

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ УКРАИНЫ
ИНСТИТУТ СИСТЕМНЫХ ИССЛЕДОВАНИЙ ОБРАЗОВАНИЯ
ХАРЬКОВСКИЙ ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ**

М. Ф. Бондаренко, Н. С. Лесная, И. Ю. Шубин

**ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ РЕХХ
В ПОДСИСТЕМЕ ДИАЛоговой
ОБРАБОТКИ СИСТЕМЫ
ВИРТУАЛЬНЫХ МАШИН ЕС ЭВМ**

Киев 1993

МИНИСТЕРСТВО ОБРАЗОВАНИЯ УКРАИНЫ
ИНСТИТУТ СИСТЕМНЫХ ИССЛЕДОВАНИЙ ОБРАЗОВАНИЯ
ХАРЬКОВСКИЙ ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ

М.Ф.Бондаренко, Н.С.Лесная, И.Ю.Шубин

ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ REXX
В ПОДСИСТЕМЕ ДИАЛОГОВОЙ ОБРАБОТКИ СИСТЕМЫ
ВИРТУАЛЬНЫХ МАШИН ЕС ЭВМ

Утверждено Институтом системных исследований
образования Украины в качестве
учебного пособия для студентов специальности
"Программное обеспечение вычислительной
техники и автоматизированных систем"

Киев 1993

УДК 681.3

Бондаренко М.Ф., Лісна Н.С., Щубін І.Ю. Основи програмування мовою **REXX** у підсистемі діалогової обробки системи віртуальних машин ЕС ЕОМ: Навч. посібник. - К.: ІСДЮ, 1998. - 116 с. - Рос.мовою.

Розглядаються нові принципи проектування програмного забезпечення ЕОМ типу ЕС. Викладено підхід до вивчення підсистеми діалогової системи віртуальних машин ЕС ЕОМ. Розглянуто мову програмування **REXX** - мову командних процедур. Дано типові завдання до лабораторних занять і для самостійної роботи студентів.

Посібник призначений для студентів спеціальності ПЗ ОТАС /22.04/, аспірантів, спеціалістів у галузі проектування програмного забезпечення та автоматизованих інформаційних систем.

Іл. 19. Бібліогр.: 4 назви

Рассматриваются новые принципы проектирования программного обеспечения ЭВМ типа ЕС. Изложен подход к изучению подсистемы диалоговой обработки системы виртуальных машин ЕС ЭВМ. Рассмотрен язык программирования **REXX** - язык командных процедур. Даны типовые задания к лабораторным занятиям и для самостоятельной работы студентов.

Пособие предназначено для студентов специальности ПО ВТАС /22.04/, аспирантов, специалистов в области проектирования программного обеспечения и автоматизированных информационных систем.

Ил. 19. Библиогр.: 4 назв.

Ответственный за выпуск М.Ф.Бондаренко, д-р техн.наук, проф.

Рецензенты: Евдокимов А.Г., д-р техн.наук, проф
Бузницкая Э.М., канд.техн.наук, доц.

ISBN 5-7763-2438-6

© М.Ф.Бондаренко, Н.С.Лесная, И.Ю.Щубин

ПРЕДИСЛОВИЕ

Широкое распространение персональной вычислительной техники трудно переоценить. Однако существует ряд классов реальных задач, которые не под силу персональным компьютерам или их использование становится малоэффективным, например:

расчет моделей сложных технических устройств (для моделирования замедляющих систем - на IBM PC - 3-4 ч , на ЕС ЭВМ - 3-5 мин)

создание вычислительных систем, обеспечивающих работу АСУ. Такие системы имеют иерархическую структуру: на нижнем уровне - персональные ЭВМ, на верхнем - ЕС ЭВМ;

создание реальных СУБД (для библиотечных комплексов) невозможно на IBM PC;

создание вычислительных сетей также предполагает использование высокопроизводительных машин.

Все это свидетельствует о том, что существует настоятельная потребность в приобретении навыков работы на больших ЭВМ.

Долгое время работа на ЕС ЭВМ в операционной системе ОС ЕС была малопривлекательной из-за сложности команд ОС ЕС, недостаточной сервисной поддержки пользователя. Однако с использованием операционной системы CBM ЕС (системы виртуальных машин) дело улучшилось коренным образом. Изменилось положение пользователя. CBM ЕС обеспечивает коллективный доступ к ЭВМ через дисплейные комплексы и предоставляет удобные и эффективные средства диалога.

В среде CBM ЕС общение с машиной осуществляется посредством команд, информация о выполнении которых сразу выдается на экран дисплея.

Исторически сложилось так, что развитие CBM предшествовало развитию операционной системы MS DOS для IBM PC и многое в MS DOS было заимствовано из CBM, а затем получило свое дальнейшее развитие.

Для грамотной работы в среде CBM ЕС необходима определенная подготовка. Здесь вступают в противоречие две тенденции: первая - стремление снизить требования к уровню подготовки пользователя; вторая - расширить функциональные возможности операционной системы. Эти проблемы решаются в CBM ЕС по двум направлениям:

1) применение простейших средств, рассчитанных на неподготовленного пользователя (например, возможность получить решение с помощью одной команды или даже только указав имя программы);

2) применение специальных средств и инструментальных средств, облегчающих доступ к сложным функциям системы (например, язык командных процедур, позволяющего автоматизировать сценарий диалога).

Одним из важнейших компонентов СВМ ЕС является подсистема диалоговой обработки (ПДО), позволяющая разрабатывать, отлаживать и эксплуатировать программы. Программирование в ПДО включает не только составление и отладку программ, написанных на традиционных языках программирования, таких как ПЛ/1, Фортран, Ассемблер, Паскаль, но и автоматизацию диалога с помощью языка командных процедур REXX.

Основными задачами данного курса является изучение основ функционирования и средств обеспечения пользователя в системе виртуальных машин ЕС, приобретение практических навыков программирования на языке REXX, имеющем широкое применение для автоматизации диалоговых процедур, создания систем управления базами данных, разработки макрокоманд редактирования.

Лабораторные работы выполняются в дисплейном классе ЭВМ ЕС-1066.

Настоящее учебное пособие предназначено в первую очередь для студентов специальности «Программное обеспечение вычислительной техники и автоматизированных систем», а также для студентов других специальностей и пользователей вычислительной техники, изучающих программирование в СВМ ЕС. С целью лучшего усвоения учебного материала рекомендуется отвечать на контрольные вопросы, помещенные в конце каждого раздела и решить типовые задачи, приведенные в разделе 8.

1. ОСНОВНЫЕ ПРИНЦИПЫ ПОСТРОЕНИЯ СВМ ЕС

1.1. Конфигурация виртуальной машины

Система виртуальных машин обеспечивает такой принцип параллельной работы пользователей на одной реальной ЭВМ, при котором каждому из них выделяется персональная виртуальная машина (ВМ).

ВИРТУАЛЬНАЯ МАШИНА является функциональным эквивалентом реальной ЭВМ, моделируемым с помощью реальных технических средств и управляющих программ СВМ ЕС.

