

сообщения
объединенного
института
ядерных
исследований
дубна

11-84-325

В.В.Кореньков

ПРОГРАММНЫЕ СРЕДСТВА
ДЛЯ СОЗДАНИЯ МНОГОТЕРМИНАЛЬНЫХ ЗАДАЧ
НА ЕС ЭВМ

1984

Многотерминальные задачи

Диалоговая система ТЕРМ предоставляет пользователям терминалов возможность работать в режиме дистанционной пакетной обработки и в режиме интерактивного взаимодействия с выполняющимися задачами.

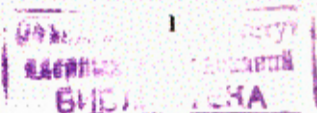
Для запуска задания в режиме пакетной обработки существует директива BATCH, а для интерактивной работы - директивы RUN и EXEC. Подробная информация об интерактивном режиме работы содержится в [1], а здесь коротко объясняется смысл этого режима.

В интерактивном режиме система интерпретирует операторы типа READ и WRITE как операторы обмена с терминалом, что очень важно для некоторых классов задач. Достоинством этого способа является то, что не требуется изменить текст программы при переходе от режима пакетной обработки к режиму диалога. В этом случае задача связывается и работает с тем терминалом, с которого была запущена. Такие задачи называются **однотерминальными**. В некоторых случаях бывает необходимо организовать диалог задачи с несколькими терминалами. Задачи, интерактивно взаимодействующие с несколькими терминалами, называются **многотерминальными**.

Многие системные программы, обслуживающие терминалы, являются многотерминальными. К ним, в частности, относятся EDITOR в терминальной системе INTERCOM на CDC-6500 [2], программа связи на БЭСМ-6 [3], диалоговая система ТЕРМ [1] и др. Однако аппарат для организации многотерминальных задач не всегда доступен пользователям, а для некоторых задач это было бы очень полезно. Можно привести несколько примеров таких задач.

1. Диалоговые, справочные, информационно-поисковые системы, которые одновременно вызываются с нескольких терминалов. Можно, конечно, работать с несколькими копиями программ, каждая из которых связана с одним терминалом, но в этом режиме чаще всего неэкономно расходуются ресурсы ЭВМ (оперативная память, центральный процессор).

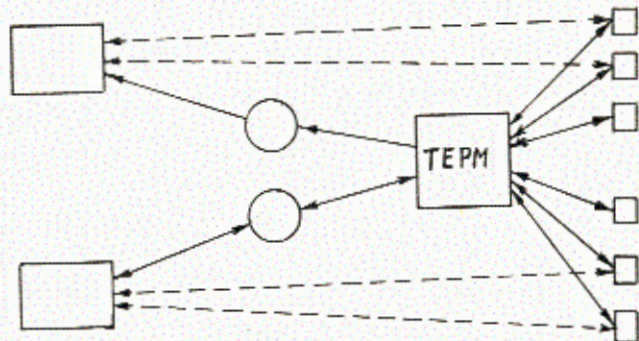
2. Диалоговые информационные системы, использующие для работы общую базу данных для просмотра и редактирования различных областей. В режиме работы с несколькими копиями программ возникает проблема синхронизации доступа к базе данных.



3. Системы, которые допускают взаимодействие пользователей, работающих за терминалами, как друг с другом, так и с управляющей программой (игровые программы, программы проектирования и т.д.).

Монитор для многотерминальных задач

Взаимодействие многотерминальной задачи с терминалами осуществляется в рамках диалоговой системы TERM через специальный монитор.



На рисунке представлена схема взаимодействия двух многотерминальных задач с двумя группами терминалов.

Запуск многотерминальной задачи осуществляется по директиве

`XEQ MULTIXXX` ,

где **xxx** - любые три символа, идентифицирующие многотерминальную задачу, которая должна находиться в системной или личной библиотеке. Если задача с таким именем не была запущена, то система активизирует монитор и сообщает ему имя задачи.

Монитор выполняет следующие функции:

- определяет адреса подключенных к нему терминалов;
- запускает многотерминальную задачу;
- буферизует и адресует сообщения между многотерминальной задачей и терминалами;
- синхронизирует работу со всеми терминалами;
- отключает терминалы от задачи;
- оканчивает работу задачи и сообщает диалоговой системе о завершении своей работы.

