

сообщения
Объединенного
института
ядерных
исследований
Дубна

11-84-324

В.С.Гончаков, В.В.Кореньков

СТРУКТУРА СИСТЕМНЫХ ОЧЕРЕДЕЙ ЗАДАНИЙ
И РАБОТА С НИМИ
В ДИАЛОГОВОЙ СИСТЕМЕ ТЕРМ НА ЕС ЭВМ

1984

Введение

Диалоговая система ТЕРМ предназначена для облегчения доступа пользователей к ЭВМ и средствам операционной системы с помощью терминалов.

ТЕРМ - универсальная система, предоставляющая широкий спектр возможностей по редактированию текстов, запуску задач и процедур, работе с очередями, диалогу с оператором, слежению за задачами и состоянием операционной системы, работе с интерактивными и много-терминальными задачами (см. /3-5/ и приложения).

Большое место в диалоговой системе ТЕРМ занимает работа с системными очередями (постановка заданий во входную очередь, просмотр очередей, возврат листинга из выходной очереди для просмотра на экране терминала).

В этой работе рассказывается об очередях заданий с той степенью детализации, которая необходима, чтобы понять работу программ системы ТЕРМ, реализующих команды работы с очередями. Краткая информация об очередях заданий есть в /1-2/. Затем в общих чертах описана реализация команд работы с очередями в системе ТЕРМ.

Структура системных очередей заданий

Рабочие очереди операционной системы располагаются в наборе данных SYS1.SYSJOBQE. Они позволяют запоминать работу в порядке поступления и выполнять ее в приоритетном порядке. Приоритет обработки задается параметром PRTY карты JOB. Рабочие очереди выполняют роль буферов между программами системного ввода (RDR), инициаторами заданий (INIT) и программами системного вывода (WTR), позволяя им работать с максимальной скоростью. Отметим в общих чертах, каким же образом используются очереди этими системными программами.

Задания поступают в систему во входном потоке. Они обрабатываются программой системного ввода, которая на основе поступивших операторов языка управления заданиями (JCL - операторы), описывающих очередное задание, строит управляющие таблицы. Эти таблицы программы RDR записывают в набор данных SYS1.SYSJOBQE, и из них строится вход во входной очереди для задания. Одновременно программа RDR создает для этого задания вход в выходной очереди, соответствующей параметру MSGCLASS карты JOB, куда она записывает сообщения операционной системы для программиста. Закончив обработку задания, RDR ставит построенный вход во входную очередь, соответствующую классу, указанному параметром CLASS карты JOB, с приоритетом, заданным для этого задания. Программа, которая ставит вход в очередь, проверяет, была ли очередь пуста, и если это так, то она в конце

своей работы делает отметку всем задачам, обрабатываемым эту очередь, сообщая им, что для них появилась работа. Входы во входные очереди обрабатываются инициаторами, которые планируют дальнейшие прохождения задания через систему. После того, как завершится последний шаг задания, вход во входную очередь уничтожается, затем завершаются и ставятся в очереди входы выходных очередей для этого задания. Входы выходных очередей обрабатываются программами системного вывода. Эти программы распаковывают сообщения операционной системы, преобразуют записи файлов системного вывода в формат, соответствующий выходному устройству, и выводят их на выходное устройство. После обработки входа выходной очереди программа системного вывода уничтожает этот вход.

Набор данных SYS1.SYSJOBQDS размещается на постоянно смонтированном томе прямого доступа. На этом томе он должен занимать непрерывный участок памяти, который ему отводится при генерации системы.

Когда набор данных очереди инициализируется, он делится на две области. Первая из них – область управляющих записей, которая занимает первые 2736 байт набора данных очереди и содержит 36-байтовые управляющие записи очереди (QCR) для каждой из 76 очередей.

Структура набора данных очереди после инициализации показана на рис. 1.

Первая из QCR называется главной QCR (MQCR). Она содержит информацию об очереди в целом и, кроме того, используется для поддержки очереди свободных дорожек. Для справок об очереди система хранит ее текущую копию в ядре.

