

## 4.9. ОСОБЕННОСТИ ОПЕРАЦИОННЫХ СИСТЕМ МНОГОМАШИННЫХ БОРТОВЫХ ЦВС

На рис. 4.26 представлена иерархическая структура операционной системы БЦВС, содержащей БЦВМ Ц101, Ц102, Ц200.

ОС БЦВМ Ц101, ОС Ц102 являются дальнейшим развитием - ОС БЦВМ Ц100 в части технологии их создания и отработки.

Операционная система Ц200 по своим функциям практически не отличается от ОС Ц101, Ц102, но в силу ограниченности ресурсов вычислителя Ц200 реализована в БЦВМ Ц102.

Передача управления той или другой операционной системе конкретной БЦВМ осуществляется программой более высокого уровня ОС БЦВС по установленному приоритету. Управление вычислительным процессом в Ц200 осуществляется по шине МПИ со стороны БЦВМ Ц102.

В состав операционной системы реального времени входит следующий базовый набор программ:

программа начального пуска, осуществляющая настройку операторов операционной системы под конкретную конфигурацию;

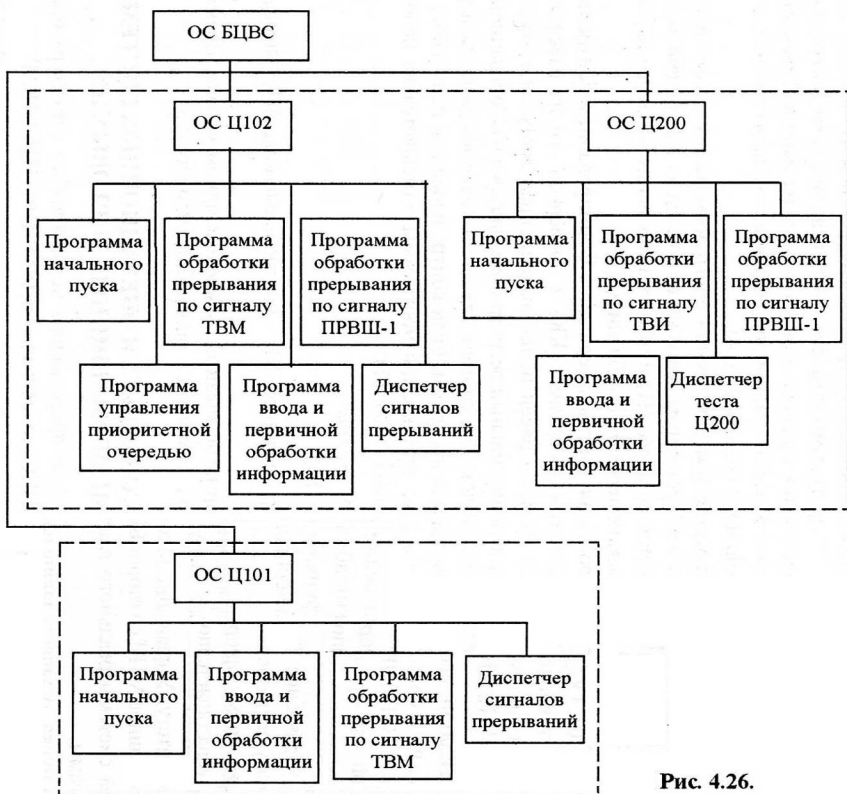


Рис. 4.26.

диспетчер обработки сигналов прерываний, осуществляющий управление приоритетами обработки сигналов прерываний, контроль за заполнением уровней слова состояния программ и буферов операционной системы, формированием заявок на обработку пакетов функциональных программ;

программы обработки прерываний по сигналам от таймеров (ТВМ – таймера временных меток и ТВИ – таймера временных интервалов);

программы обработки сигналов прерываний от внешних источников (ПРВШ1, ПРВШ2);

программа ввода и первичной обработки информации.

Результаты отработки ОС БЦВМ Ц100 в составе бортовых управляющих комплексов показали, что особая проблема заключается в поддержке устойчивого функционирования программного обеспечения в реальном времени. Например, блокировка прерываний на время принятия решений операционной системой может привести к увеличению времени реакции на события, появившиеся в этот интервал времени, а иногда, и к их потере, в результате чего может нарушиться временная диаграмма вычислительного процесса. Для решения этой проблемы были разработаны дополнительные функции управления системой прерывания, выполняющие запоминание состояния операционной системы и восстанавливающие её состояние.

