Тема 5. Операционная система (ОС)

5.1. Место и роль ОС в составе ПО ЦСК

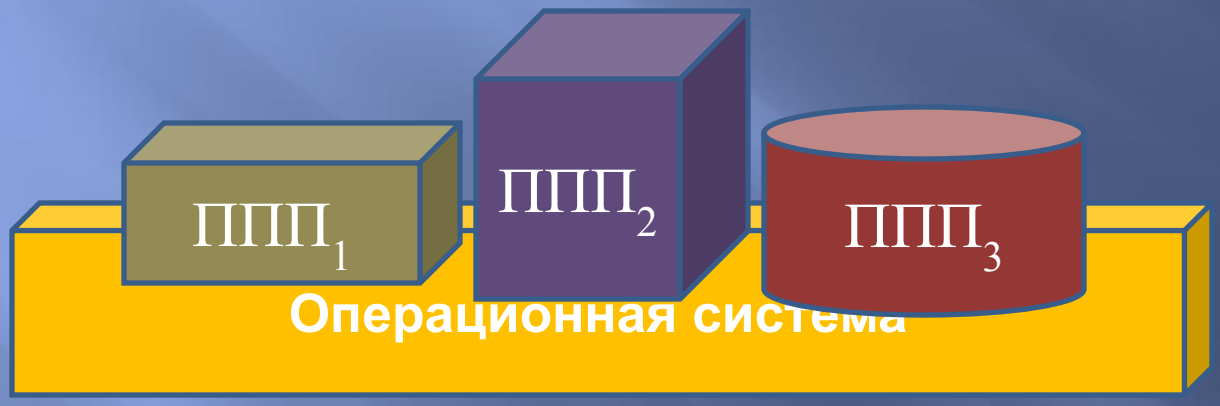
В составе внутреннего ПО ЦСК операционная система относится к ***системному ПО***.

При этом остальные компоненты внутреннего ПО (т.е. СКП, СПТО и САдмП) можно отнести к ***прикладным программам***.

Следовательно, в общих чертах положение, назначение и функции ОС в составе программных средств ЦСК не имеют существенных отличий от типовой АСУТП с использованием технологий программного управления.

Хотя специфика управления коммутационным оборудованием накладывает некоторый отпечаток.

Применительно к современной вычислительной технике операционную систему вместе с другим средствами системного ПО часто называют ***фундаментом*** (платформой) для прикладных программ.



ППП – пакет прикладных программ

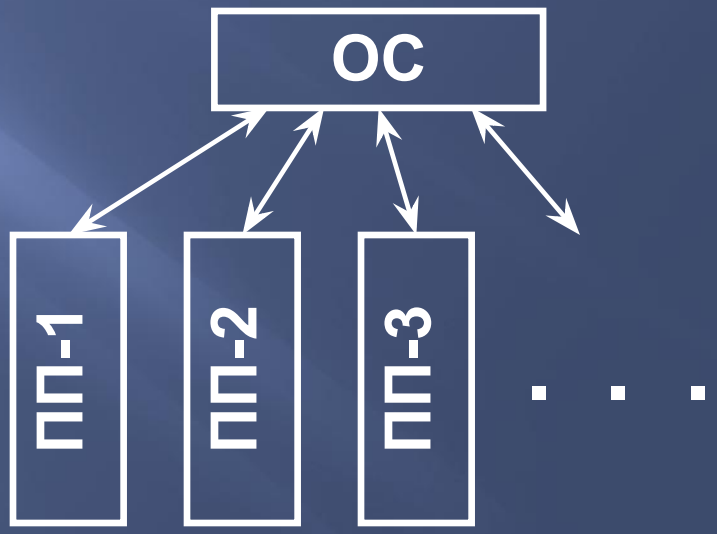
Здесь имеются ввиду две группы функций:

- управление взаимодействием отдельных устройств компьютера (или вычислительной системы);
- организация обмена данными между внешними устройствами («компьютерная периферия») и оперативной памятью ЭВМ

При наличии этих функций в составе ОС обеспечивается:

- эффективное использование вычислительных ресурсов;
- предоставление широкого набора стандартных функций ввода/вывода для прикладных программ.