Формат MQCR описан в таблице I. В этой таблице и в последующих имеют следующие колонки:

Смещение – определяет смещение описываемого поля относительно начала блока или записи. Смещение задается в десятичном виде и в шестнадцатеричном.

Длина – определяет длину поля. Если поле не начинается на границе слова, то появляются точки перед числом, указывающим длину.

Каждая из остальных QCR используется для управления определенной очередью. Из 76 очередей 56 используются системой, остальные очереди зарезервированы.

Система использует следующие очереди:

15 входных очередей, по одной для каждого класса ввода
(от А до 0)

36 выходных очередей, по одной для каждого класса вывода
(от А до Z и от 0 до 9)

Очередь задержанных заданий, которая содержит входы из входных очередей в состоянии задержки (в результате команды HOLD).

Таблица I

Смещение Дес.	Шестн.	Длина	Описание поля
0	0	8	адрес (МВВССНПР) главной QCR
8	8	1	резерв
9	9	.2	указатель на первую логическую дорожку в очереди свободных дорожек
11	B	...1	резерв
12	C	2	число логических дорожек в наборе данных
14	E	..2	число свободных дорожек в очереди
16	10	2	число логических дорожек для завершения задания
18	12	..2	число логических дорожек для инициаторов
20	14	2	указатель на последнюю логическую дорожку в очереди свободных дорожек
22	16	..2	адрес (ТТ) первой физической дорожки, на которой нет QCR
24	18	2	число QCR на физической дорожке
26	1A	..2	число записей на физической дорожке
28	1C	2	число 176-байтовых записей на логической дорожке
30	1E	..2	число логических дорожек, резервированных для одного инициатора
32	20	2	число QCR на смешанной дорожке
34	22	..2	указатель на первую запись на первой дорожке, не содержащей QCR

Очередь свободных дорожек, которая содержит дорожки, не назначенные ни одному входу очереди.

Очередь удаленного ввода заданий (RJE)

Очередь программы автоматического пакетирования системного ввода (ASB), которая содержит JCL – предложения в сжатом формате, используемом как вход для интерпретатора.

Очередь программы ввода заданий, используемой в TSO, которая содержит указатели на наборы данных для заданий TSO, запущенных через команду SUBMIT.

