Федеральное агентство связи

Сибирский государственный университет телекоммуникаций и

 информатики

Костюкович Н.Ф.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЦИФРОВЫХ СИСТЕМ КОММУТАЦИИ

методические указания

к контрольной работе

Новосибирск-2011

## Методические указания к контрольной работе

1. Задачи рекомендуется решать в том же порядке, который указан в задании.
2. Перед решением каждой задачи необходимо написать ее условие из контрольного задания, исходные данные должны со­ответствовать Вашему варианту.
3. Не допускайте, пожалуйста, небрежного выполнения рисунков и пояснений.
4. Рисунки необходимо нумеровать, а также ссылаться на них в пояснениях.
5. В конце работы привести список источников, которыми Вы пользовались, дату окончания работы и подпишите ее.
6. Работу необходимо выполнить в отдельном файле.
7. Рецензирование работы выполняет кафедра АЭС СибГУТИ.
8. Если в работе имеются отмеченные рецензентом ошибки, их необходимо исправить, выполнив решение всей или части задачи на свободных страницах файла шрифтом синего цвета.
9. После этого работу нужно снова отправить в СибГУТИ.
10. Работа должна быть зачтена прежде, чем Вы будете сдавать зачет по 1-й части курса.
11. В контрольную работу включено четыре задачи, которые охватывают важнейшие разделы 1-й части курса.

  **ЗАДАНИЕ К КОНТРОЛЬНОЙ РАБОТЕ**

В контрольной работе имеется несколько вариантов. Номер варианта для выполняемого магистрантом задания должен соответствовать последней цифре пароля.

**Задача 1**

Изобразить схему алгоритма приема информации о новых вызовах в СКПУ (программ ПСК1 и ПСК2). Привести пример обработки данных в процессе приема, используя исходные данные из табли­цы 1. Запишите заявки в буфер предварительных заявок (БПЗ) и буфер заявок для обработки новых вызовов (БЗО). Нумерация оконечных устройств начинается с правого нулевого разряда в нулевой группе (К=0).

Обозначения в таблице 1:

* СОС1 - слово очередного сканирования один;
* СОС2 - слово очередного сканирования два;
* СПС - слово предыдущего сканирования;
* К – номер линейки.

При организации процесса сканирования необходимо предусмотреть защиту от помех. Для этого используется повторное сканирование. Повторное сканирование выполняется со сдвигом во времени относительно первичного сканирования, и опрашиваются только те линейки, где выявились изменения. Совпадения результатов первичного и повторного сканирования свидетельствует о поступлении действительно входного сигнала. Такое сравнение обеспечивается благодаря хранению предварительных результатов первичного сканирования в так называемом буфере предварительных заявок (БПЗ).

Функции ввода информации выполняются программами сканирования двух видов - ПСК1 и ПСК2. Структурная схема взаимодействия этих программ с массивами данных и аппаратными средствами приведена на рисунке 1.