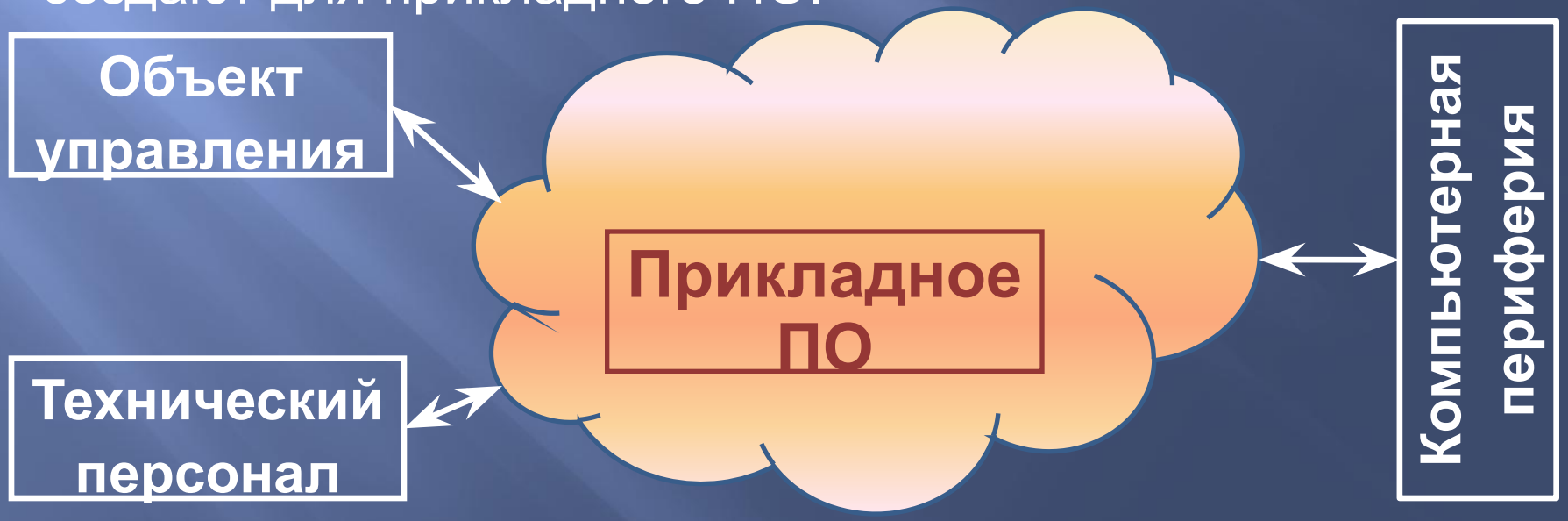
С другой стороны, при использовании ЭВМ для управления сложными объектами операционная система выполняет также важную функцию ***координации*** совместного (параллельного) решения многочисленных прикладных задач.



ПП –прикладная программа

При активном взаимодействии системы управления с обслуживающим персоналом важную роль играют функции ОС по поддержке ***диалога «человек-машина»***.

Если совместно рассматривать все перечисленные функции ОС, то можно говорить о ***стандартной среде***, которую ОС вместе с другими средствами системного ПО создают для прикладного ПО.



5.2. Структура операционной системы

1) Программы организации вычислительного процесса (управляющие программы):

- программы распределения процессорного времени (диспетчеры или планировщики);
- программа распределения оперативной памяти (менеджер памяти)

2) Программы информационного обмена между прикладными процессами
3) Программы ввода/вывода (драйверы) – для управления обменом данными между внешними устройствами и оперативной памятью
4) Программы связи «человек-машина»
5) Программы загрузки, перезагрузки и начального

5.3. Требования к организации вычислительного процесса в системе управления ЦСК

ПО ЦСК осуществляет поддержку достаточно большого количества разнообразных функций:

1) обработка вызовов (основная функция);
2) автоматизация эксплуатационно-технического и административного управления (вспомогательные функции).

По этой причине, программные средства для ЦСК емкостью 10 тыс. абонентов на интервале ЧНН должны ***одновременно*** (***параллельно***) держать под контролем:

- ≈ 1 тыс. соединений в фазе разговора;
- ≈ 200 соединений на этапах установления или разрушения;

Одновременно с этим действуют ***жесткие ограничения по времени реакции*** программ:

- при обработке сигнализации — ≤0,01÷0,1 сек;
- для функций обработки вызова — ≤0,1÷1 сек;
- для диалога «человек-машина» — ≤1÷3 сек;
- для операций технической эксплуатации — ≤1÷10 сек.

Источник: Гольдштейн Б.С. Системы коммутации: Учебник для вузов. — СПб., 2004.

При ***централизованном управлении*** большинство перечисленных задач возлагается на ***центральный процессор***.

Отсюда вытекает основное требование к этому процессору — обеспечивать эффективное

Многопрограммное функционирование организуется путем ***оперативного разделения во времени*** вычислительных ресурсов процессора — машинное время процессора.

При ***распределенном управлении*** с применением микропроцессоров такой же режим работы имеет место в модульных устройствах управления (МУУ), хотя общее число активных задач для отдельного МУУ гораздо ниже.