КОНФИГУРАЦИЯ ВМ включает следующие виртуальные элементы:

центральный процессор (ЦП);

пульт управления (ПУ);

основную память (ОП);

каналы ввода-вывода (КВВ);

периферийные устройства: устройство ввода-вывода перфокарт, канал, АЦПУ, НМЛ, НМД.

Все перечисленные виртуальные устройства позволяют создать у пользователя иллюзию того, что он располагает персональной ЭВМ.

ЛОГИЧЕСКАЯ СХЕМА СВМ изображена на рис.1.

Все средства СВМ разделены на реальные и виртуальные.

Реальная ЭВМ управляется операционной системой СВМ ЕС, а ВМ — собственными операционными системами, причем при одновременной работе нескольких ВМ в каждую из них можно загрузить свою операционную систему.

1.2. Методы моделирования виртуальных устройств

Виртуальные устройства моделируются программно-аппаратными средствами реальной ЭВМ. При этом используются различные методы моделирования.

Виртуальный центральный процессор (ЦП) моделируется методом квантования времени реального процессора.

Для каждого виртуального ЦП создается специальный управляющий блок, в котором формируются следующие виртуальные элементы:

регистр PSW (слова состояния программы);

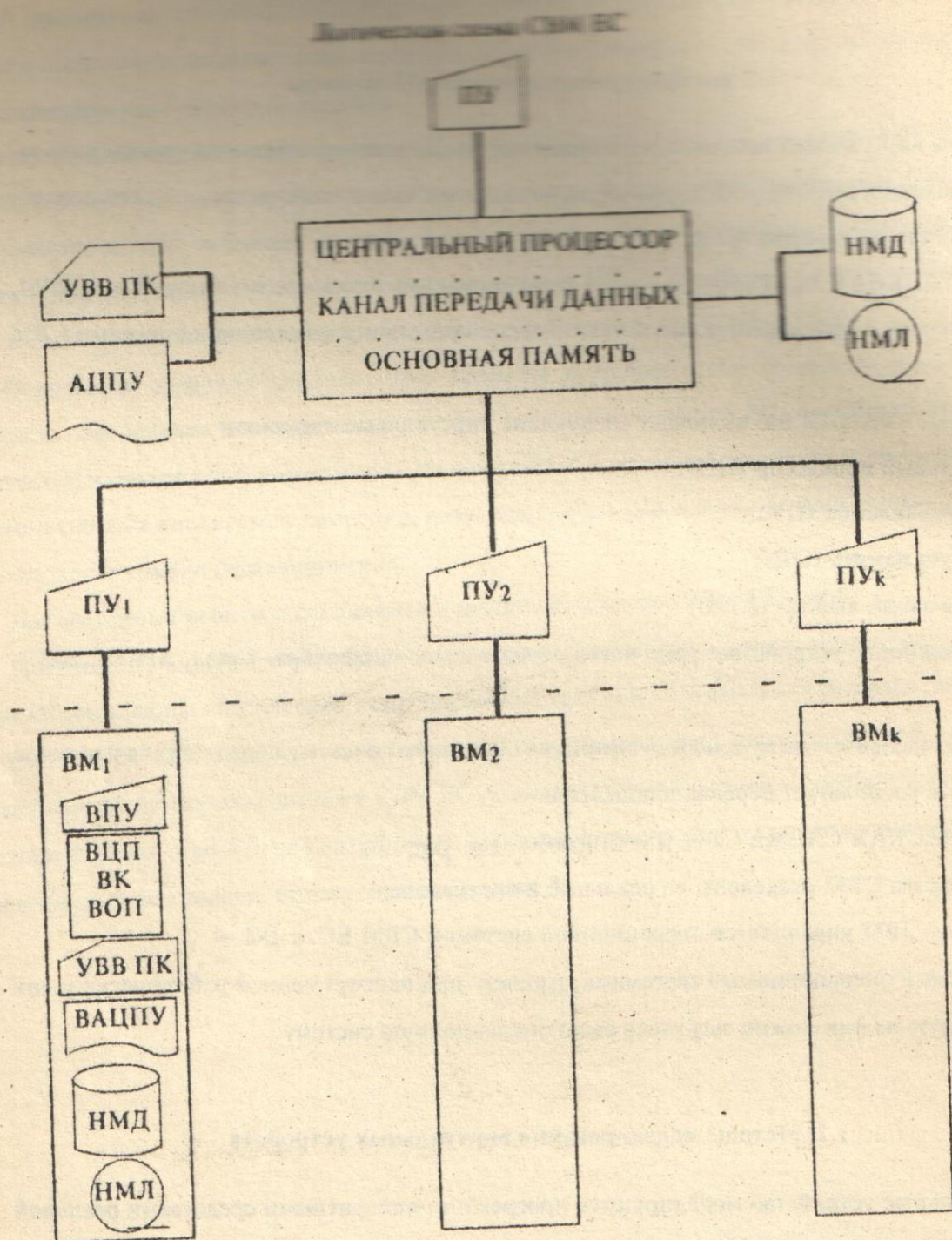


Рис. 1

регистры с плавающей запятой;

регистры управления;

постоянно-распределенные ячейки памяти.

Эти элементы обеспечивают независимость виртуальных ЦП, хотя все они моделируются на одном реальном процессоре.

Как только виртуальный процессор получает управление реальным, все перечисленные виртуальные элементы из управляющего блока загружаются в соответствующие устройства реального процессора, настраивая его на работу с данной ВМ.

Виртуальные ЦП получают управление реальным процессором на определенные отрезки времени, называемые квантами.

Кванты времени зависят от частоты поступления сигналов прерывания с пульта управления. Тем ВМ, от которых чаще поступают прерывания (т.е. пользователь активнее) реальный процессор выделяется чаще, но величина кванта меньше, чем для ВМ с малой активностью пользователя.

Виртуальная память выполняет функцию основной памяти ВМ. Виртуальная память состоит из виртуальных страниц размером 4К, которые объединяются в сегменты объемом 64К.

Моделирование виртуальной основной памяти производится как ее отображение на реальную основную и внешнюю память. При этом виртуальные страницы, используемые задачей в некотором интервале времени, размещаются в реальной основной памяти на выделенных им физических рамках памяти (блоках размером 4К), остальные виртуальные страницы - в специальной внешней памяти на томах прямого доступа.

Виртуальная система ввода-вывода включает виртуальные каналы и периферийные устройства, которые моделируются на базе реальных устройств или чисто аппаратным способом. При этом используют различные методы моделирования.

Для моделирования ВНМЛ используется метод выделения, т.е. отображение виртуального устройства на реальное. Этот метод прост, но требует большого числа реальных устройств.

Для моделирования ВНМД применяют метод разделения. Этот метод позволяет использовать одно реальное устройство одновременно несколькими виртуальными. Моделирование ВНМД основано на выделении мини-дисков (МД), т.е. целого числа цилиндров, пронумерованных подряд и имеющих собственную организацию (рис.2).

Моделирование виртуальных каналов ввода-вывода производится методом накопления, т.е. каждому виртуальному устройству выделяется файл накопления, размещенный на устройстве прямого доступа. Информация, предназначенная для ввода-вывода, поступает в файл накопления, затем этот файл можно передать на любое реальное устройство.