Рисунок 1- Взаимодействие программ ПСК1 и ПСК2

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| № Вар. | МС АК (СПС) | СОС1 | СОС2 |
| 1 | 0 1 2 3 4 50 0 1 1 0 0 1 1 1 0 0 0 1 0 2 0 1 0 0 0 1 | 0 1 2 3 4 50 0 1 0 0 0 1 1 1 0 0 0 1 0 2 0 0 1 1 0 0 | 0 1 2 3 4 50 0 1 1 0 1 1 1 1 0 0 0 1 0 2 0 0 1 1 0 0 |
| 2 | 0 1 2 3 4 50 0 1 1 0 1 0 1 1 0 1 0 1 0 2 0 0 0 0 0 1 | 0 1 2 3 4 50 0 0 0 0 0 1 1 1 0 1 0 1 0 2 0 0 1 1 0 0 | 0 1 2 3 4 50 0 1 1 0 0 1 1 1 0 0 0 1 0 2 0 0 1 1 0 0 |
| 3 | 0 1 2 3 4 50 0 0 0 0 0 1 1 1 0 0 0 1 1 2 0 1 0 0 0 1 | 0 1 2 3 4 50 0 1 0 0 0 1 1 1 0 0 1 1 0 2 0 0 1 1 0 0 | 0 1 2 3 4 50 0 1 1 0 1 1 1 1 0 0 1 1 0 2 0 1 0 0 0 1 |
| 4 | 0 1 2 3 4 50 1 0 1 0 0 0 1 1 1 0 0 0 0 2 0 1 0 1 0 1 | 0 1 2 3 4 50 0 1 0 0 0 1 1 1 0 0 1 1 0 2 0 0 1 1 0 0 | 0 1 2 3 4 50 0 1 1 0 0 0 1 1 0 0 1 1 0 2 0 0 0 1 0 0 |
| 5 | 0 1 2 3 4 50 0 0 0 1 1 1 1 1 0 0 0 1 1 2 1 1 0 0 0 1 | 0 1 2 3 4 50 1 0 0 0 0 1 1 1 0 0 1 1 0 2 1 1 0 0 0 0 | 0 1 2 3 4 50 0 1 1 0 1 1 1 1 0 0 0 1 1 2 0 0 1 1 0 0 |
| 6 | 0 1 2 3 4 50 0 0 0 0 0 0 1 1 1 0 1 1 0 2 0 1 0 1 1 1 | 0 1 2 3 4 50 0 1 0 1 0 1 1 1 0 0 1 1 0 2 0 0 1 1 0 0 | 0 1 2 3 4 50 0 1 1 0 1 1 1 1 0 0 0 1 0 2 0 0 1 1 1 1 |
| 7 | 0 1 2 3 4 50 0 1 1 1 0 1 1 1 0 0 0 1 1 2 1 1 1 0 0 1 | 0 1 2 3 4 50 0 1 0 0 0 1 1 1 0 0 1 1 0 2 0 0 1 1 0 0 | 0 1 2 3 4 50 0 1 1 1 1 1 1 1 0 0 1 1 0 2 0 0 1 1 0 0 |
| 8 | 0 1 2 3 4 50 0 0 0 1 1 1 1 1 0 1 0 1 0 2 0 1 0 0 1 1 | 0 1 2 3 4 50 0 1 0 0 0 1 1 1 0 1 0 1 0 2 0 0 1 1 0 0 | 0 1 2 3 4 50 0 1 1 0 1 1 1 1 0 1 0 1 0 2 1 0 1 1 0 1 |
| 9 | 0 1 2 3 4 50 1 1 1 0 0 1 1 1 0 1 0 1 0 2 0 1 0 1 1 1 | 0 1 2 3 4 50 1 1 0 0 0 1 1 1 0 0 1 1 0 2 1 0 1 1 0 0 | 0 1 2 3 4 50 0 1 1 0 1 1 1 1 0 0 0 1 0 2 1 0 1 1 0 0 |
| 0 | 0 1 2 3 4 50 0 0 0 0 0 1 1 1 0 0 1 1 1 2 0 1 1 1 0 1 | 0 1 2 3 4 50 0 1 0 0 0 1 1 1 0 0 1 1 0 2 0 0 1 1 0 1 | 0 1 2 3 4 50 0 1 1 0 1 1 1 1 0 0 1 1 0 2 0 0 1 1 0 1 |

Литература: 1. Битнер В.И., Мейкшан В.И. Программная организация процесса ввода и предварительной обработки информации в ЭУС. Методические указания. Новосибирск 1988

2. Приложение А

**Задача 2**

Изобразить схему алгоритма поиска пути в КП. Для коммутационного поля, изображенного на рисунке 2, и содержимого МСПЛав, МСПЛвс, МСН на рис. 3 привести пример обработки данных в процессе работы ЭУС, используя исходные данные из таблицы 2.



Рисунок 2 – Трехзвенное КП



Рисунок 3 - Содержимое МСПЛав, МСПЛвс, МСН

Запишите заявки в буфер заявок на формирование ППК (БЗППК) и на подключение КПЗ (БЗКПЗ). Если путь найден, то номера ПЛ и номер выхода записать в регистр вызова (РВ).

Таблица 2

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № вар | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |
| Номера АК | 2,5 | 5,7 | 7,0 | 0,4 | 3,1 | 6, 4 | 8, 3 | 1,5 | 4,3 | 7,2 |
| НомерНаправления  | Н0 | Н1 | Н2 | Н0 | Н1 | Н2 | Н0 | Н1 | Н2 | Н0 |

Пример решения рассмотрен в приведенной для данной задачи литературе.

Литература: 1. Журихина Г.Ф. Ромашова Т.И. Поиск пути в двухзвенном коммутационном поле. Методические указания. Новосибирск 2000г.

2. Приложение Б

**Задача 3**

Изобразить схему алгоритмов формирования и выдачи последовательности периферийных команд. Для коммутационного поля изображенного на рисунке 4 сформировать содержимое РВ для заданных в таблице 3 этапов обслуживания вызовов (ЭОВ).