Использование этих функций позволило уменьшить время блокировки сигналов прерывания до минимального значения, исключив тем самым потерю событий, появляющихся во время работы операционной системы.

На рис. 4.27 схематично изображено взаимодействие между функциями, реализованными операторами при исполнении ОС РВ.

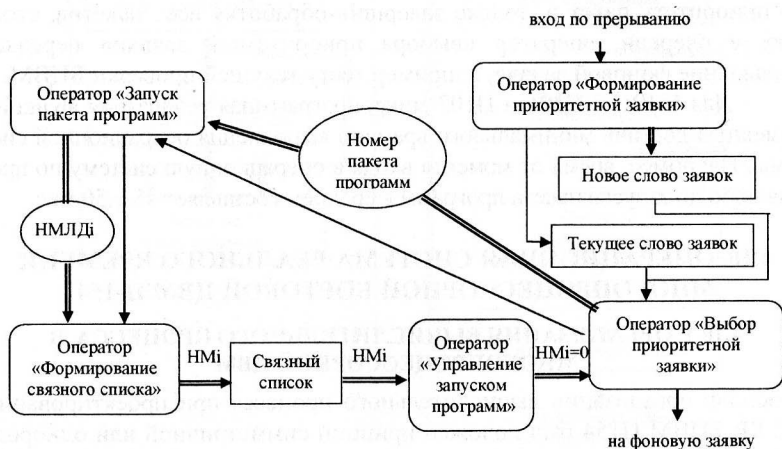


Рис. 4.27

На рисунке двойные стрелки означают взаимодействие по информации, одиночные стрелки - взаимодействие по управлению. В результате выполнения операторов, настроенных на конкретный вычислительный процесс, проводятся последовательно обработка заданных типов прерывания, формирование заявок на запуск пакетов программ, выбор приоритетной заявки с учетом текущего состояния процесса.

Оператор «выбор приоритетной заявки» анализирует приоритеты вновь пришедших заявок («новое слово заявок») и обрабатываемых в текущий момент времени («текущее слово заявок»). Если новая заявка более приоритетная, чем обрабатываемые в текущий момент времени, то оператором запускается пакет программ, соответствующий новой заявке, а текущая запоминается и будет обработана после завершения более приоритетной. В противном случае приоритетной является текущая заявка и она выбирается оператором для завершения. Идентификатором пакета программ является его номер, который поступает на вход оператора «Запуск пакета программ». В соответствии с номером пакета оператор выбирает номер модуля локального диспетчера данного пакета (НМЛДі) и запускает его. Локальный диспетчер выполняет операторы «формированного связного списка».

В свою очередь оператор «формирование связного списка» позволяет записывать необходимую последовательность программ и корректировать ее в зависимости от ситуации. Оператор «управления запуском программ» осуществляет запуск программ и контроль их завершения. Следует отметить, что пакет программ включает программы одного приоритетного уровня. Состав пакета может меняться в зависимости от внешних воздействий на систему.

После обработки всех программ пакета запускается следующий по приоритету пакет и, только завершив обработку всех пакетов, стоящих в очереди, оператор «выбора приоритетной заявки» передает управление фоновой задаче, например тесту текущей проверки БЦВМ.

Для БЦВС с Ц101 и Ц102 микропрограммная реализация функций позволила достичь минимального времени выполнения операционной системы. Например, время от момента входа в операционную систему по прерыванию до запуска пакета программ в среднем составляет 45...50 мкс.

## **4.10. ОПЕРАЦИОННАЯ СИСТЕМА РЕАЛЬНОГО ВРЕМЕНИ МНОГОПРОЦЕССОРНОЙ БОРТОВОЙ ЦВМ Ц-154**

### **4.10.1. ОРГАНИЗАЦИЯ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА В МНОГОПРОЦЕССОРНОЙ ЦВМ**

В основу организации вычислительного процесса при проектировании ОС РВ БЦВМ Ц154 был положен принцип симметричной или однородной обработки программ во всех процессорах. При симметричной орга-

низации все процессоры идентичны, поэтому любая задача может обрабатываться любым из них. Процессор, ответственный в данный момент времени за реализацию системных функций, называется мониторным, другие процессоры называются рабочими. В каждый момент времени мониторным может быть только один процессор, благодаря чему предотвращаются конфликты при работе с глобальной системой информации. Основными функциями мониторного процессора являются обработка прерываний, выполнение программ операционной системы, контроль за состоянием рабочих процессоров и обеспечение функционирования вычислительной системы при отказах одного или двух процессоров. Процесс функционирования распределенной ОС БЦВМ 154 иллюстрируется на рис. 4.28.

