карта поступить в выходной файл или в общий список сборщика, над ней производится редакция согласно уже накопленной информации в списке.

Более подробно все директивы сборщика описаны в инструкции для пользователя<sup>1/</sup>. Формат общего списка программы-сборщика приведен в Приложении 1.

Идеология хранения текстов больших программ в виде РАМ-файлов и опыт работы с программой-сборщиком РАТСНУ показывают, что обычно в выходные файлы передается небольшая часть информации, содержащейся во входном файле, т.е. просмотр всего входного файла, вызванный предположением об его последовательности, в большей части излишен.

Структура РАМ-файла и логика работы сборщика такова, что, вообще говоря, имеется прямой доступ к каждой отдельной карте, адрес которой состоит из трех компонент: имя петча, имя колоды, номер карты. Для карты из определенной последовательности адрес состоит из двух компонент: имя последовательности, номер карты. Последовательная организация РАМ-файла существенна только при определении порядка выполняемых сборщиком директив.

### 3. РЕАЛИЗАЦИЯ ПРОГРАММЫ-СБОРЩИКА ДЛЯ ФАЙЛА ПРЯМОГО ДОСТУПА

Организация РАМ-файла в виде файла прямого доступа позволяет избежать просмотра невыбираемых элементов и использовать достоинства дисков как устройств прямого доступа, сократив тем самым время работы программы-сборщика.

Для обеспечения прямого доступа к структурным элементам РАМ-файла по их идентификаторам, РАМ-файл должен быть снабжен каталогом. Поскольку РАМ-файл является текстовым эталоном, а оперативные исправления находятся в колоде пользователя, каталог РАМ-файла создается один раз при занесении его на диск. Для создания каталога естественно воспользоваться программой-редактором UPPA<sup>2/</sup> (директива \*INDEX), однако у нас эта процедура организована в виде отдельной программы \*WRAT. Каталог организован как иерархическая структура, представляющая собой связанный неориентированный граф в виде дерева блоков петчей и колод. Каждый блок петча имеет связь к списку своих колод и

к последующему петчу. Структура этого каталога отражает последовательную организацию исходного РАМ-файла. Формат блоков петчей и колод в каталоге приведен в Приложении 2.

Собственно РАМ-файл на диске полностью повторяет ленточный РАМ-файл. Карты на диск записываются в виде записей переменной длины — последние пробелы в карте опускаются. Длина записи записывается в первом слове.

Для РАМ-файла прямого доступа алгоритм программы-сборщика остается таким же, как и для последовательного РАМ-файла, кроме того, что переход к рассмотрению очередной колоды или петча осуществляется путем просмотра каталога, а не просмотром последовательного РАМ-файла. Карты директив из РАМ-файла читаются и распознаются при этом обычным образом. Чтение карт из дискового РАМ-файла ведется блоками размером 1К слов. Такой размер буфера был выбран в результате анализа распределения длин адресуемых структурных элементов часто используемых РАМ-файлов программ системы обработки фильмовой информации с пузырьковых камер.

Реализованный вариант программы-сборщика может обрабатывать и последовательные РАМ-файлы, находящиеся в хранилище файлов (filestore) под контролем системы GEORGE<sup>5/</sup>. Переключение режимов осуществляется после анализа директивы указателя чтения, на которой определяется тип файла. Формат этой карты описан в Приложении 3.

#### 4. ОПТИМИЗАЦИЯ РАБОТЫ ПРОГРАММЫ-СБОРЩИКА

Программа-сборщик РАТСНУ в основном написана на языке ФОРТРАН. Наиболее очевидным способом повышения эффективности её работы является создание специализированного пакета подпрограмм ввода/вывода, непосредственно обращающегося к системным процедурам обмена. Отказ от стандартного форматного обмена позволяет выиграть время работы программы и сэкономить память.

Кроме того, большого эффекта можно добиться за счёт отказа от использования системных средств отладки (TRACE), так как программа-сборщик RAUCIU является системной программой и работает бессбойно. Для этих целей созданы специальные пакеты подпрограмм.

Анализ особенностей системы адресации ЭВМ ICL 1906A и её работы в режиме мультимоступа показывает, что наиболее эффективной является программа, использующая прямую моду адресации (15AM, DEM)<sup>/5/</sup>. Это означает, что размеры программы не должны превышать 32K слов. Аналогичное требование возникает и на ЭВМ БЭСМ-6, у которой длина математического адреса принципиально ограничена 15 разрядами. Однако некоторые широко используемые РАМ-файлы (например, LBCG<sup>/6/</sup>) требуют для сборки такую длину списка, что программа-сборщик превышает эти границы.