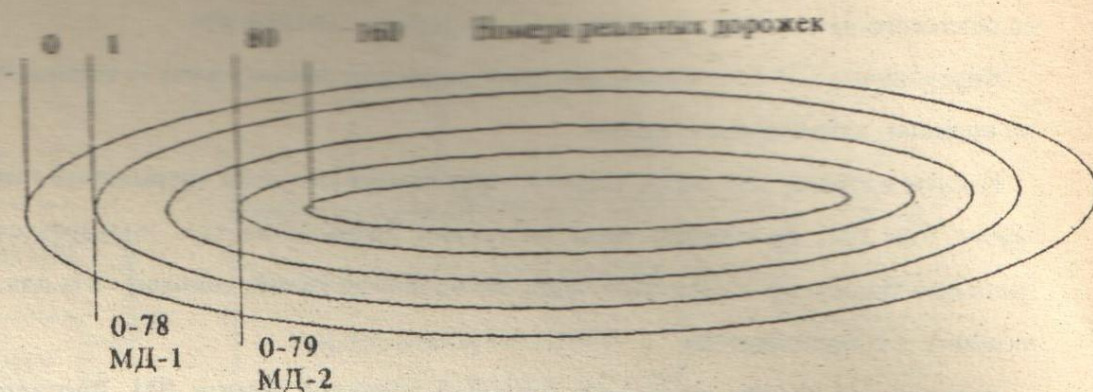


Рис. 2

Виртуальный пульт управления служит для управления работой ВМ и связан с операционной системой. Он моделируется методом выделения. Его физическим аналогом может быть дисплей ЕС-7927 или пультовая пишущая машинка ЕС-7070.

1.3. Состав СВМ ЕС

В состав СВМ ЕС входят следующие основные компоненты:

- монитор виртуальных машин /МВМ/;
- подсистема диалоговой обработки;
- подсистема сетевой передачи файлов;
- подсистема логической коммутации абонентских пунктов;
- подсистема анализа дампов;
- подсистема контроля технических средств;
- средства генерации и обслуживания.

МОНИТОР ВИРТУАЛЬНЫХ МАШИН является управляющей программой СВМ, которая распределяет ресурсы реальной ЭВМ и управляет их работой. МВМ создает ВМ и обеспечивает их параллельную работу.

Программы МВМ выполняются на реальной ЭВМ в состоянии «супервизор», что позволяет реализовать любые привилегированные функции. Остальные программные средства, в том числе и операционные системы ВМ, выполняются на реальной машине в состоянии «задача». Это гарантирует защиту ВС от вмешательства пользователей.

ПОДСИСТЕМА ДИАЛоговой ОБРАБОТКИ (ПДО) является операционной системой для ВМ. Она ориентирована на одного пользователя и обеспечивает развитые диалоговые средства взаимодействия с ВМ. Пользователь обслуживается только в однопрограммном режиме, т.е. не обеспечивается мультипрограммирование. Этим ограничением достигается высокая эффективность /производительность/ системы.

ПДО поддерживается монитором. Поэтому ПДО не может работать в качестве самостоятельной системы на реальной машине. Зависимость ПДО от МВМ заключается не только в том, что ПДО загружается в ВМ, порожденную монитором, но и в том, что ряд функций (например, управление вводом-выводом, обработка сбоев технических средств и т.д.) выполняется монитором.

ПОДСИСТЕМА СЕТЕВОЙ ПЕРЕДАЧИ ФАЙЛОВ (ПСП) - это специализированная операционная система ВМ. Она предназначена для передачи данных между ЭВМ, связанными в сеть с помощью каналов передачи, адаптера канал-канал и общего поля памяти на запоминающих устройствах прямого доступа. В сеть, поддерживаемую ПСП, кроме ЭВМ, могут входить абонентские пункты ЕС-8504 и печатающие устройства дисплейных комплексов.

Работой элементов сети должны управлять собственные операционные системы. Например:

СВМ с подсистемой передачи файлов (ПСП);

ОС ЕС с подсистемой ввода заданий КРОС;

другие операционные системы;

ЭВМ, входящие в сеть, называются узлами.

В сети, поддерживаемой ПСП, могут передаваться файлы накопления, сообщения пользователей ВМ и команды оператора ПСП.

Файл, посланный в некоторый узел для вывода на реальное печатающее или перфокарточное устройство, помещается виртуальной машиной ПСП данного узла на соответствующее виртуальное устройство. О приеме файла извещается пользователь этого узла. Пользователь ВМ может просмотреть полученный файл на устройстве отображения, сохранить его во внешней памяти и передать в любой узел ПСП сообщение и запросить информацию о состоянии любого другого узла сети или выполнить другие действия.

Возможность обмена файлами и сообщениями представляется также пользователю ВМ, работающему в ПДО, где для этого есть специальные средства взаимодействия с ПСП.

ПОДСИСТЕМА ЛОГИЧЕСКОЙ КОММУТАЦИИ (ПЛК) абонентских пунктов также используется в сетях ЭВМ, но обеспечивает иные, чем ПСП, возможности сетевого взаимодействия.

ПЛК работает на ВМ под управлением ПДО. ПЛК предназначена для динамической коммутации устройств отображения ВМ (дисплеев) между различными узлами сети.

При такой коммутации устройство отображения, физически подключенное к одной ЭВМ, входящей в состав сети, логически может быть подключено к любой другой ЭВМ этой сети.

В сеть ПЛК может входить произвольное число ЭВМ, связанных между собой каналами передачи данных или адаптерами канал-канал.

Пользователь узла ПЛК может получить доступ к любому узлу сети, если к нему есть путь от данного узла.

После подготовки ПЛК к работе необходимо специальной командой, вводимой с пульта ПДО; установить с ней связь, то есть пользователь становится абонентом ПЛК и может через свой дисплей связаться с любым узлом сети и выполнить следующие действия:

- 1) сделать свой дисплей пультом управления той ВМ, которая описана в данном узле;
- 2) присоединить свой дисплей к ВМ, на которой работает система с коллективным доступом;
- 3) сделать свой дисплей доступным для реальной ЭВМ с операционной системой, отличной от СВМ. При таком присоединении данный дисплей будет устройством отображения для другой ЭВМ.

ПОДСИСТЕМА АНАЛИЗА ДАМПОВ (ПАД). Средство системного программиста, служащее для учета и регистрации аварийных ситуаций из-за сбоев технических и программных средств и диалогового анализа дампов реальной основной памяти.

ПОДСИСТЕМА ДИСТАНЦИОННОЙ ПЕРЕДАЧИ ФАЙЛОВ (ПДП) является прототипом ПСП, сохраненным от ранних версий СВМ и выполняет подмножество функций ПСП.

ПОДСИСТЕМА КОНТРОЛЯ ТЕХНИЧЕСКИХ СРЕДСТВ (ПКТ). Используется дежурными электронщиками и служит для предупреждения отказов аппаратуры и уменьшения их влияния на эффективность вычислительной системы. Она дает возможность оперативно контролировать все устройства реальной ЭВМ путем динамического анализа параметров состояния и вывода предупреждающих сообщений о возможном отказе.