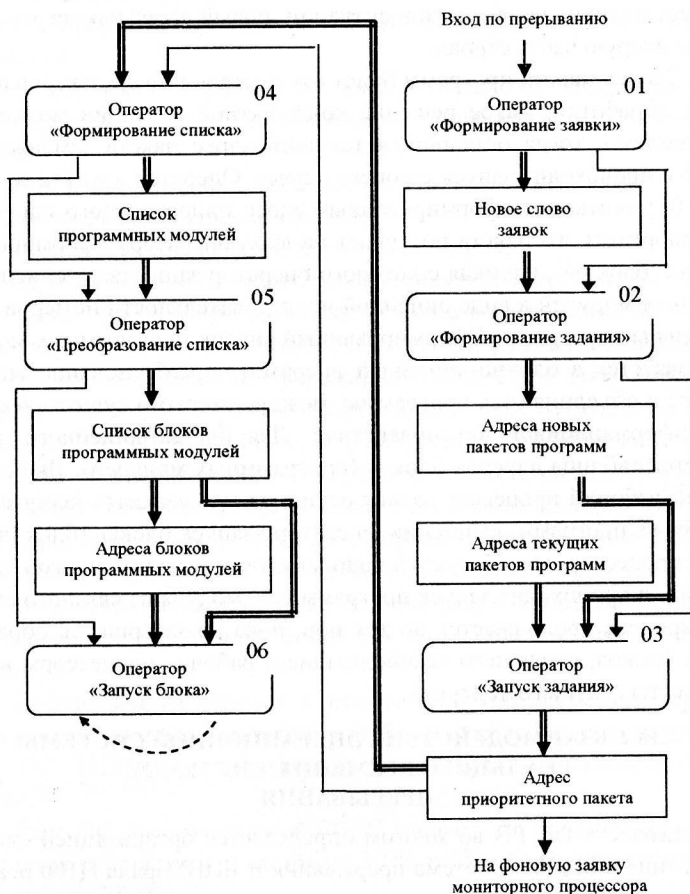


Рис. 4.28

Функцией рабочих процессоров является выполнение функциональных программ (запуск и контроль) и организация межпроцессорного обмена. Запуск программ выполняет оператор «запуск блока».

Обработка заявок, сформированных по сигналам прерывания или программным путем, начинается в мониторном процессоре с формирования приоритетной очереди заявок («новое слово заявок», см. рис. 4.28 блок 01).

В результате обработки «нового слова заявок» оператором «формирование задания» (блок 02) формируется таблица адресов пакетов программ. Строка таблицы состоит из двух частей, каждая из которых заполняется новым или текущим адресом пакета программных модулей. Если первая часть строки занята текущим адресом пакета программ, что означает наличие конфликтной ситуации, новый адрес пакета записывается во вторую часть строки.

Запуск пакета программ будет осуществлен после завершения текущей обработки. Такое решение конфликтной ситуации может быть неприемлемо, тогда оставшийся текущий адрес пакета заменяется на новый и происходит запуск с нового адреса. Оператор «запуск задания» (блок 03) считывает сформированный адрес приоритетного пакета, адрес приоритетного пакета поступает на вход оператора «формирование списка» (блок 04), начиная с которого оператор записывает список программных модулей в виде линейной последовательности номеров функциональных программ. Сформированный список программных модулей преобразуется в блочно-линейный оператор «преобразование списка». В блоке и объединяются программы, между которыми существует сильное информационное взаимодействие. Для блочно-линейного списка создается таблица адресов блоков (программных модулей). Любой свободный рабочий процессор может осуществлять «захват» необработанного блока программ, выполнив оператор «запуск блока» (блок 06). Рабочий процессор преобразует линейный список «захваченного» блока в связный, и происходит запуск программного модуля из связного списка. Этот процесс продолжается до тех пор, пока не завершится обработка списка блоков, после чего освободившиеся рабочие процессоры выполняют тесты текущего контроля.

#### **4.10.2. ВЗАИМОДЕЙСТВИЕ ОПЕРАЦИОННОЙ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ С СИСТЕМОЙ ПРЕРЫВАНИЯ**

Эффективность ОС РВ во многом определяется организацией системы прерывания в БЦВМ. Система прерывания В БЦВМ ряда Ц100 реализована на микрокомандном уровне с возможностью включения в нее от-

дельных функций операционной системы, наиболее тесно связанных с обработкой прерывания в БЦВМ.