При анализе структуры списка сборщика выясняется, что около 40-60% его длины занимают тексты карт корректов. Одним из путей сокращения требуемой длины списка в оперативной памяти может быть следующий: на внешнем накопителе создается рабочий файл прямого доступа, в который записываются тексты карт корректов и последовательностей. При этом только адреса этикет карт в рабочем файле заносятся в список сборщика. При надлежащей буферизации обмена с этим файлом можно добиться приемлемого количества обменов с внешним накопителем.

Такой способ и был реализован на ЭВМ БЭСМ-6 и ICL 1906A, причём он пригоден для оптимизации работы программы-сборщика независимо от структуры РАМ-файла.

Для РАМ-файла прямого доступа возможен ещё один способ оптимизации. Он состоит в том, что все директивы, содержащиеся в РАМ-файле, раскодируются во время записи его на диск. При этом создается специальный каталог, аналогичный списку сборщика, который и используется в дальнейшей работе, а обращение к РАМ-файлу прямого доступа будет происходить только при необходимости переписки из него карт в выходные файлы. Это должно дать существенную экономию времени работы програм-

мы-сборщика, так как оно тратится, в основном, на декодировку директив и занесение соответствующих блоков в список сборки.

## Б. ЗАКЛЮЧЕНИЕ

Описанная программа-сборщик для РАМ-файла прямого доступа была реализована на ЭВМ ICL 1906A (и частично на ЭВМ БЭСМ-6). Способ обращения к ней приведен в Приложении 3.

Реализованная программа имеет прямую моду адресации (15AM, DEM) и работает в 7-10 раз быстрее обычной версии PATCHY, использующей последовательный РАМ-файл. Полученные параметры программы позволяют эффективно использовать её при работе в режиме мультидоступа.

В заключение авторы считают своим долгом выразить благодарность группе системного программирования ЭВМ ICL за стимулирование настоящей работы, а Ю.Л. Петровых и Е.А. Слободок - за поддержку и проявленный интерес.

## ЛИТЕРАТУРА

1. E. Pagiola. Program PATCHY, CERN Computer 6000 Series Program Library II, CERN, Geneva, 1968;  
Л.Г.Каминский, С.В.Клименко, С.В.Тарасевич. Препринт ИФВЭ 73-8, Серпухов, 1973.
2. Program UPPA, CERN Computer 6000 Series Program Library II, CERN, Geneva, 1968;  
Л.Г.Каминский, С.В.Клименко, В.М.Кубик. Препринт ИФВЭ 73-88, Серпухов, 1973.
3. 6000 Series Computer Services. CERN, Geneva, 1970;  
CERN Computer 6000 Series Scope General Reference. CERN, Geneva, 1969.  
Г.Сакман. Решение задач в системе человек-ЭВМ, М., "Мир", 1973.
4. Г.Л.Мазный. Мониторная система ДУБНА, ОИЯИ, 11-5974, Дубна, 1972.
5. ICL 1900 Series Technical Publication, London, 1971.  
"Супервизоры и операционные системы". Сборник под редакцией Дж. Каттля и П. Робинсона. Перевод с английского, М., "Мир", 1972.

Рукопись поступила в издательскую группу  
28 июня 1974 года.

Приложение 1

**ФОРМАТ ОБЩЕГО СПИСКА ПРОГРАММЫ-СБОРЩИКА**

**A. Начало списка.**

**Основные параметры.**

**LIST (1) LFREE**

- адрес первого свободного слова списка,

**(2) NP**

- количество редактируемых петчей,

**(3) IP**

- адрес первого блока петча,

**(4) NPATCH**

- адрес специального имени в таблице  
идентификаторов,

**(5) NZ**

- количество последовательностей,

**(6) IZ**

- адрес первого блока последователь-  
ностей,

**(7) NINDR**

- адрес специального идентификатора  
даты (DATEQQ) в таблице,

**(8)**

- начало первого блока последователь-  
ностей

...

**B. Конец списка сборщика, содержащий отчет о логике выбора петчей и  
соответствующего режима вывода.**

## Обозначения

LMAX		- адрес последнего слова списка,
NPATCH		- адрес имени выбранного петча в таблице идентификаторов,
LFROM		- адрес в таблице имени петча, откуда произведен соответствующий выбор,
IFORT		- режим вывода,
NP		- количество выбранных петчей,
LIST	(LMAX )	NP
	(LMAX-1 )	LFROM <sup>1</sup> , IFORT <sup>1</sup>
	(LMAX-2 )	NPATCH <sup>1</sup>
	...	
	(LMAX-2·NP+1)	LFROM <sup>NP</sup> , IFORT <sup>NP</sup>
	(LMAX-2·NP )	NPATCH <sup>NP</sup>

С. Структура блока корректов в списке сборщика.