Определить номера свободных БПК по МСБПК, сформировать последовательность периферийных команд (ППК) на ЭОВ согласно содержимому РВ, записать заявку в ПЗ на выдачу ППК.Скорректировать содержимое БПК и МСПУУ по результату работы программы выдачи ППК.

Таблица 3

|  |  |  |  |
| --- | --- | --- | --- |
| № Вар. | МСБПК | ЭОВ | МСПУУ |
| 1 | 01110000 | 1 | 1110010 |
| 2 | 00111111 | 2 | 1011111 |
| 3 | 01000111 | 3 | 0111100 |
| 4 | 00000011 | 4 | 0101111 |
| 5 | 00010111 | 5 | 1110111 |
| 6 | 00100011 | 6 | 0111011 |
| 7 | 00111100 | 1 | 1001111 |
| 8 | 00001101 | 5 | 1110011 |
| 9 | 01010100 | 3 | 0011111 |
| 0 | 10100111 | 4 | 1111100 |

Пример решения рассмотрен в приведенной для данной задачи литературе.

Литература: 1. Битнер В.И.,Костюкович Н.Ф. Программная организация процессов формирования и выдачи периферийных команд. Новосибирск. СибГУТИ,2000.

2. Приложение В

**Задача 4**

Изобразить схему алгоритмов работы диспетчеров программ приоритетного уровня (ППУ) и программ основного уровня (ПОУ).

Составить таблицу расписания для запуска ППУ согласно данных таблицы 4, определить СВП для ППУ в N-ом первичном периоде, если задано слово активности программ САП.

Таблица 4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № Вар. | П1 | П2 | П3 | П4 | П5 | N | САП |
| 1 | 1пп | 6пп | 3пп | 2пп | 5пп | 5 | 10011 |
| 2 | 3пп | 4пп | 1пп | 7пп | 2пп | 4 | 01111 |
| 3 | 4пп | 5пп | 2пп | 2пп | 3пп | 6 | 11101 |
| 4 | 6пп | 3пп | 3пп | 6пп | 5пп | 3 | 11001 |
| 5 | 2пп | 7пп | 6пп | 3пп | 2пп | 2 | 01011 |
| 6 | 7пп | 1пп | 5пп | 2пп | 5пп | 3 | 01100 |
| 7 | 5пп | 6пп | 3пп | 2пп | 1пп | 4 | 01110 |
| 8 | 1пп | 7пп | 3пп | 2пп | 3пп | 6 | 11111 |
| 9 | 3пп | 4пп | 5пп | 2пп | 1пп | 5 | 11011 |
| 0 | 4пп | 7пп | 2пп | 3пп | 7пп | 7 | 01001 |

Изобразить диаграмму работы программ ППУ и ПОУ в N-ом первичном периоде. Определить адрес прерывания и номер прерванной ПОУ, если задано СВП ПОУ и время работы ДП, ДППУ, ДПОУ составляет 0,2 мс.

Сигнал прерывания поступает через каждые 10 мс. Время выполнения ППУ, время выполнения ПОУ и НА ПОУ заданы в таблице 9.