СРЕДСТВА ГЕНЕРАЦИИ И ОБСЛУЖИВАНИЯ. Предназначены для генерации на ВЦ конкретных вариантов СВМ ЕС, соответствующих конфигурации технических средств и условиям эксплуатации.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ К РАЗДЕЛУ I

1. Что включает в себя понятие «виртуальная машина»?
2. Какова конфигурация ВМ?
3. Охарактеризуйте основные методы моделирования виртуальных устройств.
4. Назовите состав СВМ ЕС.
5. Изобразите логическую схему СВМ.
6. Каковы основные принципы построения СВМ ЕС?
7. Каково назначение подсистемы сетевой передачи файлов?
8. Какие подсистемы СВМ ЕС обеспечивают сетевое взаимодействие?

2. ОБЩИЕ СВЕДЕНИЯ О МОНИТОРЕ ВИРТУАЛЬНЫХ МАШИН

2.1. Назначение и основные функции монитора виртуальных машин

Основное назначение МВМ - обеспечить одновременное функционирование множества ВМ на одной реальной ЭВМ.

Так как ВМ - это совокупность виртуальных устройств, управляемых своей собственной операционной системой, а реальные технические средства ЭВМ находятся в ведении МВМ, то монитор должен:

- 1) следить за состоянием всех ВМ;
- 2) обеспечивать физическую поддержку виртуальных устройств реальными;
- 3) моделировать функции виртуальных устройств.

Для этого МВМ должен выполнять следующие основные функции:

учет всех зарегистрированных ВМ;
учет, распределение и координацию работы ресурсов реальной ЭВМ;
обработку всех прерываний (как от ВМ, так и от собственных программ);
организацию виртуальной памяти (как для ВМ, так и для своих нужд);
управление страничным обменом;
управление операциями ввода-вывода на виртуальных и физических устройствах;

организацию службы времени;

контроль доступа к ресурсам вычислительной системы.

Большинство перечисленных функций являются привилегированными, поэтому программы MBM (и только они) выполняются на центральном процессоре в режиме «супервизор».

Учет реальных устройств ЭВМ выполняется с помощью специальных управляющих блоков, содержащих необходимую информацию об устройствах.

Для учета VM создается специальный файл, называемый оглавлением CBM EC.

Оглавление формируется на специальном томе монитора с помощью команды DIRECT или системно независимой программы DMKDIR.

Для каждой VM в оглавлении создается отдельный элемент с именем VMBLOCK, который содержит следующую информацию:

идентификатор VM;

пароль;

учетный номер;

стандартный и максимальный размер виртуальной памяти, выделяемой VM;

классы допустимых команд MBM;

приоритет VM;

режимы работы VM по умолчанию;

адреса и типы виртуальных устройств, включенных в конфигурацию VM;

ссылки на другие блоки, связанные с данной VM.

Управление VM выполняется с помощью прерываний. Прерывания могут поступать при выполнении как программ VM, так и программ MBM. Их различают по состоянию центрального процессора.

Если в момент прерывания процессор был в состоянии «супервизор», то прервана программа MBM, а если в состоянии «задача» - прервана программа VM.

При выполнении программ VM разрешаются все прерывания.

При выполнении программ MBM запрещаются внешние прерывания и прерывания по вводу-выводу.

Если управление центральным процессором передается некоторой VM, монитор настраивает PSW и регистры в соответствии со значением виртуальных PSW и регистров данной VM. Кроме того, в PSW устанавливается состояние «задача» (разряд 14) и режим EC-режим расширенного управления (12-й разряд PSW) и разрешаются все прерывания.

После получения реального центрального процессора ВМ управляет им с помощью своей операционной системы до истечения выделенного кванта времени или до наступления одного из следующих событий:

- выполнения привилегированной команды;
- обращения к супервизору;
- окончания операции ввода-вывода;
- обращения к виртуальной странице, отсутствующей в реальной памяти.

Во всех этих случаях возникает прерывание и управление переходит к монитору. МВМ запоминает состояние ЦП в управляющем блоке ВМ для того чтобы обеспечить продолжение работы ВМ при получении следующего кванта времени.

Запомнив состояние прерванной ВМ и выполнив функцию, запрошенную прерыванием, монитор снова выбирает машину, которой предоставляется ЦП на очередной квант времени.

Одной из основных функций МВМ является также защита ВМ и мини-дисков от несанкционированного доступа с помощью паролей.

Пароль защиты ВМ действует при обработке команды LOGON, с которой начинается сеанс работы с машиной.

Пароль защиты мини-диска при выполнении команды LINK (доступ к мини-диску).

2.2. Принципы управления виртуальными машинами

На ресурсы реальной ЭВМ претендуют ВМ и программы монитора.

Чтобы обеспечить эффективность управления устанавливается приоритет программ монитора над программами ВМ. Среди программ МВМ наибольшее предпочтение отдается программам управления вводом-выводом.

Все эти объекты управления представлены в СВМ ЕС управляющими блоками.

Таким образом, все претенденты на ЦП реальной ЭВМ выстраиваются в три очереди:

- 1) очередь блоков IOBLOK - очередь программ управления вводом-выводом;
- 2) очередь блоков CPXVBLOK - очередь других программ МВМ;
- 3) очередь блоков VMBLOK - очередь ВМ.

Между объектами этих очередей действует абсолютная приоритетность с прерыванием, то есть объект из очереди с меньшим приоритетом (большим номером) не обслуживается, пока не будут исчерпаны очереди объектов с большим приоритетом. При поступлении объектов в очередь с большим приоритетом обслуживание объекта с меньшим приорите-

том прерывается. Все программы монитора обслуживаются ЦП до своего завершения. Виртуальным машинам реальный ЦП выделяется квантами. При выполнении программ монитора запрещаются внешние прерывания и прерывания ввода-вывода. При работе ВМ разрешены все прерывания.

Управление виртуальными машинами производится на основе предварительного планирования. При планировании определяются последовательность и длительность использования ВМ реального ЦП.

Последовательность получения ЦП определяется положением ВМ в очереди. Упорядочение ВМ в очереди производится по граничному приоритету (ГП), который вычисляется как некоторая функция от времени постановки ВМ в очередь (ТО) и от ее диспетчерского приоритета (ДП):

$$\text{ГП} = f(\text{ТО}, \text{ДП}). \quad (1)$$

В простейшем случае граничный приоритет можно вычислить по формуле

$$\text{ГП} = \text{ТО} + \text{ДП}, \quad (2)$$

где ТО - текущее время при постановке в очередь.

При этом граничный приоритет позволяет разместить ВМ на временной оси в том порядке, в котором они должны получать реальный ЦП (рис. 3).