Для каждой из очередей (кроме очереди свободных дорожек, которая

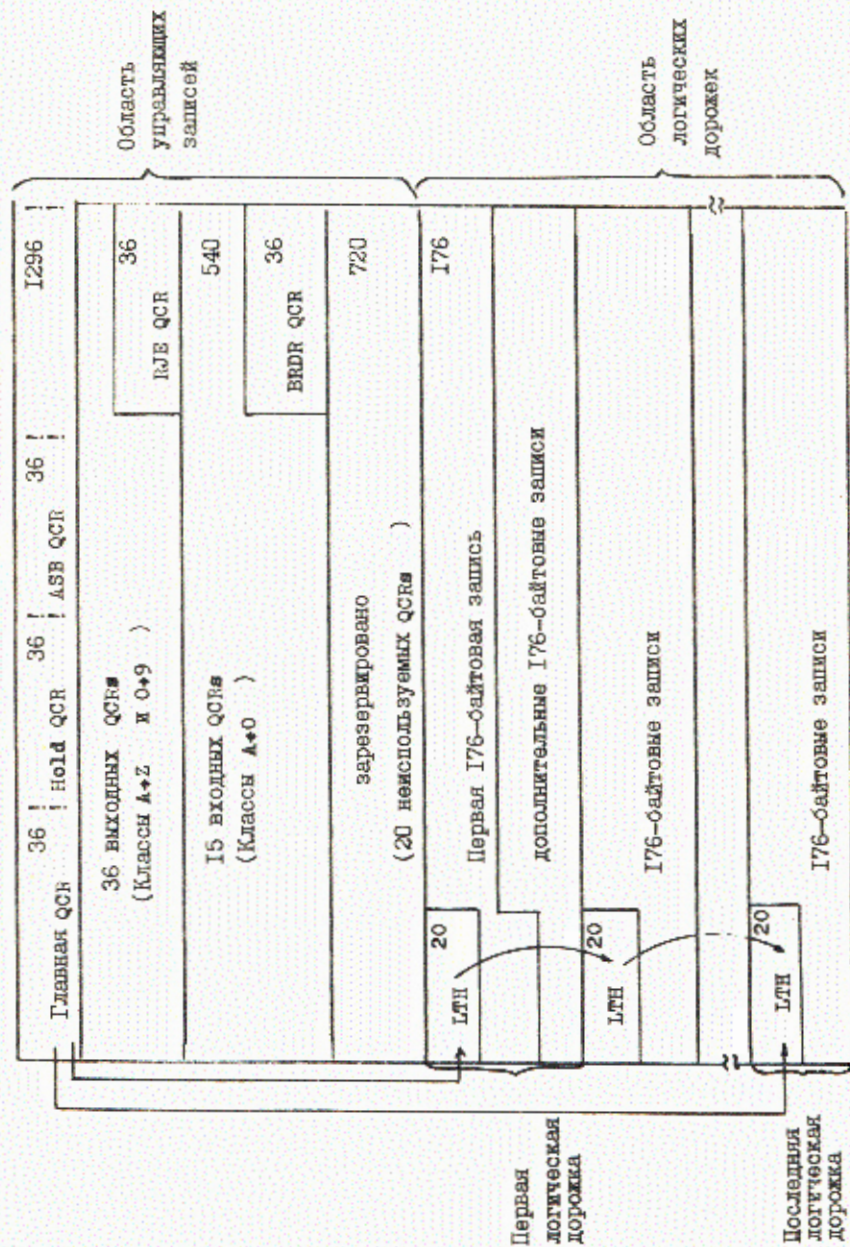


Рис. 1. Набор данных очереди заданий после инициализации

Замечание: После инициализации набора данных очереди все логические дорожки принадлежат очереди свободных дорожек, которая поддерживается через MQCR. Остальные QCR - нулевые.

поддерживается через MQCR) используется своя QCR. Ее положение в области управляющих записей полностью определяется типом соответствующей очереди.

Вторая из основных областей - это область логических дорожек. Логическая дорожка - это непрерывная область очереди (ей не обязательно соответствует физическая дорожка). Первая ее запись имеет длину 20 байтов и называется заголовком логической дорожки (LTN). LTN служит для связи логических дорожек между собой в различных очередях. Вслед за LTN следует некоторое число N 176-байтовых записей. В этих записях система хранит управляющие таблицы, описывающую вход в очередь. Число N одинаково для всех логических дорожек. Оно задается при генерации системы, и его можно изменить при загрузке системы в ответ на запрос о желаемых параметрах очереди. Оно должно быть в пределах от 10 до 255. Указатели от QCR к LTN и между записями LTN поддерживаются в форме NN. Это 2-х байтовое число, представляющее номер соответствующей записи относительно начала области логических дорожек. NN для первой записи области логических дорожек (это первая LTN очереди) равно 1, для второй - 2 и т.д. Форматы QCR и LTN приведены на таблицах 2 и 3.

Вход в очередь - это некоторое число логических дорожек, связанных между собой в последовательный список через поле LTN+12. От всех этих логических дорожек есть ссылка на первую логическую дорожку входа через поле LTN+9, кроме того, последние логические дорожки входов одной и той же очереди связаны в список, в котором они представлены по убыванию их приоритетов, через поле LTN+18. Поле LTN+14 в LTN последней логической дорожки входа показывает количество логических дорожек, назначенных этому входу. Из QCR соответствующей очереди через поле QCR+0 есть ссылка на самый верхний вход очереди. Это - основной вход в очередь, и через него в очередь обрабатываются в приоритетном порядке. Для того чтобы можно было легко добавлять в очередь новый вход некоторого приоритета, в QCR есть ссылки на последние входы в очередь соответствующих приоритетов. Структура типичной очереди показана на рис. 2.

Очередь свободных дорожек устроена более просто. Для LTN, описывающих логическую дорожку этой очереди, имеет значение только поле LTN+12, через которое они связаны в последовательный список. В MQCR система хранит количество дорожек в этом списке и ссылки на его первый и последний элемент.