Таким образом, независимо от общей структуры управления в ЦСК, организация вычислительного процесса по совместному решению задач обслуживания вызовов и других вспомогательных задач требует ***оперативного распределения ограниченных вычислительных ресурсов*** между этими задачами.

5.4. Принципы организации вычислительного процесса в системе управления ЦСК

Чтобы обеспечить многопрограммный режим реального времени, применяется два основных принципа:

- запуск программ по ***заявкам***, т.е. программа может получить управление только при наличии заявки на ее выполнение;
- принцип ***приоритетности*** заявок.

Заявки, по которым осуществляется запуск программ, обычно имеют ***разную природу*** (происхождение):

- программные заявки, формируемые одними программами для запуска других программ;
- аппаратные сигналы прерывания от датчика времени (так называемые метки времени);
- аппаратные сигналы прерывания от внешних устройств и другого аналогичного оборудования;

Заявкам могут присваиваться разные ***приоритеты*** в зависимости от допустимых задержек при их обработке.

В случае ***абсолютного приоритета*** более важные заявки ***прерывают*** обработку заявок более низкого приоритета.

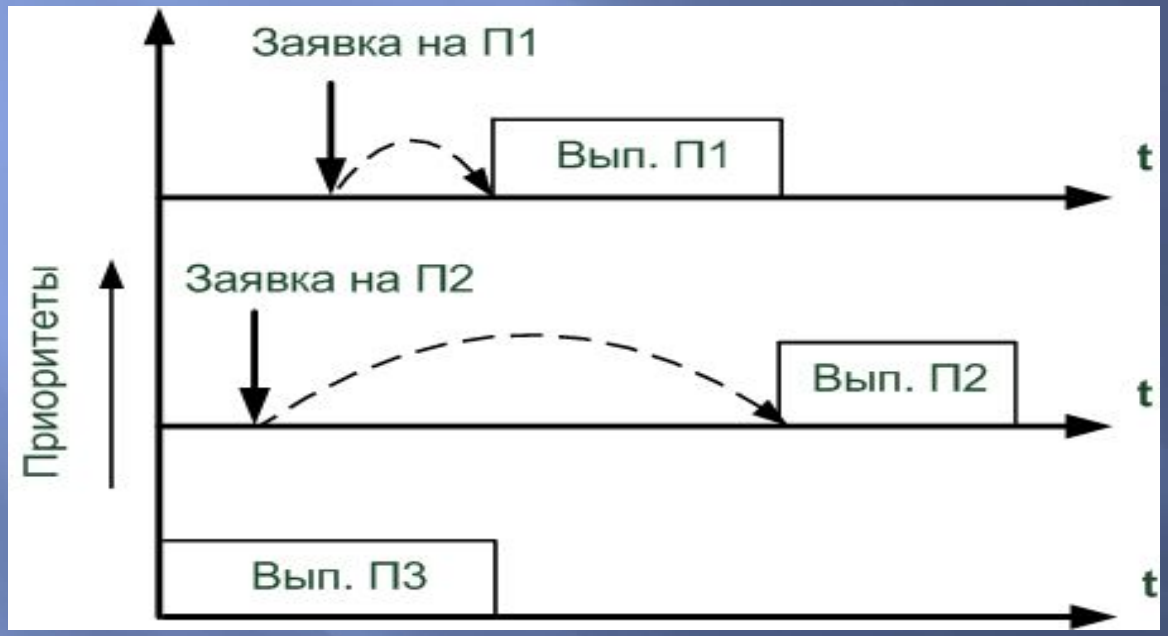
Следовательно, задержки при обработке приоритетных заявок ***снижаются***.

Для прерываемой программы необходимо:

- запомнить ***адрес точки прерывания***;
- сохранить ***текущее состояние программы***, т.е. содержимое ее стека (область рабочей памяти) и регистров процессора.

В дальнейшем это позволяет продолжить выполнение программы с точки прерывания, а не с самого начала.

В случае ***относительных приоритетов*** прерывания не происходит, и обслуживание более приоритетной заявки начинается только после завершения текущей программы.



В очереди на ожидание начала обслуживания заявки располагаются согласно их приоритетов.

Реализация относительного приоритета проще (т.к. нет прерываний), но здесь нельзя исключить ***задержки*** даже для самых важных заявок.