Упорядочение очереди ВМ

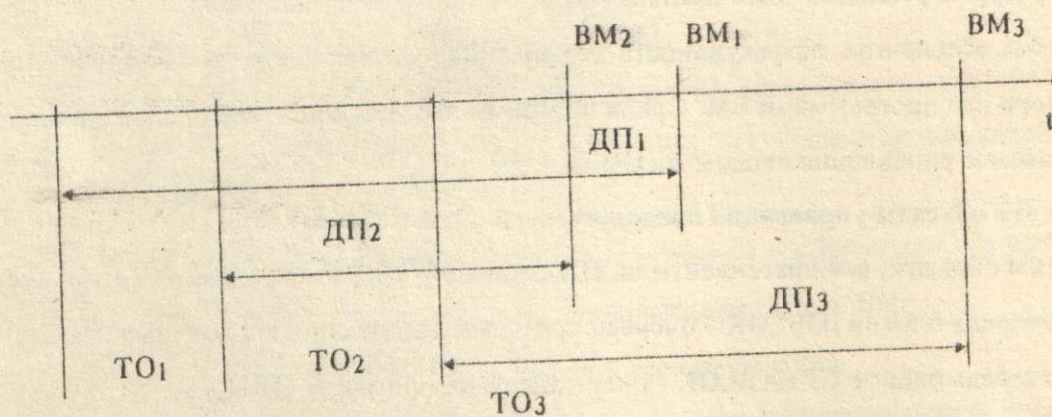


Рис.3

Длительность использования ЦП отдельной ВМ определяется величиной кванта и частотой его получения, которая, в свою очередь, зависит от количества ВМ, претендующих на процессор.

Функции разделения времени реального ЦП виртуальными машинами реализуются программами диспетчера и планировщика МВМ.

Программа диспетчер из трех приоритетных очередей выбирает претендента на обслуживание ЦП.

Программа планировщик управляет очередями ВМ по следующим правилам:

- 1) соблюдение нормативов потерь времени на страничный обмен (при слишком большом числе ВМ растут потери времени на организацию страничного обмена);
- 2) равноправное использование реального ЦП всеми ВМ за счет выравнивания суммарной продолжительности использования ЦП на большом отрезке времени;
- 3) обеспечение гарантированного времени ответа при заданной загрузке системы.

Для соблюдения этих правил применяется принцип обратной связи. Суть его состоит в том, что планировщик через каждые $2048 \Delta t$ (Δt величина кванта) подсчитывает показатели текущего состояния системы и вносит поправки на параметры управления следующим образом:

вычисляет затраты времени реального ЦП на страничный обмен в процентах от общего времени его работы и сравнивает их с нормативными. Если фактические затраты превышают нормативы, то уменьшается количество ВМ в обслуживании;

на отрезке времени, равном $16384 \Delta t$, подсчитывает среднее время использования реального ЦП одной ВМ;

определяет фактическое время использования ЦП каждой ВМ и сравнивает его со средним временем. Если фактическое время ВМ превышает среднее, то такую ВМ «наказывают», уменьшая ее диспетчерский приоритет;

контролирует время пребывания ВМ в очередях и учитывает это время при пересчете ее диспетчерского приоритета. Кроме того, этот показатель регулируется назначением кванта и интервала обслуживания (суммарного времени использования ЦП; после исчерпания своего интервала ВМ уступает свое место в очереди другим машинам) в зависимости от активности пользователя ВМ.

Таким образом, основная задача планировщика - определить место ВМ в очереди на обслуживание на основе регулярно пересчитываемого граничного приоритета, установить величину интервала обслуживания ВМ и задать значение кванта времени.

2.3. Состояния виртуальных машин

Состояния виртуальных машин учитываются при определении диспетчерского приоритета.

Состояние виртуальной машины после начала сеанса определяется поведением пользователей, ходом вычислительного процесса, состоянием системы СВМ ЕС.

Планировщик и диспетчер МВМ различают следующие состояния виртуальной машины (рис. 4):



Рис. 4

Вычислительная машина становится непланируемой, если:
ожидает завершения ввода-вывода с пульта управления;
выполняет команду МВМ;
ожидает внешнего прерывания.

Во всех остальных случаях виртуальная машина является планируемой.

Планируемая ВМ может быть активной или неактивной.

Активной является ВМ, включенная в очередь на обслуживание RUNLIST (3-я очередь).

Число активных ВМ ограничено возможностями обслуживания их запросов в динамической страничной памяти.

Неактивными становятся ВМ, не попавшие в очередь на обслуживание из-за ее ограниченности. Они являются кандидатами на обслуживание.

Активные ВМ могут быть диспетчируемыми и недиспетчируемыми.

Диспетчируемой ВМ является в том случае, если нет никаких препятствий для использования реального центрального процессора.

Недиспетчируемой ВМ становится, если она ожидает завершения одного из следующих событий:

моделирования в ее интересах ресурсов реальной ЭВМ (привилегированная команда, основная память и т.д.);

операций ввода-вывода, не связанных с пультом управления.

В зависимости от поведения пользователя ВМ может быть диалоговой, недиалоговой или пакетной.

Диалоговой ВМ становится после прерывания по вводу-выводу с ПУ (от пользователя).

ВМ считается недиалоговой, если в течение заданного отрезка времени не поступали сигналы прерывания по вводу-выводу с ПУ.

Понятие пакетной ВМ рассмотрим позднее.

Все планируемые ВМ (вернее, их управляющие блоки VMBLOCK) находятся в очередях планировщика:

активные ВМ - в очереди обслуживания;

неактивные - в очереди претендентов на обслуживание.

Все активные ВМ находятся в одной очереди, но приоритеты будут разные в зависимости от того, является ли ВМ диалоговой, недиалоговой или пакетной.

Для ВМ-претендентов на обслуживание есть две очереди:

для неактивных диалоговых (1-я);

для неактивных недиалоговых (2-я).

Причем 1-я из этих очередей имеет предпочтение на обслуживание.

Машины из 2-й очереди выбираются только, если 1-я очередь пуста.

Виртуальная машина, 6 раз подряд включенная в очередь RUNLIST из 2-й очереди, минуя 1-ю, получает характеристику чисто пакетной машины.

Диспетчерский приоритет ВМ периодически изменяется в зависимости от ее функционирования. Планировщик пересчитывает приоритеты ВМ при каждом удалении из очереди RUNLIST, а для диалоговых - ВМ и при постановке их в очередь неактивных диалоговых. При нахождении ВМ в неактивных очередях (1-й и 2-й) их приоритеты не меняются.

Для каждой из этих машин (с учетом ее состояния) существует своя формула для определения диспетчерского приоритета [3, с. 44-46].

2.4. Разделение времени центрального процессора между виртуальными машинами

После пуска СВМ ВМ регистрируются и поступают в очередь RUNLIST, а после ее наполнения - в очередь неактивных недиалоговых.

Так как характеристики ВМ, необходимые для планирования, собираются планировщиком в процессе ее работы, то сразу после регистрации они отсутствуют. Поэтому в качестве граничного приоритета используется время регистрации ВМ.

При постановке в очередь RUNLIST для каждой ВМ определяется величина кванта Δt и интервал обслуживания ΔT (суммарное время использования ЦП).

Значение кванта вычисляется и устанавливается специальными командами MBM. Начальное значение кванта зависит только от заданного базового значения и быстродействия ЭВМ.

$$\Delta t = \Delta t_0 \frac{R_1}{R_2}, \quad (3)$$

где $\Delta t_0 = 50$ мс - базовое значение кванта для ЭВМ ЕС-1035;

R_1 - быстродействие ЕС-1035;

R_2 - быстродействие ЕС-1066.