Таблица 2. Формат QCR

Смещение Дес.	Шестн.	Длина	Описание поля
0	0	2	указатель (в форме NN) на последний LTN входа самого высокого приоритета, если входов в очереди нет, то 0.
2	2	..2	указатель (в форме NN) на последний LTN самого последнего входа, имеющего приоритет I4. Если входов приоритета I4 нет, то 0.
30	IE	..2	указатель (в форме NN) на последний LTN самого последнего входа, имеющего приоритет 0. Если входов приоритета 0 нет, то 0.
32	20	1	старший бит этого байта используется как признак задержки очереди, следующие 3 бита не используются. В младших 4-х битах записывается величина наибольшего приоритета входов очереди, если входов очереди нет, то там 0.
33	2I	.3	адрес BCB первой задачи, ожидающей работу с очередью.

Таблица 3. Формат LTN

Смещение Дес.	Шестн.	Длина	Описание поля
0	0	8	Имя задания
8	8	1	Код функции
9	9	..2	NN первой логической дорожки, назначенной этому входу
11	B	...1	Количество записей, назначенных на этой дорожке
12	C	2	NN следующей логической дорожки задания
14	E	..1	Количество назначенных логических дорожек
15	F	...1	Код типа задания
16	10	1	Код состояния задания
17	11	.1	Приоритет
18	12	..2	NN следующего входа в очередь

Так как очередь заданий может использоваться одновременно различными параллельно работающими задачами, то система разделяет доступ к ней через захват и освобождение ресурсов (макрокоманды ENQ и DEQ). Используется два системных ресурса. В обоих случаях имя очереди ресурсов SYSTEFSD и имена ресурсов Q2 - для защиты MQCR и Q1 - для защиты всех остальных QCR. Каждая программа, которая ссылается или модифицирует QCR выдает макрокоманду ENQ, задавая имя соответствующего ресурса, перед попыткой доступа к QCR и выдает макрокоманду DEQ, когда обработка QCR завершена.

Система работает с очередью заданий, используя метод доступа EXCR. В поле CVT+12 она хранит адрес секции IEFJOB, информацию из которой она использует при работе с очередью. Первые 52 байта секции IEFJOB занимает открытый DSB для очереди заданий, следующие 48 байтов занимает DEV для SYS1. SYSJOBQE, начиная с IEFJOB+100, 36 байтов занимает образ MQCR в памяти. В DEV для SYS1. SYSJOBQE в соответствующем поле проставлен ключ защиты, равный нулю, а супервизор ввода-вывода проверяет соответствие ключей защиты, поэтому через блоки, заданные в секции IEFJOB обмен с очередью могут вести только задачи, работающие с нулевым ключом защиты памяти.

Реализация команд работы с очередями в диалоговой системе TERP

Система TERP использует очередь заданий при реализации команд запуска программ в пакетном и интерактивном режимах, просмотра входных и выходных очередей и возврата листинга из выходной очереди на терминал. В командах для запуска программ (WATCH, XEQ, RUN) система TERP не работает с очередью заданий непосредственно. Она подготавливает блок STB (буфер информации о команде) для запуска процедуры RDRT по команде START, задав информацию о локальном файле и классе сообщений, и передает этот блок программе планирования команд главного планировщика (через SVC 34). Затем операционная система планирует запуск процедуры системного ввода, которая ставит задание во входную очередь.

При реализации остальных команд система TERP уже непосредственно работает с очередью заданий.