Микропрограмма организации прерывания осуществляет идентификацию типа прерывания, запоминание состояния процессора на момент прерывания исполняемой программы, передачу управления на программу или микропрограмму обработки принятого запроса на прерывание.

Микропрограмма вызывается процессором после того, как им будет принят очередной запрос на прерывание. Прерывание допускается в любой момент исполнения программы, если операционная система не выставила блокировку того или иного типа прерывания. В основном обработка прерывания выполняется самой микропрограммой входа в прерывание.

Остановимся подробнее на данной обработке. Микропрограмма входа в прерывание от таймера выполняет следующие дополнительные функции:

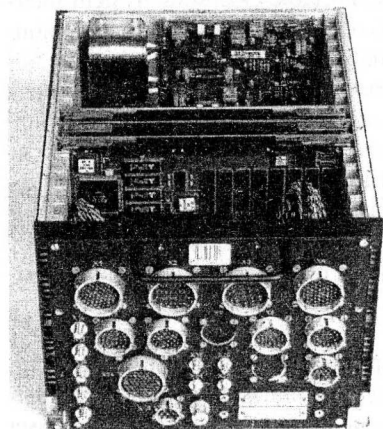
- контроль признаков исправности,
- контроль прохождения тестов,
- контроль смены режима тестовых проверок,
- идентификацию штатного режима текущей самопроверки.

Таким образом, при организации входа в прерывание по таймеру контролируется состояние БЦВМ, и если обнаружена неисправность, микропрограмма обработки прерывания сообщает об этом операционной системе. Микропрограмма входа в прерывание осуществляет перезагрузку БЦВМ, если ОС РВ не взяла функции обработки неисправности на себя, обеспечив реакцию операционной системы на информацию, поступающую от системы прерывания о состоянии БЦВМ. После этого можно продолжить вычислительный процесс при наличии сбоев без перезапусков БЦВМ. Если в ходе вычислительного процесса продолжают появляться сбои, например три подряд, ОС БЦВМ формирует отказ и передает управление тестам для диагностики и выявления неисправности.

Предложенный подход к построению и реализации бортовой операционной системы позволил разработать набор базовых операторов (ядро ОС БЦВС), настраиваемых на конкретный вычислительный процесс.

Ядро ОС многомашинной БЦВС требует для размещения около 2 Кбайт памяти программ и занимает 5-6% такта обработки информации (такта работы БРЛС).

# Программное обеспечение БЦВМ «Багет»



Программное обеспечение содержит специализированное ПО реального времени, загрузчик в РПЗУ; программно-технологический комплекс кроссразработки и отладки прикладных программ .

## 5.7. КОНЦЕПЦИЯ МОБИЛЬНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Концепция мобильности прикладного ПО ЭВМ семейства «Багет», суть которой состоит в возможности переноса базового и прикладного ПО на различные аппаратные платформы, основывается на мировых стандартах в области технологии создания базового и прикладного ПО. Основы этого направления были заложены в начале 70-х гг. и связаны с появлением первой версии ОС UNIX как прототипа единых, унифицированных ОС и унифицированной технологии разработки прикладного ПО для ЭВМ различной архитектуры. Существенно, что ОС UNIX была задумана и реализована сообществом пользователей ЭВМ прежде всего как альтернатива "фирменным" ОС, жестко привязывавшим потребителя к конкретным фирмам-изготовителям ЭВМ и ПО. В течение весьма небольшого времени ОС UNIX была портирована на все наиболее известные в то время архитектуры ЭВМ. Постепенно практически все ведущие фирмы-изготовители разработали собственные варианты UNIX, присвоив им уникальные названия; компания IBM — AIX, компания Sun — Solaris, компания Hewlett-Packard — HP-UX, компания Silicon Graphics — IRIX и т.д., что послужило причиной немобильности прикладного ПО, разрабатываемого для разных вариантов UNIX.

Дальнейшее развитие основной концепции ОС UNIX привело к созданию международного стандарта (ISO/IEC 9945— мобильный ин-

терфейс операционной системы) POSIX (Portable Operating System Interface), специфицирующего интерфейс (сопряжение) между операционной системой и прикладной программой.