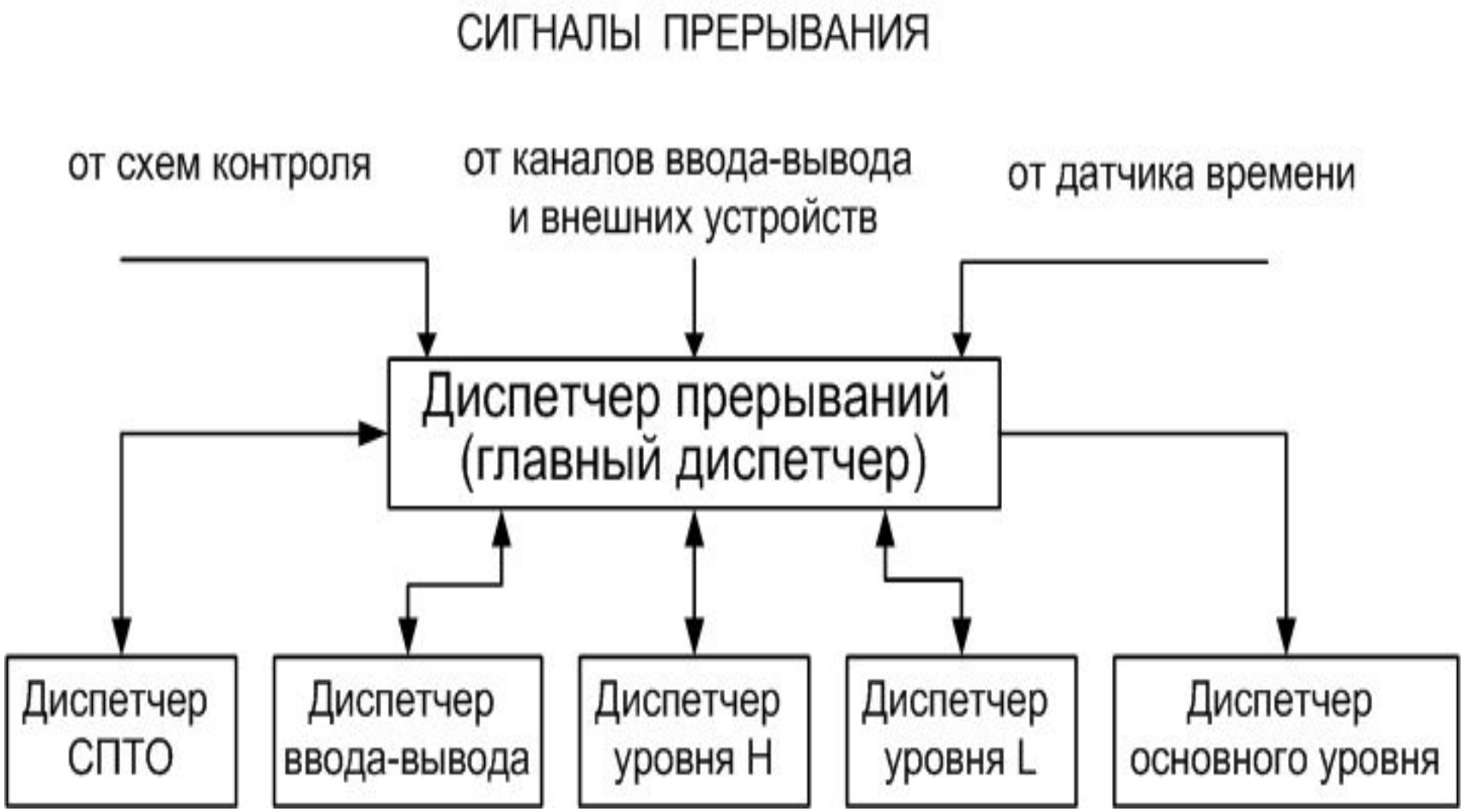
Смешанный приоритет – это комбинация (объединение) абсолютных и относительных приоритетов вместе с их достоинствами и недостатками.

Все программы разделяются на несколько групп, которые называют ***приоритетными уровнями***.

Между программами разных приоритетных уровней действует абсолютный приоритет, а внутри каждого уровня применяется относительный приоритет.

На практике, т.е. в зарубежных и отечественных системах коммутации с программным управлением, такая дисциплина обслуживания заявок получила ***наиболее широкое распространение***.

Управляющие программы для организации многопрограммного режима реального времени



Главный диспетчер (или диспетчер прерываний) анализирует приоритеты заявок, поступающих в виде аппаратных сигналов прерывания.

После анализа он производит запуск одного из подчиненных диспетчеров более низкого уровня, и этот диспетчер должен организовать ***непосредственную обработку*** поступившего сигнала.

Каждый подчиненный диспетчер управляет своей группой программ, которые образуют отдельный ***приоритетный уровень***.