Команды просмотра входных и выходных очередей имеют следующий формат:

```
Q . I.[входной класс]
Q . O.[выходной класс]
```

По умолчанию входной класс - A и выходной - C. Совокупность параметров (тип очереди и класс) однозначно определяют

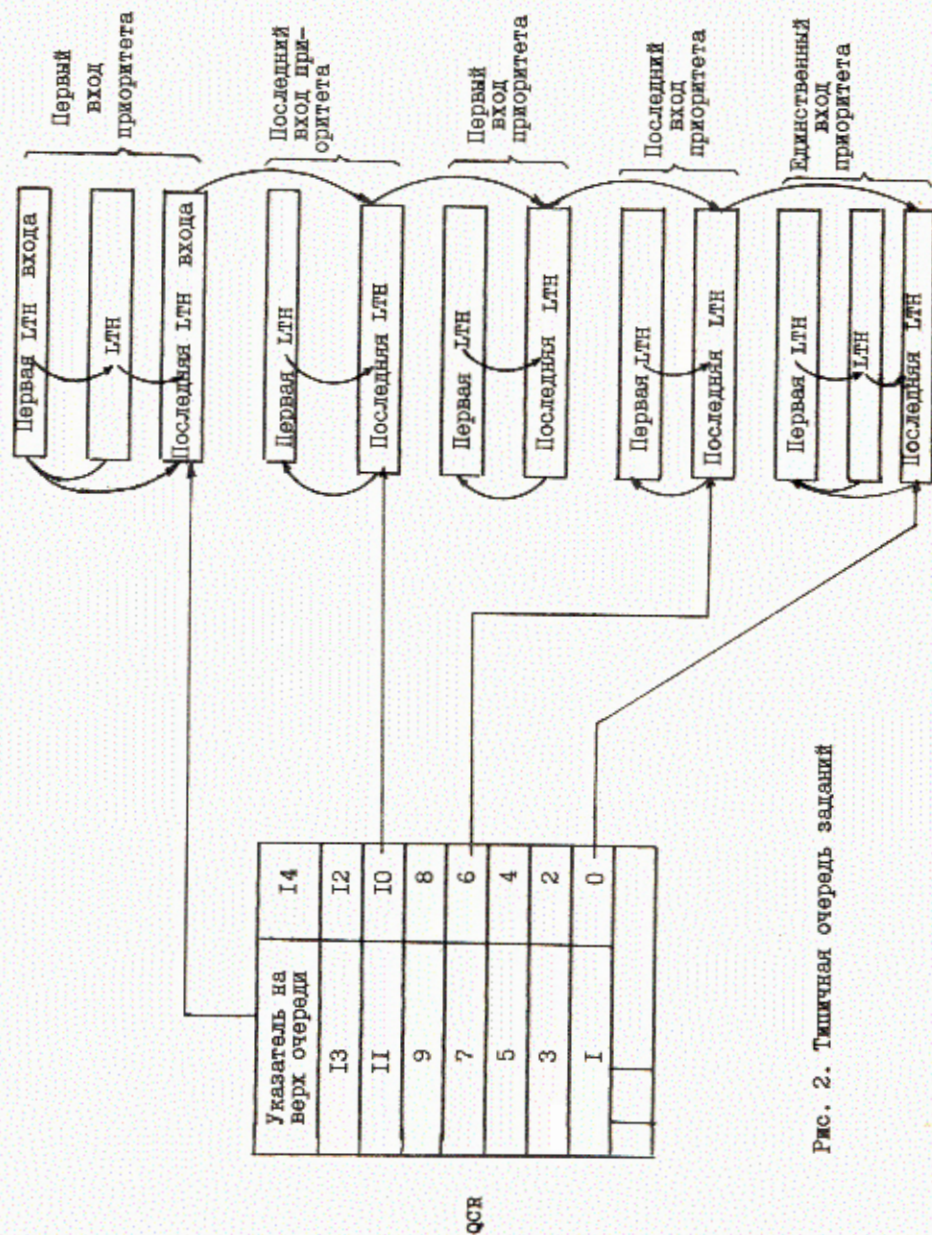


Рис. 2. Типичная очередь заданий

Замечание 1. На этом рисунке изображена типичная очередь заданий.

Она имеет два входа приоритета 10, 2 входа приоритета 6 и один вход приоритета 0. В QCR для этой очереди все поля ссылок, кроме полей ссылок для указателя на верх очереди и указателей на последние входы приоритетов 10, 6 и 0, нулевые.

Замечание 2. Для простоты на рисунке для каждой логической дорожки изображен только ее заголовок (запись LTH). В наборе данных очереди 176-байтовые записи, принадлежащие логической дорожке, расположены вслед за ее заголовком.