Интервал обслуживания зависит от величины кванта Δt и активности пользователя:

$$\Delta T = \begin{cases} 8\Delta t - \text{для диалоговой ВМ,} \\ 64\Delta t - \text{для недиалоговой ВМ,} \\ 512\Delta t - \text{для пакетной.} \end{cases} \quad (4)$$

Разделение реального времени ЦП между ВМ выполняет диспетчер MBM. Он выбирает 1-ю ВМ из очереди RUNLIST и запускает ее на ЦП. При первом обслуживании ВМ выделяется квант Δt . При повторном обслуживании величина кванта определяется по правилу:

$$\Delta t_n = \begin{cases} \Delta t, & \text{если на предыдущем кванте было прерывание по вводу-выводу с ПУ} \\ 4\Delta t - & \text{в противном случае} \end{cases}$$

При таком разделении времени ЦП и правилах формирования приоритетов диалоговые машины получают кванты меньшей величины, но чаще (этим достигается сокращение времени ответа).

Использование очередного кванта ВМ может завершиться одним из событий:

1) квант израсходован полностью;

В) поступило прерывание по вводу-выводу с ПУ или введена команда MBM;

З) прерывание по одной из следующих причин: задание операции ввода-вывода, не связанное с ПУ; отсутствие виртуальной страницы в реальной основной памяти.

В 1 -м случае (квант полностью израсходован) пересчитывается величина интервала обслуживания

$$\Delta T = \Delta T - \Delta t \quad \text{и проверяется условие } \Delta T > 0. \quad (5)$$

Если интервал обслуживания не исчерпан, то пересчитывается граничный приоритет

$$ГП = ГП_{\text{старый}} + \frac{ГП_{\text{старый}} - ГП_{\text{текущий}}}{4}. \quad (6)$$

Машина возвращается в очередь RUNLIST с новым граничным приоритетом.

Если ВМ исчерпала свой интервал обслуживания, то она переводится в неактивное состояние. Ей присписывается состояние недиалоговой и вычисляется новое значение диспетчерского и граничного приоритетов. Затем ВМ устанавливается в очередь неактивных недиалоговых машин с новым значением граничного приоритета.

Во 2 -м случае (сигнал прерывания с ПУ) ВМ становится непланируемой, исключается из очереди RUNLIST и остается вне списков планировщика.

В 3 -м случае (прерывания, не связанные с ПУ) в виртуальном PSW машины устанавливается признак «ожидание», она переводится в состояние недиспетчируемой и остается в очереди RUNLIST.

При обслуживании непланируемая ВМ ожидает завершения ввода-вывода с ПУ, а по окончании этой операции переводится в состояние диалоговой неактивной. Для нее вычисляется новое значение граничного приоритета и она устанавливается в очередь неактивных диалоговых ВМ.

Недиспетчируемая ВМ при изменении в виртуальном PSW состояния «ожидание» на состояние «счет» переводится в ранг недиспетчируемой недиалоговой, для нее пересчитывается приоритет и она устанавливается в соответствующую очередь.

Таким образом, ВМ, получившая реальный ЦП, может изменить свое состояние в соответствии со схемой движения ВМ по очередям (рис.5).

Состояние ВМ изменяется следующим образом:

- используя временной квант, ВМ может либо вернуться в очередь RUNLIST с новым приоритетом, если не исчерпан интервал обслуживания, либо попасть в очередь неактивных недиалоговых, если ΔT исчерпан;

Схема движения ВМ по очередям

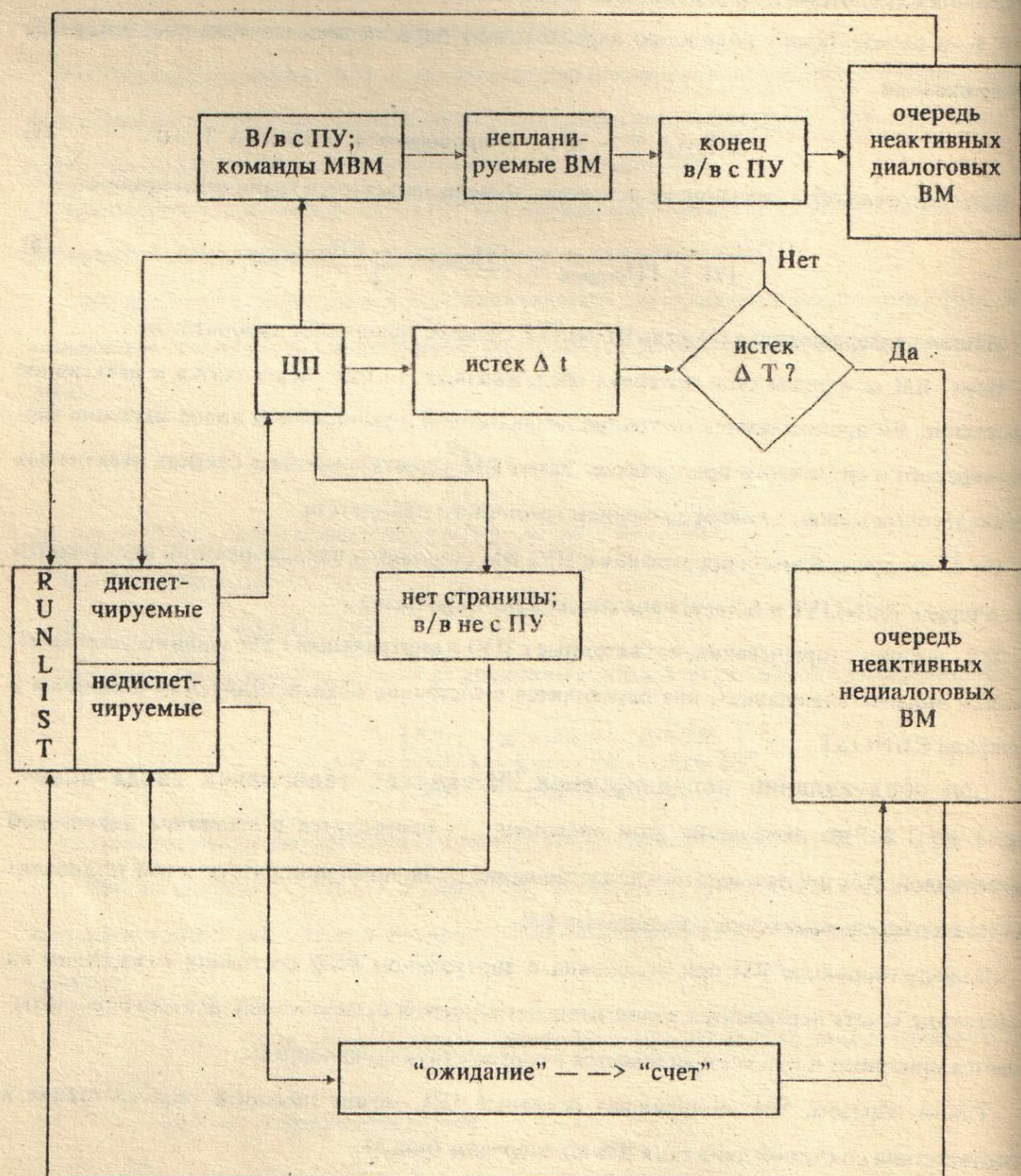


Рис. 5

ВМ может быть прервана с ПУ и стать непланируемой с последующей перспективой перейти в состояние диалоговой и попасть в очередь неактивных диалоговых ВМ;

ВМ может перейти в состояние ожидания событий, не связанных с ПУ, стать недиспетчируемой, но сохраниться в очереди RUNLIST с перспективой попасть в очередь неактивных недиалоговых ВМ.