Программы, относящиеся к диспетчеру СПТО, имеют ***наивысший абсолютный приоритет*** и запускаются по ***аварийным сигналам*** от схем аппаратного контроля.

Диспетчер ввода-вывода управляет программами для обработки сигналов прерывания от каналов ввода-вывода

Приоритетный уровень *H* объединяет ***периодические программы***, которые запускаются по сигналу прерывания от датчика времени и требуют ***строгого соблюдения*** периодичности запуска.

Приоритетный уровень *L* выделен для ***периодических программ***, которые также запускаются по сигналу от датчика времени, но ***допускают небольшие отклонения*** от заданной периодичности запуска.

Диспетчер ***основного уровня*** управляет программами, которые имеют самый низкий приоритет.

Для этих программ задержки при запуске и прерывания при выполнении не приводят к каким-либо заметным негативным последствиям.

Главный диспетчер (или диспетчер прерываний), который регулирует переходы между разными приоритетными уровнями, имеет ***два режима работы***:

1) режим прерывания;
2) режим восстановления.

Режим прерывания начинается с момента поступления нового сигнала прерывания, у которого приоритет выше, чем приоритет выполняемой программы.

В этом случае главный диспетчер сохраняет для прерываемой программы все промежуточные результаты, которые получены к текущему моменту, а также запоминает адрес точки прерывания.

Затем управление передается диспетчеру того приоритетного уровня, на котором будет происходить

Главный диспетчер переходит в ***режим восстановления*** при получении управления от одного из подчиненных диспетчеров после завершения всех запланированных вычислительных работ на этом приоритетном уровне.

Основная задача для этого режима – выбрать другой приоритетный уровень, программы которого требуют запуска.

При выборе анализируются следующие заявки:

- в виде сигналов прерывания, которые ожидают начала обработки;
- из очереди прерванных программ.

Среди всех этих вариантов предпочтение получает заявка с самым высоким приоритетом.

Вариант 1. Если выбрана заявка из очереди прерванных программ, то главный диспетчер восстанавливает ее состояние на момент прерывания.

После этого управление передается на команду, которая следует сразу за точкой прерывания.

Вариант 2. При выборе очередного сигнала прерывания никаких восстановительных действий не требуется и управление сразу получает подчиненный диспетчер, который должен организовать обработку этого сигнала.

5.5. Организация запуска программ по расписанию

На приоритетном уровне ***H*** располагаются программы, которые должны выполняться ***строго периодически***.

Для уровня ***L*** реальный период запуска может иметь незначительные отклонения от заданного значения.

Реализация этих требований достигается запуском программ по ***меткам времени*** (МВ) – специальным сигналам прерывания от датчика времени.

Диспетчер приоритетного уровня (*H* или *L*), получив управление от диспетчера прерываний при поступлении очередной МВ, запускает программы этого уровня согласно ***расписанию***, которое хранится в памяти

При работе по расписанию ось времени делится на дискретные интервалы, которые называют ***первичными периодами***.