адрес QCR соответствующей очереди. Система TЕРМ захватывает ресурс Q1 читает соответствующую QCR, а далее просматривает весь список LTH, начиная с самого приоритетного (ссылка на него указана в QCR). Таким образом, можно узнать все задания, стоящие в очереди, и информацию о них вывести на экран терминала.

Когда задание завершается, можно посмотреть листинг на экране терминала. Для этого служит команда

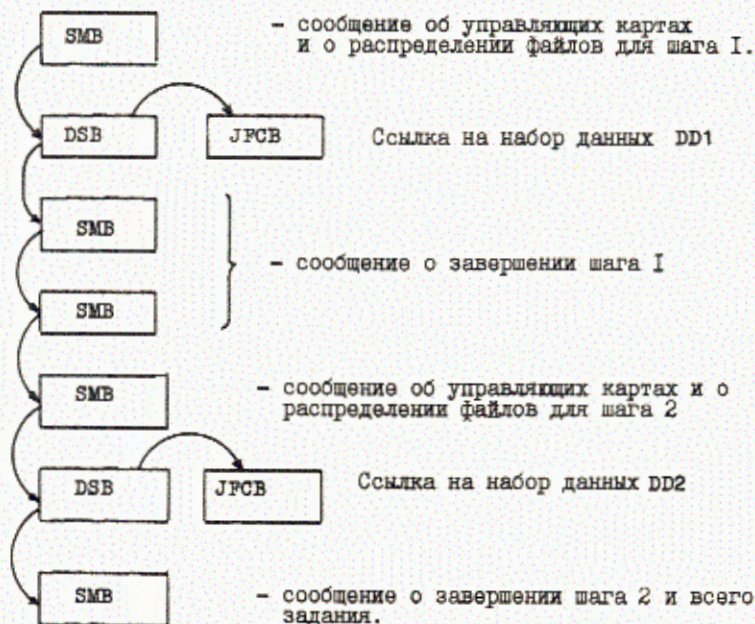
```
ВАТСН < имя задания > LOCAL [выходной класс]
```

По этой директиве листинг задания из выходной очереди указанного класса (по умолчанию с) переписывается в локальный файл, и его можно вывести на экран терминала с помощью команд PAGE или LIST, а в случае необходимости распечатать в машинном зале по команде

```
ВАТСН < имя задания > PRINT
```

Рассмотрим теперь, как реализована программа выполнения команды ВАТСН в режиме LOCAL. Она распаковывает и переписывает сообщения системы и наборы данных системного вывода в локальный файл с тем же именем, что и имя задания. Ее работа очень похожа на работу программы системного вывода (WTR). Элемент выходной очереди состоит из 176-байтовых блоков SMB (блок сообщений системы) и DSB (блок набора данных), связанных между собой в последовательный список. Структура типичного элемента выходной очереди показана на рис. 3. В блоке SMB в упакованном виде записаны сообщения системы. Блок DSB представляет собой ссылку на соответствующий набор данных системного вывода (через JCSB - блок управления файлом задания). Алгоритм работы программы, реализующей ВАТСН в режиме LOCAL, следующий.

Сначала программа захватывает нужный ресурс (q1), читает QCR для заданного выходного класса и просматривает эту очередь. Если задание с указанным именем в очереди не обнаружено, то на экран выдается диагностика и работа программы оканчивается.



```
//JOB1 JOB MSGCLASS=A
//STEP1 EXEC PGM=PGM1
//DD1 DD SYSOUT=A
//STEP2 EXEC PGM=PGM2
//DD2 DD SYSOUT=A
//
```

Если задание найдено в очереди, но начинает последовательно читать блоки из списка, проверяя тип блока.

Если это блок SMB, то распаковываем содержащиеся в нем сообщения системы и переписываем их в локальный файл. Если встретился блок DSB, то генерируем управляющую карту DD, исходя из информации, содержащейся в этом блоке (адрес UCS, JFCB и т.д.). Затем проверяем, существует ли этот набор данных на диске. Если существует, то открываем набор данных в режиме TYPE=J и переписываем его содержимое в локальный файл. По условию конца данных закрываем набор данных системного вывода и идем на обработку следующего блока.