После каждого удаления ВМ из очереди RUNLIST планировщик делает попытку пополнения этой очереди неактивными машинами. При этом сначала выбираются ВМ из очереди неактивных диалоговых, а когда она исчерпана, то из очереди неактивных недиалоговых.

Целесообразность пополнения очереди RUNLIST новыми ВМ оценивается с учетом объема реальной основной памяти, доступной для страничного обмена, и запросов активных ВМ на эту память.

Для оценки потребности ВМ в реальной основной памяти вводится специальная характеристика - рабочий набор, который равен количеству страничных рамок, запрошенных ВМ за весь интервал обслуживания. Рабочий набор определяется планировщиком автоматически по предыдущему интервалу.

Целесообразность пополнения очереди RUNLIST определяется условием

$$\sum S_i \leq N_p - C_r, \quad (7)$$

где S_i - рабочий набор i -й неактивной ВМ;

N_p - количество рамок основной памяти, доступных для страничного обмена;

C_r - суммарный рабочий набор ВМ, уже находящихся в очереди RUNLIST.

Условие (7) определяет только необходимость пополнения очереди RUNLIST. Но количество добавляемых машин зависит от другого условия. Для сокращения частоты переходов ВМ из неактивных очередей в активную планировщик продолжает пополнение очереди RUNLIST с учетом 25 %-ной перегрузки динамической страничной области.

Поэтому пополнение очереди продолжается до нарушения условия

$$\sum S_i \leq N_p - 0.75 C_r. \quad (8)$$

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ К РАЗДЕЛУ 2

1. В чем состоит основное назначение МВМ?
2. Перечислите основные функции МВМ.
3. Какую информацию содержит оглавление СВМ ЕС?
4. Охарактеризуйте основные принципы управления ВМ.
5. Каковы основные состояния ВМ?

Порядковый номер строки процедуры, при обработке которой возникла исключительная ситуация, определенная оператором SIGNAL ON, присваивается специальной переменной SIGL и может быть использован программистом для локализации ошибки.

Необходимо помнить, что оператор SIGNAL ON производит "перехват" только 1-й ошибки с указанным ключевым словом. Дальнейший "перехват" ошибок, аналогичных той, что уже была "перехвачена", нужно делать с помощью нового оператора SIGNAL ON с тем же ключевым словом.

Пример обработки исключительной ситуации ERROR.

```
SIGNAL ON ERROR
COPY . . .
```

```
RENAME . . .
```

```
EXIT /* КОНЕЦ ОСНОВНОЙ ПРОГРАММЫ */
/* ОБРАБОТКА ОШИБОК, СВЯЗАННЫХ С */
/* НЕНУЛЕВЫМ КОДОМ ВОЗВРАТА */
```

```
ERROR:
```

```
SAY " НЕПРЕДВИДЕННЫЙ КОД ВОЗВРАТА " RC,
    "!" ИЗ КОМАНДЫ:"
```

```
SAY " " SOURCELINE(SIGL)
```

```
SAY "В СТРОКЕ" SIGL "."
```

Оператор EXIT необходим для того, чтобы аварийный фрагмент не выполнялся при нормальном завершении программы.

Пример "выключения" проверки кода возврата для некоторой команды:

```
SIGNAL ON ERROR
COPY . . .
```

```
RENAME . . .
```

```
SIGNAL OFF ERROR
```

```
ERASE OLD LISTING A
```

```
SIGNAL ON ERROR
```

```
COPY . . .
```

В данном фрагменте есть 4 команды, которые могут дать ненулевой код возврата. Однако пользователь предполагает, что ненулевой код может дать команда ERASE, так как не знает есть ли этот файл на диске. Но поскольку данный файл все равно придется удалять, то для "выключения" оператора SIGNAL ON ERROR для команды ERASE используется оператор SIGNAL OFF ERROR. Затем обработка исключительной ситуации ERROR снова "включается" следующим оператором SIGNAL ON.

Пример обработки синтаксической ошибки:

```
SIGNAL ON SYNTAX
```

```
SYNTAX:
```

```
SAY "ОШИБКА" RC "В СТРОКЕ" SIGL
```

```
SAY SOURCELINE(SIGL) /* ТЕКСТ СТРОКИ */
```

```
SAY ERRORTXT(RC) /* ТЕКСТ СООБЩЕНИЯ ОБ ОШИБКЕ */
```

```
EXIT RC
```

Использование "перехвата" ошибки оператором SIGNAL ON NOVALUE может оказаться полезным для проверки правильности программ, в которых для повышения надежности все цепочки, не являющиеся числами, заключены в кавычки.

СПИСОК ЛИТЕРАТУРЫ

1. Программирование в системе виртуальных машин ЕС ЭВМ: Справ. изд. /Г.И.Коваль, Т.М.Коротун, Е.М.Лаврищева. - М.: Финансы и статистика, 1990. - 256 с.
2. Программирование в подсистеме диалоговой обработки СВМ ЕС /А.В.Алексеев, Д.Д.Горбатенко, А.В.Сержантов. - М.: Радио и связь, 1990. - 256 с.
3. Тимонин В.И. СВМ ЕС: Основы функционирования и средства обеспечения пользователя. - М.: МАИ, 1990. - 232 с.
4. ЕС ЭВМ. СВМ. Подсистема диалоговой обработки. Расширенный текстовый редактор. Описание команд. Е1.00005-039105.

ОБРАБОТКА ОШИБОК В REXX-ПРОЦЕДУРАХ

Во время выполнения программы на языке REXX возникают ошибки. При появлении ошибки в некотором предложении REXX - процедуры интерпретация прекращается и на экран выводится сообщение.

Однако ошибки могут быть не только синтаксическими. Ошибки возникают также из-за некорректных входных данных, нехватки памяти, ошибок ввода-вывода, из-за неправильного выполнения команд. Поэтому программист должен предусмотреть в программе обработку ошибок. Для этой цели в языке REXX используется оператор SIGNAL, причем есть два варианта конструкции SIGNAL:

- 1) SIGNAL
- 2) SIGNAL ON

Первая конструкция служит для безусловного перехода на метку в случае аварийного завершения программы, вторая конструкция — для обработки ("перехвата") ошибок, вызванных различными причинами.

Рассмотрим первую конструкцию.