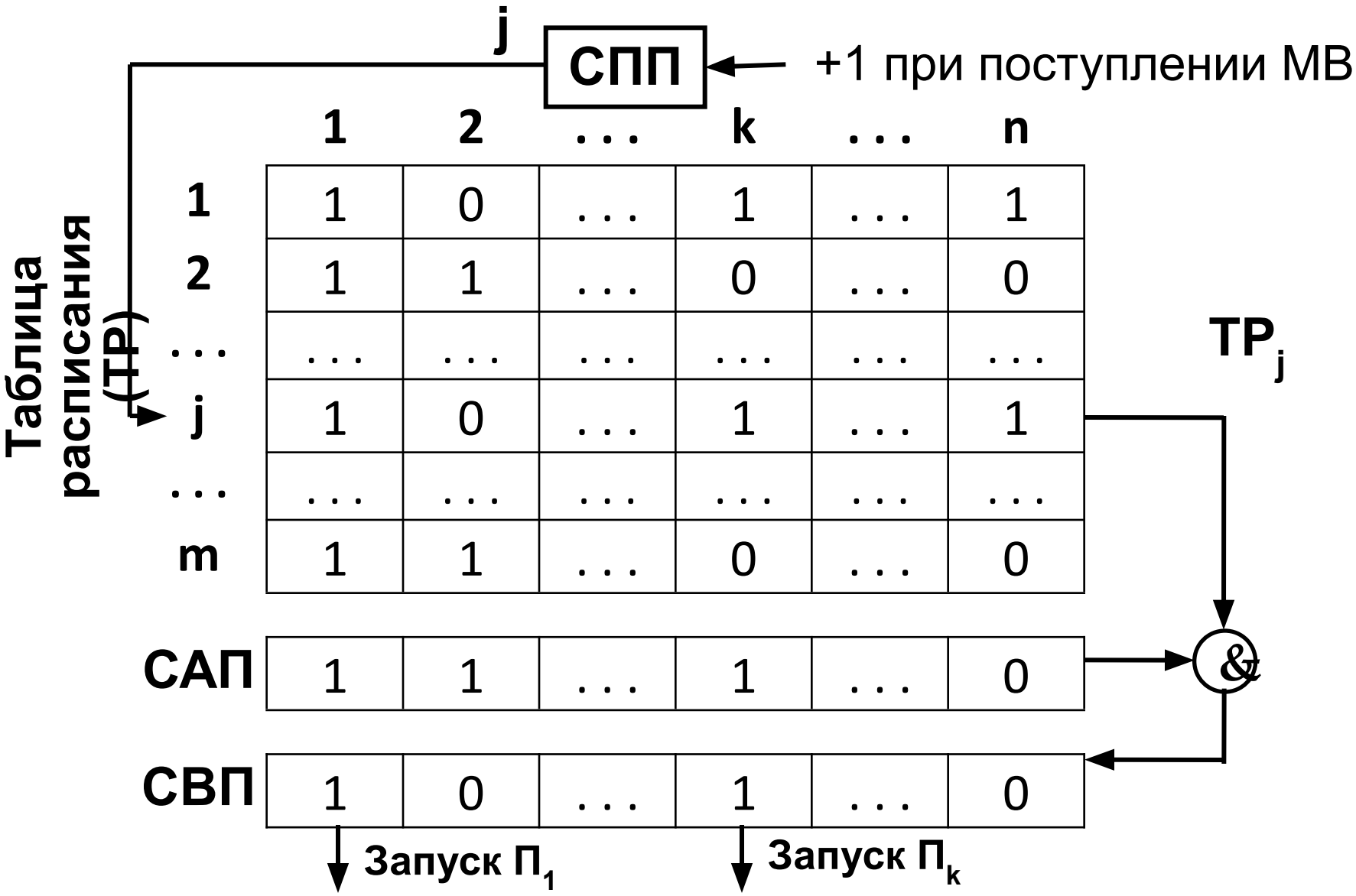
Длина первичного периода равна периоду поступления меток времени – обычно 5÷10 мс.

Расписание представляет собой таблицу, в которой номера столбцов соответствуют номерам программ, а номера строк — номерам первичных периодов в текущем цикле реализации расписания.

Каждая строка таблицы заполняется ***двоичными элементами***.

Если j-я строка в позиции k-го столбца содержит значение 1, то в j-м первичном периоде ***требуется запуск***

Схема запуска программ по расписанию



Если некоторый столбец таблицы расписания содержит только значения 1, то соответствующая программа будет запускаться ***в каждом*** первичном периоде.

Чтобы понизить частоту запуска некоторой программы (т.е. увеличить длительность периода запуска), нужно в соответствующий столбец добавить значения 0.

Требование равномерности расписания: число запускаемых программ в каждом первичном периоде (т.е. число значений 1 в каждой строке) должно быть ***одинаковым*** (хотя бы ***приблизительно***).

Номер текущего первичного периода в очередном цикле реализации расписания задается ***счетчиком первичных периодов*** (СПП).

Этот счетчик наращивается на единицу при поступлении очередной метки времени.

Чтобы обеспечить ***циклический характер*** расписания (т. е. последовательное повторение циклов реализации расписания), содержимое СПП изменяется по формуле $j:=(j+1) \bmod m$.

Слово активности программ (САП) используется для того, чтобы ***запретить*** выполнение некоторых из них. Например, в условиях перегрузки можно запретить (заблокировать) текущее тестирование оборудования. Если k-я программа ***разрешена*** к выполнению, то k-й разряд в слове САП должен содержать ***значение 1***.

Реальный перечень программ, которые должны быть запущены в j-м первичном периоде, определяется ***словом выполняемых программ***:

$$СВП := ТР \ \& \ САП$$

Особенности выполнения программ на основном уровне

На основном уровне некоторые из программ также запускаются по своему расписанию.

Однако, в отличие от приоритетных уровней *H* и *L*, эти программы могут выполняться только на случайных по длительности интервалах времени, которые остаются после выполнения более важных программ.

Переход к следующей строке расписания происходит не по очередной метке времени, а сразу после окончания работы с текущим СВП.