Таким образом, блок за блоком обрабатывается весь список очереди. Когда все блоки обработаны, закрываем локальный файл, освобождаем ресурс Q1.

Теперь необходимо удалить это задание из выходной очереди. Для этого сначала вычеркиваем ссылку в QCR на это задание. Затем пере-

даем занимаемые этим заданием логические дорожки в очередь свободных дорожек и модифицируем MQCR. На этом работа программы, реализующей BATCH в режиме LOCAL завершается.

Приложение

Формат команд системы TERP

```
LOGIN
LOGOUT
FILES
AUDIT <имя файла> [V = <номер тома>] [ИМЯ]
FETCH <имя файла> [V = <номер тома>] [OVERWRITE]
STORE <имя файла> [V = <номер тома>] [OVERWRITE]
DISCARD <имя файла> [V = <номер тома>]
PAGE <имя файла> [V = <номер тома>]
RETURN <имя файла>
BATCH <имя файла> { {MINE
LOCAL [класс]
PRINT} }
IEQ <имя файла>
RUN <транслятор> FILE = <имя файла> [NOEX]
Q { I [,класс]
E
O [,класс] }
PULT <команда оператора>
MESSAGE <текст>
CONNECT <имя DD>
DISCONT <имя DD>
HELP
```

Команды редактирования

```
CREATE [[line [INCR]]]
ADD [[line [INCR]]]
RESEQ [[line [INCR]]]
line = text
UPDATE [line]
LIST { {ALL
LINE1 [ {LINE2} ]
LAST [ {LAST} ] } } [ /text/ [(col1 [col2])] [UNIT]] [SUP]
```

```

DELETE { ALL
        LINE1 [ { LINE2 } ]
        LAST } [ /text / (col1 [ col2 ] ) ] [ UNIT ]

/text1 /= /text2 / { ALL
                    LINE1 [ { LINE2 } ]
                    LAST } [ (col1 [ col2 ] ) ] [ UNIT ]

EDIT <имя файла> [ SEQUENCE ]

SAVE <имя файла> { ALL
                  LINE1 [ LINE2 ]
                  LAST } [ /text / (col1 [ col2 ] ) ] [ UNIT ]
                    [ MERGE ] [ NOSEQ ]

FORMAT { { <имя>
          [ TAB=C ] [ CN= <длина> ] [ N= <число> [ T1 [ T2 [ T3 ... ] ] ] ] } }
        [ SHOW ]

```

Команды оператора

```

SHOW
STOP
HOLD
GO
SEND { *
      индекс } <text>

```

Литература

1. Данилочкин В.П., Одинцов Б.В., Пеледов Г.В. Справочник системного программиста по операционной системе ОС ЕС. М., "Финансы и статистика", 1982.
2. Тимонин В.И. Операционная система ОС ЕС. Основы функционирования. М., "Финансы и статистика", 1983.
3. Гончаков В.С., Кореньков В.В. ПИ-82-289, Дубна, 1982.
4. Кореньков В.В. ПИ-82-290, Дубна, 1982.
5. Кореньков В.В. ПИ-82-291, Дубна, 1982.

Рукопись поступила в издательский отдел
11 мая 1984 года.

Гончаков В.С., Кореньков В.В. 11-84-324
Структура системных очередей заданий
и работа с ними в диалоговой системе TERM на ЕС ЭВМ

Дано описание структуры системных очередей заданий в операционной системе ОС ЕС ЭВМ. Описаны форматы команд работы с системными очередями в диалоговой системе TERM. Рассмотрена логика работы программ, реализующих эти команды.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1984

Перевод О.С.Виноградовой

Gonchakov V.S., Korenkov V.V. 11-84-324
System Job Queue Structure
in ES TERM Dialogue System

System job queue structure in OS ES computer operational system is described. The formats for commands, operating with system queues in TERM conversational system are described. The logic of operation of the programs realising these commands is considered.

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

Communication of the Joint Institute for Nuclear Research. Dubna 1984