Формат оператора:

SIGNAL МЕТКА

По этому оператору происходит безусловный переход к другой части программы, начинающейся с конструкции:

МЕТКА:

Необоснованное использование оператора SIGNAL подобно оператору GOTO (из других языков программирования) крайне вредно, так как если SIGNAL находится внутри конструкции SELECT или цикла DO, то при выходе из этих конструкций возвратиться обратно и продолжить работу уже невозможно. Поэтому для организации обычных переходов внутри программы лучше использовать операторы IF и SELECT. Оператор SIGNAL МЕТКА лучше использовать только для аварийного завершения программы. Оператор SIGNAL запоминает номер строки, в которой он расположен, в специальной переменной SIGL языка REXX.

Пример использования этой конструкции:

```
SIGNAL AVARIA
```

EXIT

```
/* КОНЕЦ ОБЫЧНОГО ЗАВЕРШЕНИЯ ПРОГРАММЫ */
/* АВАРИЙНОЕ ЗАВЕРШЕНИЕ ПРОГРАММЫ */
```

AVARIA:

```
SAY " АВАРИЙНОЕ ЗАВЕРШЕНИЕ В СТРОКЕ " SIGL,
```

```
"ПРОДОЛЖЕНИЕ НЕВОЗМОЖНО"
```

EXIT

Первый оператор EXIT служит для того чтобы предотвратить выполнение фрагмента начинающегося с метки

AVARIA:

при нормальном последовательном выполнении программы.

Рассмотрим конструкцию SIGNAL ON. Она используется для того чтобы указать интерпретатору на необходимость "перехватить" ошибки (исключительные ситуации) различного типа и переходить в случае их возникновения к соответствующим меткам.

Формат конструкции:

```
SIGNAL ON  ERROR
           OFF  HALT
           NOVALUE
           SYNTAX
```

Назначение ключевых слов:

- ON - "включить" перехват исключительных ситуаций, определенных одним из ключевых слов ERROR, HALT, NOVALUE, SYNTAX;
- OFF - "выключить" (отменить) перехват исключительных ситуаций с одним из ключевых слов ERROR, HALT, NOVALUE, SYNTAX;
- ERROR - определяет ситуацию, связанную с получением ненулевого кода возврата при выполнении команды ПДО, МВМ или редактора;
- HALT - исключительным событием является прекращение выполнения программы извне с помощью команды HI, прерывающей интерпретацию;
- NOVALUE - определяет в выполняемом выражении появление символического имени, не являющегося именем переменной;
- SYNTAX - исключительным событием является синтаксическая ошибка.

При возникновении исключительной ситуации, соответствующей одному из указанных в операторе SIGNAL ON ключевому слову ERROR, HALT, NOVALUE, SYNTAX, производится переход на первое встретившееся с начала процедуры предложение с меткой ERROR:, HALT:, NOVALUE:, SYNTAX: соответственно.

ОГЛАВЛЕНИЕ	
Предисловие	3
1. ОСНОВНЫЕ ПРИНЦИПЫ ПОСТРОЕНИЯ СВМ ЕС	5
1.1. Конфигурация виртуальной машины	5
1.2. Методы моделирования виртуальных устройств	8
1.3. Состав СВМ ЕС	11
Контрольные вопросы и задания	11
2. ОБЩИЕ СВЕДЕНИЯ О МОНИТОРЕ ВИРТУАЛЬНЫХ МАШИН	11
2.1. Назначение и основные функции монитора виртуальных машин	13
2.2. Принципы управления виртуальными машинами	16
2.3. Состояния виртуальных машин	18
2.4. Разделение времени центрального процессора между виртуальными машинами	21
Контрольные вопросы и задания	22
3. РАЗРАБОТКА ПРОГРАММ В ПОДСИСТЕМЕ ДИАЛоговой ОБРАБОТКИ	22
3.1. Общая характеристика подсистемы диалоговой обработки	26
3.2. Начало и завершение сеанса работы на ВМ	28
3.3. Создание и редактирование файлов	31
3.4. Трансляция, загрузка и выполнение программы	34
3.5. Использование универсальной процедуры RUN и командной REXX-процедуры	38
Контрольные вопросы и задания	38
4. ОСНОВНЫЕ КОНСТРУКЦИИ ЯЗЫКА ПРОЦЕДУР	38
4.1. Общие сведения и основные элементы	38

4.2. Типы предложений	46
4.3. Операторы ввода-вывода	46
4.4. Организация ветвления программы	48
4.5. Циклы	51
Контрольные вопросы и задания	56
5. ПОДПРОГРАММЫ И ФУНКЦИИ ЯЗЫКА REXX	57
5.1. Структура REXX-программы	57
5.2. Подпрограммы и функции пользователя	58
5.3. Сходство и различия подпрограмм и функций	67
5.4. Встроенные функции	68
Контрольные вопросы и задания	78
6. ИСПОЛЬЗОВАНИЕ ПРОГРАММНОГО СТЕКА И ОБРАБОТКА ФАЙЛОВ	78
6.1. Структура стека пульта и программного стека	78
6.2. Операторы REXX и команды ПДО для работы с программным стеком	80
6.3. Чтение файлов	83
6.4. Запись файлов	87
Контрольные вопросы и задания	89
7. ИСПОЛЬЗОВАНИЕ ЯЗЫКА REXX ДЛЯ РАЗРАБОТКИ МАКРОКОМАНД РЕДАКТИРОВАНИЯ	90
7.1. Правила написания макрокоманд	90
7.2. Обработка кодов возврата подкоманд редактора в макрокомандах	92
7.3. Подкоманды редактора, предназначенные для использования в макрокомандах	93

7.4. Примеры разработки макрокоманд	95
Контрольные вопросы и задания	99
8. ТИПОВЫЕ ЗАДАНИЯ ДЛЯ ЛАБОРАТОРНЫХ ЗАНЯТИЙ И САМОСТОЯТЕЛЬНОЙ РАБОТЫ	99
8.1. Общие замечания	99
8.2. Работа 1. Обработка массивов данных	100
8.3. Работа 2. Применение подпрограмм и функций	101
8.4. Работа 3. Обработка текстовых данных	102
8.5. Работа 4. Работа с файлами	103
8.6. Работа 5. Разработка REXX-процедур, содержащих команды ПДО	104
8.7. Работа 6. Разработка макрокоманд редактирования	106
Приложения	108
СПИСОК ЛИТЕРАТУРЫ	113

Навчальне видання

Бондаренко Михайл Федорович
Лісна Наталія Советівна, Щубін Ігор Юр'євич

Основи програмування мовою REXX
у підсистемі діалогової обробки
системи віртуальних машин ЕС ЕОМ

Навчальний посібник

Темплан І993, поз.440

Редактор А.П.Костіна
Коректор Н.Ф.Слоніна

Підп. до друку 20.09.93. . Формат 60×84^{1/16}. Папір
друк. № 3 . Друк офсетний. Ум. др. арк. 6,29. Ум. фарбо-відб. 6,85.
Облік.-вид. арк. 5,22. . Тираж 500
Зам. № 1533. Ціна 1 коп.

Інститут системних досліджень освіти України
252070, Київ, вул. Сагайдачного, 37

Фирма «ВИПОЛ»
252151, Київ, вул. Волинська, 60.