Поэтому расписание основного уровня задает периодичность (частоту) запуска программ только в

5.6. Организация вычислительного процесса по обработке вызовов в ЦСК

Чтобы реализовать многопрограммный режим реального времени при обслуживании поступающих вызовов, разделим все программы, участвующие в этом процессе, на ***две группы***:

1)***Приоритетные программы*** (высокая срочность):

- ❖программы ввода информации (приема сигналов);
- ❖программы вывода информации (выдачи периферийных команд)

2)Программы основного уровня (***без приоритетов***):

- ❖программы анализа сигналов и обработки данных;
- ❖программы формирования периферийных команд (ПК).

Функциональная схема многопрограммной обработки вызовов в режиме реального времени

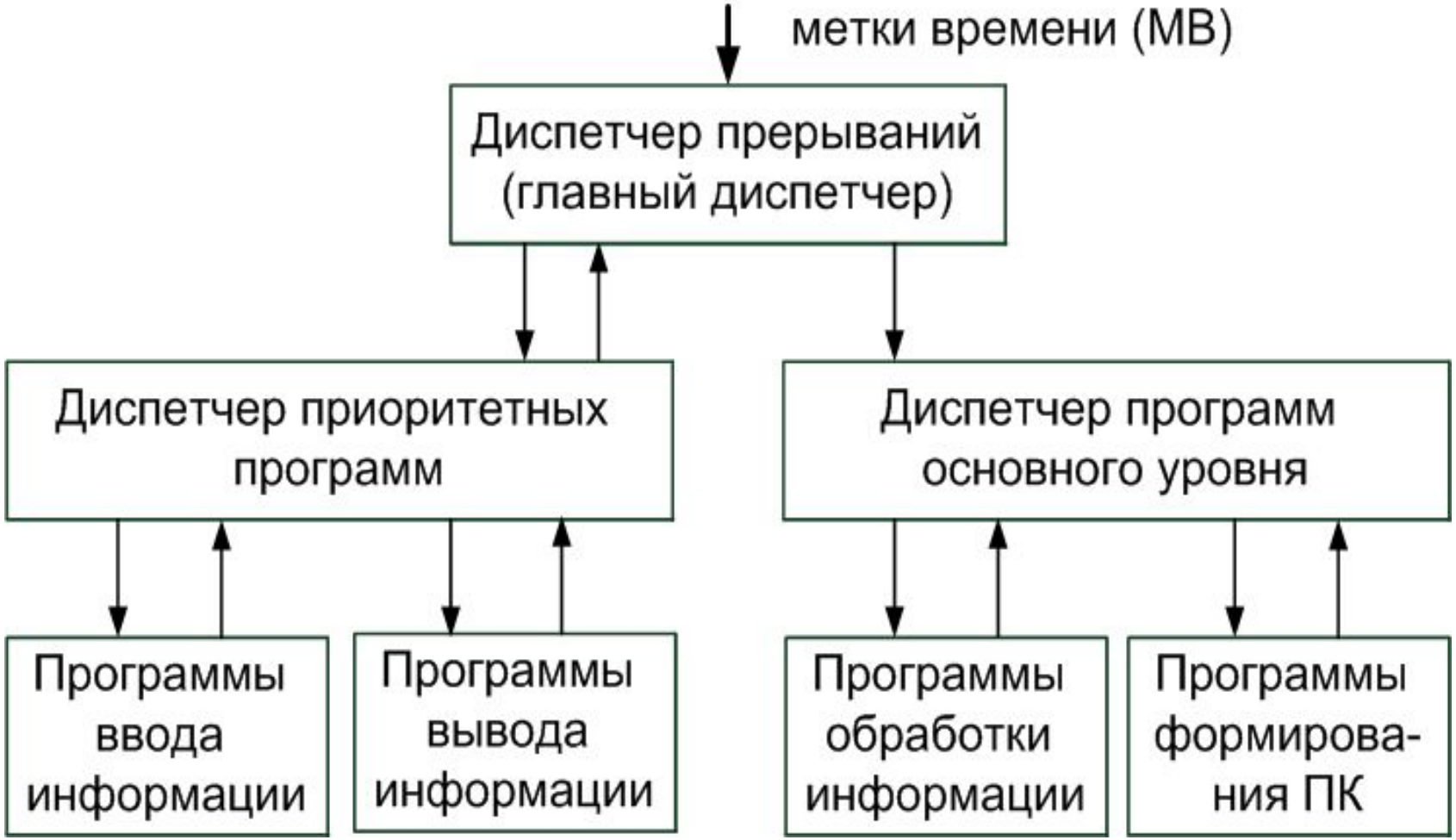
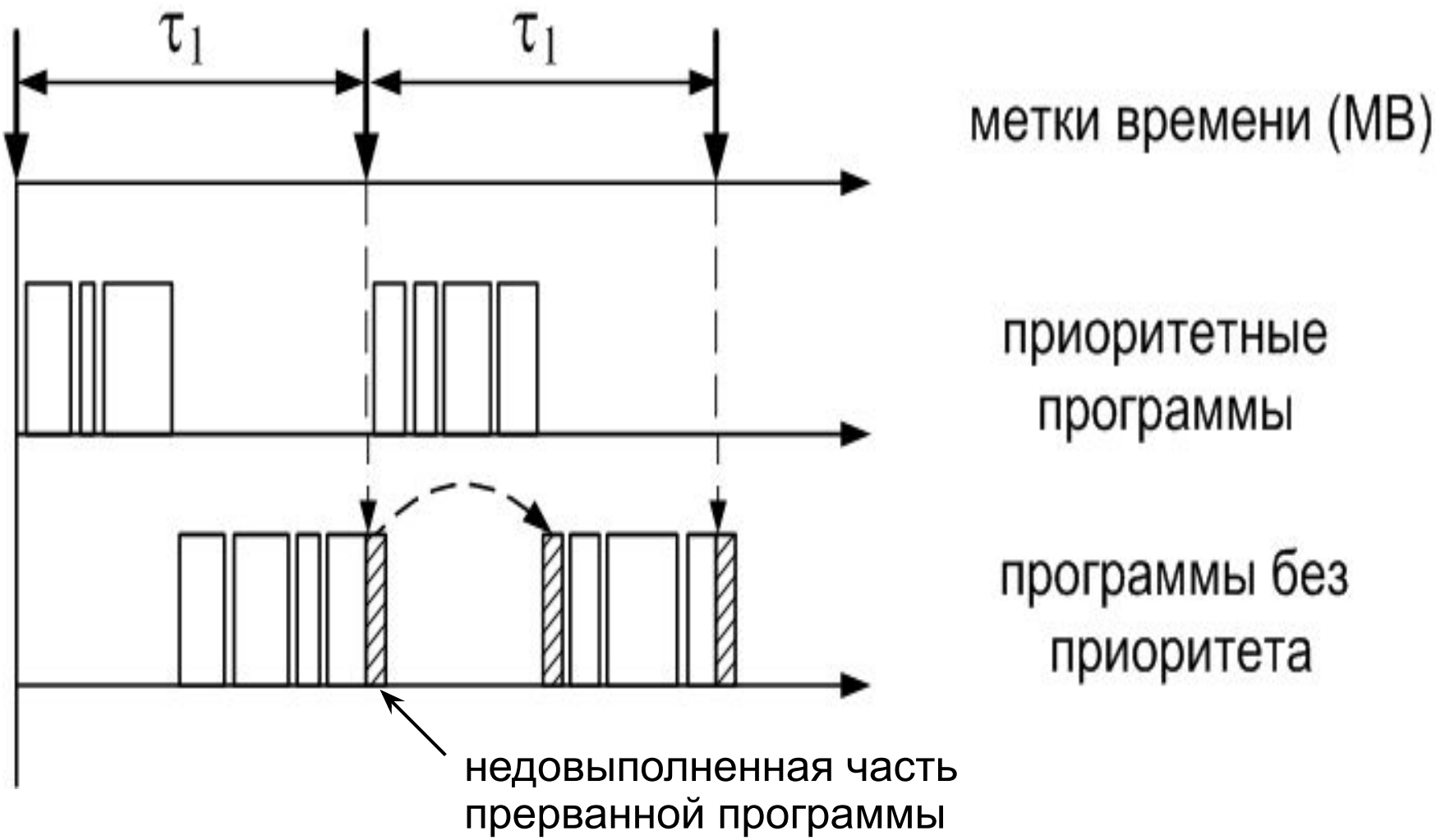


Диаграмма выполнения программ обслуживания вызовов



5.7. Организация информационного обмена между прикладными процессами

В системах с распределенным управлением (например, S-12) фактически присутствует локальная сеть специализированных микро-ЭВМ.

При решении многих задач обработки вызовов эти микро-ЭВМ (модульные устройства управления – МУУ) должны взаимодействовать друг с другом.

Следовательно, при такой структуре системы управления необходимы еще и функции «***почтовой службы***».

Список этих функций, которые выполняет ОС в каждом МУУ, включает в себя:

- формирование и отправка сообщений по заявкам (запросам) от прикладных процессов (выполнение оператора SEND);
- выделение прикладным процессам специальных буферов для приема и хранения поступающих сообщений;
- передача прикладному процессу требуемого сообщения из буфера (выполнение оператора RECEIVE);
- реализация заданной дисциплины обслуживания сообщений, поступающих в буфер (например,