LIST	(IP )	IP	- адрес следующего блока петча,
	+1	NPATCH	- адрес имени петча в таблице идентификаторов,
	+2	ND	- количество редактируемых колод,
LIST	(IP+3 )	ID	- адрес следующего блока колоды,
	+4	NDECK	- адрес имени колоды в таблице идентификаторов,
	+5	NC	- количество карт корректов в колоде,
LIST	(IP+6 )	II0	- адрес следующего блока карты,
	+7	IACG1	- код операции редактирования,
	+8	NCARD	- начало действия корректа,
	+9	NCONT	- конец действия корректа,
	+10	NINDR	- адрес имени последовательности в таблице идентификаторов,

+11	LEVEL	- уровень редактирования,
+12	COR, IPORT	- название корректа, режим вывода,
+13	NP CUR	- адрес источника карты корректа (петч, колода, карта),
+14	NDCUR	
+15	NCCUR	
+16		- рабочая ячейка,
+17		- тексты данных
...		карты корректа *) .

## Приложение 2

### КАТАЛОГ РАМ-ФАЙЛА ПРЯМОГО ДОСТУПА

#### A. Блок петча

- 2	LD	- связь к первому блоку колоды,
- 1	LP	- связь к следующему блоку петча,
L + 0	NAME	- имя петча,
+ 1		
+ 2	NCARD	- количество карт в бланковой колоде,
+ 3	NREC, LCARD	- адрес петча на диске.

#### B. Блок колоды

- 1	LD	- связь к следующей колоде,
L + 0	NAME	- имя колоды,
+ 1		

\*) Для РАМ-файла прямого доступа здесь записываются адреса карт в рабочем файле.

- |     |             |                             |
|-----|-------------|-----------------------------|
| + 2 | NCARD       | - количество карт в колоде, |
| + 3 | NREC, LCARD | - адрес начала колоды.      |

Здесь NREC - номер записи на диске, LCARD - адрес в этой записи.

### Приложение 3

Система хранения и редактирования текстов PITCHU на ICL 1906A состоит из трех отдельных программ:

- |        |  |
|--------|--|
| # SPAT | - программа-сборщик;   |
| # RPAT | - программа переписки RAM-файла из дискового файла прямого доступа в последовательный файл общего хранилища; |
| # WPAT | - программа создания RAM-файла в моде прямого доступа с каталогом.   |

Программы #SPAT и #RPAT предназначены для пользователей, а # WPAT эксплуатируется ответственным системным программистом, обслуживающим личные файлы на дисках.

Программа-сборщик работает под контролем макрокоманды SPAT. Обращение к ней имеет вид

SPAT CRADLE

где CRADLE - имя файла, в котором находится колода пользователя. Выходные файлы имеют стандартные имена:

SCRATCH - для собранной программы,

PLIST - для распечатки собранной программы с картиной сборки.

Адрес RAM-файла задается в колоде пользователя директивой

+ FILE, RAM = file description,

где file description - описание файла в формате операционной систе-

мы GEORGE . Например, для указания РАМ-файла HYDRAPAM , находящегося на диске 770023 в режиме прямого доступа, эта карта имеет вид

+ FILE, RAM = (770023, HYDRAPAM).

Для работы с последовательным РАМ-файлом из общего хранилища файлов эта карта имеет вид

+ FILE, RAM = :USER. GEOMRAM .

Следует отметить, что на +FILE карте запрещается использовать комментарии, в отличие от стандартного формата директив системы PATCHY.

Для работы с программой #RPAT используется следующая последовательность команд системы GEORGE :

FIND # RPAT

EN 0, (770023, HYDRAPAM)

Здесь РАМ-файл будет переписан с диска в файл общего хранилища с тем же именем HYDRAPAM . Если необходимо переписать текст в файл с другим именем (SERIALPAM) , то в качестве второго параметра указывается это имя:

EN 0, (770023, HYDRAPAM) , SERIALPAM

Цена 8 коп.

© - Институт физики высоких энергий, 1974.

Издательская группа И Ф В Э

Заказ 427. Тираж 290. 0,7 уч.-изд.л. Т-08368.

Июль 1974. Редактор Н.В. Ежела.