Таблица 5

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № Вар. |  | П1 | П2 | П3 | П4 | П5 | СВП ПОУ |
|  | Тв ППУ | 1.2 | 2.4 | 1.1 | 2.1 | 1.8 |  |
| 1 | Тв ПОУ | 3.3 | 4.1 | 2.8 | 4.1 | 6.0 | 11010 |
|  | НА ПОУ | 100 | 200 | 300 | 400 | 500 |  |
|  | Тв ППУ | 2.4 | 1.6 | 2.2 | 1.4 | 1.2 |  |
| 2 | Тв ПОУ | 4.0 | 3.1 | 2.6 | 5.0 | 3.2 | 10011 |
|  | НА ПОУ | 110 | 220 | 330 | 440 | 550 |  |
|  | Тв ППУ | 1.6 | 2.1 | 1.1 | 1.0 | 1.5 |  |
| 3 | Тв ПОУ | 4.1 | 3.5 | 2.8 | 4.0 | 5.1 | 11010 |
|  | НА ПОУ | 200 | 350 | 500 | 650 | 800 |  |
|  | Тв ППУ | 1.1 | 2.2 | 1.5 | 1.7 | 2.3 |  |
| 4 | Тв ПОУ | 3.0 | 2.4 | 2.0 | 3.2 | 4.1 | 10101 |
|  | НА ПОУ | 250 | 340 | 460 | 580 | 660 |  |
|  | Тв ППУ | 2.0 | 1.0 | 1.2 | 1.7 | 1.5 |  |
| 5 | Тв ПОУ | 4.2 | 5.0 | 2.4 | 2.8 | 3.3 | 11001 |
|  | НА ПОУ | 190 | 330 | 480 | 580 | 700 |  |
|  | Тв ППУ | 1.1 | 2.2 | 1.0 | 1.9 | 1.8 |  |
| 6 | Тв ПОУ | 3.5 | 4.0 | 3.8 | 4.2 | 5.0 | 11110 |
|  | НА ПОУ | 100 | 220 | 400 | 510 | 700 |  |
|  | Тв ППУ | 2.2 | 1.6 | 2.2 | 1.5 | 1.2 |  |
| 7 | Тв ПОУ | 4.0 | 3.5 | 4.6 | 2.9 | 3.2 | 10011 |
|  | НА ПОУ | 110 | 250 | 370 | 490 | 620 |  |
|  | Тв ППУ | 1.4 | 2.1 | 1.3 | 1.0 | 1.4 |  |
| 8 | Тв ПОУ | 4.3 | 3.5 | 3.8 | 4.0 | 2.1 | 11010 |
|  | НА ПОУ | 200 | 350 | 520 | 690 | 850 |  |
|  | Тв ППУ | 1.1 | 2.0 | 1.5 | 1.8 | 2.1 |  |
| 9 | Тв ПОУ | 3.4 | 2.8 | 2.3 | 3.7 | 4.1 | 10101 |
|  | НА ПОУ | 250 | 360 | 480 | 580 | 680 |  |
|  | Тв ППУ | 2.4 | 1.0 | 1.3 | 1.4 | 1.8 |  |
| 0 | Тв ПОУ | 4.3 | 3.3 | 2.9 | 3.8 | 3.3 | 11001 |
|  | НА ПОУ | 190 | 340 | 480 | 590 | 700 |  |

Литература:

1 Мейкшан В.И., Ромашова Т.И. Организация процесса диспетчирования программ обслуживания вызовов. Методические указания. Новосибирск. СибГУТИ,1994.

2. Приложение Г

**ПРИЛОЖЕНИЕ А**

**Программа сканирования АК**

 При большом количестве КТ они первоначально группируются в линейки, каждая из которых обычно объединяет точки с одинаковым периодом опроса. Затем из отдельных линеек формируется матрица контрольных точек (МКТ), которая вместе с устройством выбора линейки (УВЛ) называется определителем (рисунок1).

В процессе ввода информации выполняются три действия:

1. считывание информации с контрольных точек;
2. предварительная обработка информации с целью выявления наличия изменения в состоянии КТ (предыдущее состояние сравнивается с текущим);
3. запись в буфер заявок на запуск программ обработки информации (БЗО) адреса комплекта, в котором КТ изменила свое состояние.

 Для выполнения процесса сканирования необходимо иметь следующие данные:

1. количество опрашиваемых определителей;
2. число опрашиваемых линеек в каждом определителе;
3. расписание сканирования;
4. таблицы пересчета номера КТ в номер комплекта;
5. количество ячеек в БЗО.

 Процесс ввода информации запускается периодически программой управления (диспетчером) в соответствии с заранее подготовленным расписанием. В общем случае период сканирования определяется двумя обстоятельствами:

1. скоростью старения информации, (наиболее критичным является процесс приема адресной информации);
2. максимально допустимым временем реакции системы (например, при поступлении сигнала занятия с соседней АТС система должна быть готова к приему номера не позже, чем через 100 мс)

 При слишком частом сканировании всех КТ потребовались бы дополнительные ресурсы машинного времени. Поэтому для отдельных АК эта величина составляет до 400 мс.

 В оперативной памяти ЭУС за каждым определителем закрепляется массив состояний КТ (МСКТ), в котором каждой КТ соответствует один разряд (бит) для запоминания ее состояния при предыдущем сканировании. В сканирования эта информация используется при сравнении с текущими результатами с целью обнаружения изменений состояний. Для удобства обработки данной информации структура МСКТ выбирается идентичной структуре МКТ определителя (рисунок 2)



 При организации процесса сканирования необходимо предусмотреть защиту от помех. Для этого используется повторное сканирование. Повторное сканирование выполняется со сдвигом во времени относительно первичного сканирования, и опрашиваются только те линейки, где выявились изменения. Совпадения результатов первичного и повторного сканирования свидетельствует о поступлении действительно входного сигнала. Такое сравнение обеспечивается благодаря хранению предварительных результатов первичного сканирования в так называемом буфере предварительных заявок (БПЗ).

