

сообщения
Объединенного
Института
Ядерных
Исследований
Дубна

3377/82

19/4-82

P11-82-289

+

В.С.Гончаков, В.В.Кореньков

ВНУТРЕННЯЯ СТРУКТУРА НАБОРНОГО ФАЙЛА
ДИАЛоговой СИСТЕМЫ TERM ЕС ЭВМ

1982

Введение

- Диалоговая система TERM предоставляет пользователям широкие возможности доступа к ЭВМ и средствам операционной системы с помощью терминалов.

Главными требованиями для современных диалоговых систем являются следующие:

1. Язык общения пользователя с системой должен быть простым, гибким, удобным для работы, но обладающим широкими возможностями контекстного редактирования.

2. Время ответа на большинство команд пользователя должно находиться в пределах 3-5 секунд для относительно небольшого числа одновременно работающих терминалов (порядка 10).

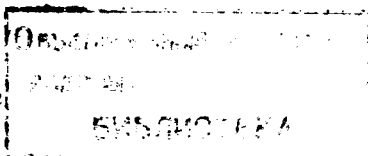
3. Должен быть реализован эффективный рестарт (перезызов) системы в аварийных ситуациях.

4. Необходимо минимизировать расход оперативной памяти.

Эти требования находятся в видимом противоречии друг с другом, поэтому, чтобы создать диалоговую систему, максимально удовлетворяющую этим условиям, нужно особенно тщательно продумать структуру наборного файла, над которым производятся все операции редактирования.

Если в оперативной памяти хранить много информации, то можно обеспечить хорошую реактивность системы, однако при этом существенно увеличивается расход оперативной памяти и делается затруднительным рестарт системы после машинных и системных ошибок.

Если всю информацию о наборном файле хранить на дисках, то замедляется реакция системы и невозможно обеспечить удовлетворительное время ответа на команды пользователя. О попытке найти "золотую середину" и пойдет речь в данной работе.



Внутренняя структура наборного файла

Прежде чем описывать внутреннюю структуру наборного файла, определим это понятие.

Наборный файл - временный оперативный рабочий файл, который предназначен для приема, накопления и редактирования различных текстов.

Каждому пользователю, работающему за терминалом, отводится свой наборный файл, который уничтожается по окончании сеанса.

В диалоговой системе TERM наборный файл состоит из листов фиксированной длины. Каждый лист находится или в оперативной памяти, или на диске в качестве раздела библиотечного набора данных с определенным именем.

Алгоритм образования имен имеет существенное значение для восстановления информации после машинных и программных ошибок.

Каждый лист состоит из записей фиксированной длины, связанных друг с другом ссылками. В качестве ссылки используется относительный адрес следующей записи. Все записи внутри листа имеют списковую структуру и поэтому легко редактируются путем замены ссылок и добавления новых записей, если лист еще не заполнен до конца.

В работе (3) содержатся описание различных структур данных, в том числе и их списковой организации, и алгоритмы работы с ними. С каждым листом наборного файла связан элемент каталога, который находится в оперативной памяти и имеет следующую структуру:

NEXTADR	ADRORNAME	MINUMBER	COUNTER
---------	-----------	----------	---------

NEXTADR - адрес следующего элемента каталога, описывающего лист наборного файла. Для последнего элемента каталога в первом байте этого поля стоит X'EE'.

ADRORNAME - адрес листа наборного файла в оперативной памяти или имя раздела, если данный лист находится на диске в библиотечном наборе данных.

MINUMBER - минимальный номер записи, находящейся на данном листе.

COUNTER - реальное количество используемых записей в данном листе.

Элемент каталога, занимая всего 16 байтов, достаточно хорошо описывает каждый лист наборного файла.

Элементы каталога наборного файла составляют список, находящийся в оперативной памяти.

Получив команду редактирования, по списку элементов каталога легко найти лист наборного файла, с которым надо работать. Этот поиск практически не зависит от длины наборного файла. Подробное описание команд находится в работах ^{1/}, ^{2/}, а форматы команд - в Приложении I.

При уничтожении наборного файла или определенной его части освободившиеся листы переходят в списки свободных листов оперативной памяти или в списки свободных разделов.

Наоборот, если создается новый наборный файл или происходит добавление записи в заполненный лист, то новые листы для этого захватываются из списка свободных листов оперативной памяти или из списка свободных разделов.