Каждую многотерминальную задачу сопровождает свой монитор.

Если пользователь запросил выполнение многотерминальной задачи, а она уже запущена, то об этом сообщается монитору, и он подключает указанный терминал к этой задаче. На терминале появляется сообщение:

* ТЕРМИНАЛ ПОДКЛЮЧЕН К ЗАДАЧЕ <имя>

где <имя> - название многотерминальной задачи. После этого задача может работать и с новым подключенным терминалом. Для отключения терминала от многотерминальной задачи существует директива % A.

Когда последний терминал отключается от задачи, то монитор оканчивает работу задачи и сообщает диалоговой системе о завершении своей работы.

Программные средства для создания многотерминальных задач

Для организации многотерминальной задачи существует одна системная подпрограмма `MULSP`, которая осуществляет связь с монитором. Эту системную подпрограмму можно вызывать из программы, написанной на любом языке программирования. Обращение к `MULSP` из программы на фортране выглядит так:

`CALL MULSP (IA, IL, IFUNK, INUM, IERR)` ,

- где **IA** - адрес массива, откуда будет выдаваться или приниматься сообщение с терминала;
- IL** - максимальная длина принимаемого или выдаваемого сообщения в байтах;
- IFUNK** - целое число, определяющее номер функции, которую надо выполнить:
- 1 - прочитать сообщение с терминала;
 - 2 - выдать сообщение на терминал;
 - 3 - стереть экран и выдать сообщение на терминал;
 - 4 - прочитать содержимое всего экрана;
- INUM** - переменная с целым значением, которое указывает относительный номер терминала, на который посылается или от которого принимается сообщение (1-254).

Если при выдаче указано число 255 или 'FF', то сообщение посылается на все подключенные к задаче терминалы. Если мы хотим дать запрос на чтение от всех терминалов, то тоже заносим число 255. В любом случае после чтения в этом слове `INUM` будет указан относительный номер терминала, с которого пришло сообщение.

IERR - это слово служит для отметки о результатах работы подпрограммы. Если после обращения к подпрограмме содержимое этого слова равно 0, то работа завершилась успешно;

Если содержимое слова равно 8, то это означает, что в параметрах при обращении к MULCP допущены ошибки; если содержимое слова равно 12, то произошла ошибка при обмене с терминалом.

Пример:

```
DIMENSION IA(40)
1  IL = 160
   IFUNK=2
   INUM=255
   IERR=0
   CALL MULCP(IA,IL,IFUNK,INUM,IERR)
   IF(IERR.NE.0) GO TO 2
   . . . . .
   IFUNK=1
   CALL MULCP(IA,IL,IFUNK,INUM,IERR)
   IF (IERR.NE.0) GO TO 2
   GO TO 1
2  CONTINUE
   . . . . .
```

В этом примере дается запрос на чтение со всех терминалов, которые подключены к данной задаче. Как только первый из пользователей этой задачи нажал на клавишу "ввод", его сообщение пересылается в массив IA, а номер терминала в INUM. Далее можно делать анализ принятого сообщения и какие-то преобразования, а потом выдать на этот терминал сообщение и снова дать запрос на чтение со всех терминалов.

Описанный аппарат максимально упрощает создание многотерминальных задач.

Литература

1. Кореньков В.В., ОИЯИ, II - 64 - 216, Дубна, 1984.
2. Шириков В.П. ОИЯИ, БИ-II-12603, Дубна, 1979.
3. Каданцев С.Г. ОИЯИ, II-82-796, Дубна, 1982.

Рукопись поступила в издательский отдел
II мая 1984 года.

Кореньков В.В. 11-84-325
Программные средства для создания многотерминальных задач

Дано описание системных средств для организации многотерминальных задач, выполняющихся на уровне задач пользователя. Эти средства /монитор и системные подпрограммы/ предназначены для работы в диалоговой системе TERM и позволяют максимально упростить создание многотерминальных задач.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1984

Перевод О.С.Виноградовой

Korenkov V.V. 11-84-325
Programming Techniques for Multiuser Jobs on the BS Computer

System programming techniques for multiuser jobs are described. Multiuser jobs are constructed on the user job level. These techniques (monitor and system program) are intended to use in TERM conversational system and permit to simplify maximally the creation of multiuser jobs.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1984