Основной механизм POSIX — интерфейсная функция, которая вызывается из прикладной программы, а обрабатывается операционной системой. Стандартом специфицирован ассортимент, имена, синтаксис и семантика интерфейсных функций. Как и любой стандарт на интерфейс, POSIX содержит два вида требований: требования к операционной системе и требования к прикладной программе. Требования POSIX к операционной системе были реализованы при разработке ОС2000, выполнение требований POSIX к прикладной программе — обязанность прикладного программиста.

Мобильность прикладных программ обеспечивается, согласно идеологии POSIX, механизмом конфигурационных констант. Требования POSIX заключаются в том, что мобильная прикладная программа обязана опрашивать характеристики ОС, в которой исполняется (задаваемые, в частности, конфигурационными константами), а операционная система обязана предоставлять соответствующие средства опроса.

В основу концепции развития базового программного обеспечения положено понятие открытой системы в соответствии с определением комитета IEEE POSIX 1003.0, что в перспективе обеспечит совместную работу прикладных программ на локальных и удаленных платформах, а также минимально возможные сроки и стоимость как переноса базового и прикладного ПО на различные аппаратные платформы (мобильность программ), так и перехода пользователей от одной прикладной системы к другой (мобильность пользователей).

К настоящему времени разработана первая версия проекта профиля стандартов аппаратно-программной платформы «Багет» (сокращенно — профиль стандартов «Багет»). Этот профиль основан на следующих стандартах.

*Стандарт языка программирования С (ISO/IEC 9899).* Помимо собственно языка С этот стандарт специфицирует номенклатуру, синтаксис и семантику библиотечных функций времени исполнения.

*Стандарт на интерфейс прикладных программ (API — интерфейс прикладных программ) с операционной системой, POSIX (ISO/IEC 9945).* Этот стандарт специфицирует номенклатуру, семантику и синтаксис вызовов операционной системы.

В профиль стандартов «Багет» включены те функции POSIX, которые относятся к однопользовательским операционным системам реального времени. Работа нескольких процессоров в одном общем адресном пространстве (сильно связанные системы) профилем стандартов «Багет» не предусматривается.



*Графический стандарт de facto X Window версии 11.* Этот стандарт специфицирует интерфейс графических прикладных программ со специальной программой, управляющей графическим оборудованием и называемой «дисплейным сервером». Дисплейный сервер может быть реализован в рамках операционной системы, удовлетворяющей профилю стандартов «Багет», так что данный стандарт служит естественным дополнением POSIX для графического ПО.

*Сетевой стандарт de facto TCP/IP версии 4.* Этот стандарт специфицирует семейство протоколов обмена данными между ЭВМ, входящими в сети TCP/IP.

Операционная система реального времени OC2000 соответствует профилю стандартов «Багет». OC2000 реализована для процессорных архитектур Intel×86 и IB812. К наиболее важным отличительным особенностям этой системы относятся следующие:

реализация интерфейса прикладных программ операционной системы в соответствии со стандартом POSIX;

обеспечение многозадачности системы потоками управления (средства реального времени POSIX);

использование однопользовательской системы;

при компоновке прикладной программы включение лишь некоторых функций операционной системы (масштабируемость).

В данном разделе концепция АПП «Багет» рассмотрена только частично и на общем техническом уровне. Не приведены материалы по концепции аппаратных и программных средств «Багет» при построении высокопроизводительных мультипроцессорных вычислительных систем и серверов. Это направление в настоящее время является чрезвычайно актуальным, так как мировой опыт построения высокопроизводительных систем основан на использовании многопроцессорных архитектур.

Следует отметить, что АПП «Багет» постоянно находится в развитии как в части создания более быстродействующей аппаратной платформы с применением модулей с увеличенной рабочей частотой, так и в части разработки новых архитектурных решений. При этом наибольшие усилия направлены на создание новых быстродействующих интерфейсных модулей (в том числе сетевых).

## ЛИТЕРАТУРА К ГЛАВЕ 5

1. Багет. Семейство ЭВМ для специальных применений. – М.: КБ «Корунд-М», 2004.
2. Корнеев В.В., Киселёв А.В. Современные микропроцессоры. Издание третье, дополненное и переработанное – СПб.: БХВ – Петербург, 2003.
3. Меркулов В.И., Дрогалин В.В., Канащенков А.И. и др. Авиационные системы радиоуправления. Т.2. Радиолектронные системы самонаведения. /Под ред. А.И. Канащенкова и В.И. Меркулова. – М.: Радиотехника, 2003.
4. Рыбаков А.Н. Открытые компьютерные COST-технологии в военных приложениях. – МКА, 1999, №4.