С каждым свободным листом оперативной памяти и свободным разделом связаны элементы каталога, которые образуют список, находящийся в оперативной памяти.

При таком способе организации наборного файла в оперативной памяти хранится небольшое количество ключевой информации, что обеспечивает быстрый поиск нужных записей и обеспечивает хорошее время ответа на большинство команд пользователя.

Возникает вопрос, каким образом восстанавливается эта ключевая информация при возникновении машинных и системных ошибок и при перезапуске диалоговой системы?

Восстановление информации

Большое внимание при разработке диалоговой системы уделялось восстановлению утерянной информации при машинных и системных ошибках.

В момент возникновения аварийной ситуации на машине пропадает информация, находящаяся в оперативной памяти ЭВМ. К этой информации относятся информационные поля терминалов, каталоги наборных файлов, списки свободных листов оперативной памяти и свободных разделов, содержимое листов оперативной памяти.

Для восстановления информации придуман специальный алгоритм образования имен разделов в библиотечном наборе данных наборного файла.

Имя раздела состоит из индекса пользователя и номера. Свободные разделы имеют свой индекс.

При восстановлении информации читается очередной раздел и по его индексу определяется, свободен этот лист или принадлежит определенному наборному файлу.

Если лист принадлежит наборному файлу определенного пользователя, то после анализа информации внутри листа восстанавливаются поля MINUMBER и COUNTER в элементе каталога наборного файла.

Таким образом, один элемент каталога восстановлен, кроме поля NEXTADR. Когда восстанавливаются все элементы каталога, принадлежащие определенному пользователю, надо упорядочить их по возрастанию MINUMBER и таким образом связать их в список, заполняя поля NEXTADR. Эту процедуру надо проделать для всех наборных файлов с определенными индексами, а также для списка свободных листов.

Листы оперативной памяти восстанавливаются аналогично, так как спасаются в разделах библиотеки наборного файла с определенными именами в момент изменения информации внутри листа.

Таким образом, вся утерянная информация восстановлена. На экранах терминалов появляется сообщение:

RESTART EXECUTED

и пользователь может продолжать работу.

Заключение

Есть возможность полностью отказаться от листов наборного файла в оперативной памяти, что приведет к некоторому увеличению времени ответа на команды редактирования (особенно для файлов длиной до 50 записей), но уменьшит расход оперативной памяти.

Количество листов оперативной памяти определяется при запуске диалоговой системы TERM, что является очень важным для ЭВМ с небольшим объемом оперативной памяти (до 512К).

ПРИЛОЖЕНИЕ I

Формат команд диалоговой системы TERM

```

LOGIN
LOGOUT
FILES
AUDIT,libname
STORE,filename,libname
FETCH,filename,libname
DISCARD,filename,libname
RETURN,filename
BATCH,filename {, { INPUT [,MINE] } }
                {, { LOCAL
                { PRINT
Q, { I [,class] }
   { E
   { O [,class] }
MESSAGE, text
PULT, команда оператора

```

Команды редактирования

```

CREATE [,line [,incr]]
ADD [,line [,incr]]
RESEQ [,line [,incr]]
    line = text
LIST, { ALL
       line 1 [, { line 2 } ] } {, /text/ [ (col1 [,col2]) ] [ ,UNIT ] [ ,SUP ]
       LAST
DELETE, { ALL
        line1 [, { line2 } ] } {, /text/ [ (col1 [,col2]) ] [ ,UNIT ]
        LAST
/text1/=/text2/, { ALL
                  line1 [, { line2 } ] } { (col1 [,col2]) ] [ ,UNIT ]
                  LAST
SAVE, filename [, NOSEQ] [,MERGE]
EDIT, filename [,SEQUENCE]
RUN, { FG
      FH
      AS
      AL
      PL
      CO } , FILE=filename [,NOEX]

```

Команды оператора

```

SHOW
SEND { *
      адрес } text

```

Литература

1. Калмыкова Л.А., Шириков В.П. ОИИИ Б1-11-10705, Дубна, 1977.
2. INTERCOM REFERENCE MANUAL, VERSION 4 (60494600), ST.Paul, Minnesota (USA).
3. Кнут Д. Искусство программирования для ЭВМ. Т.1, "Мир", Москва, 1976.

Рукопись поступила в издательский отдел
16 апреля 1982 года.