 Т.О. функции ввода информации выполняются программами сканирования двух видов - ПСК1 и ПСК2. Структурная схема взаимодействия этих программ с массивами данных и аппаратными средствами приведена на рисунке 3.

Рассмотрим общий алгоритм работы программ ПСК1, показанный на рисунке 4.



Рисунок 4 – Алгоритм программ ПСК1(а), ПСК2(б)

Получив управление от диспетчера, ПСК1 формирует периферийную команду на опрос КТ i-й линейки (i=1). Из определителя комплектов (ОК) поступает СОС - слово очередного сканирования, которое содержит информацию о текущем состоянии КТ. Предыдущее состояние КТ хранится в i-м слове МСКТ и называется СПС - словом предыдущего сканирования. Обработка результатов первичного сканирования заключается в определении так называемого слова изменений (СИ).

 СИ=СОС1 & СПС (в случае изменения с 0 на 1).

Наличие хотя бы в одном разряде СИ единицы означает, что в соответствующей КТ имеет место изменение состояния. Это обстоятельство фиксируется в БПЗ, куда записывается СИ и номер линейки. После чего перечисленные операции повторяются для следующей по списку линейки, пока не будет опрошено N - линеек.

Программа ПСК2 ведет последовательный опрос лишь тех линеек, номера которых записаны в БПЗ. Аналогично ПСК1 программа ПСК2 формирует СИ2, а затем осуществляет проверку стабильности изменений, формируя ССИ=СИ1&СИ2. Наличие хотя бы в одном разряде ССИ единицы означает, что в соответствующей КТ имеет место изменение состояния. Это обстоятельство фиксируется в БЗО, куда записывается номер комплекта, изменившего свое состояние. После этого программа ПСК2 вносит изменения в МСКТ.

На рисунке 5 приведена временная диаграмма процесса сканирования АК. Здесь показан эффект повторного сканирования, заключающийся в нечувствительности процесса ввода информации к помехам, длительность которых меньше интервала времени между моментами запуска ПСК1 и ПСК2.



Рисунок 5 - Временная диаграмма процесса сканирования АК

**ПРИЛОЖЕНИЕ Б**

**ПОИСК ПУТИ В 3-х ЗВЕННОМ КП**

Задача поиска и занятия соединительного пути (СП) сводится к тому, чтобы для заданного входа найти и отметить занятым СП к одному из свободных выходов в заданном направлении. Решение этой задачи сводится к определению неизвестных координат.

Процедура поиска в коммутационном поле (КП) свободного СП между комплектами разного типа достаточно широко используется многими программами обработки вызовов при необходимости подключения определенных комплектов. Для уменьшения объема этих программ и, соответственно, требуемого объема памяти указанная процедура обычно оформляется в виде специального модуля (подпрограммы), к которой при необходимости может обратиться программа управления подключением комплек­тов путем формирования заявки.

В заявке должны быть указаны следующие данные:

1. адрес точки входа в КП, для которой необходимо выполнить поиск свободного СП;
2. адрес точки выхода из КП (или группы выходов), к которой ведется поиск пути;
3. маршрут поиска.

После завершения процесса поиска формируются указанные выше коорди­наты найденного пути.

Программная организация процесса поиска СП предполагает рациональное формирование данных о структуре КП, группировании выходов в отдельные на­правления связи, состоянии элементов КП и приборов, подключенных к выходам.

Имеются в виду следующие исходные данные:

1. количество и размерность коммутаторов на каждом звене КП;
2. число промежуточных линий (ПЛ) между каждой парой соседних звеньев;
3. количество направлений связи;
4. состояния промежуточных линий ПЛав и ПЛвс.

К оперативным данным о состояниях элементов КП относятся:

1. МСПЛав —- массив состояний промежуточных линий АВ;
2. МСПЛвс — массив состояний промежуточных линий ВС;
3. МСН — массив состояний направлений.

Кроме перечисленных массивов, в процессе управления подключением комплектов участвуют:

1. регистр вызовов (РВ);
2. буфер заявок на обслуживание (БЗО);
3. буфер заявок на формирование последовательности периферийных команд (БЗППК);
4. буфер заявок на подключение комплекта посылки сигнала «Занято» (БЗКПЗ).

В РВ хранятся следующие данные о текущем состоянии вызова:

1. номер этапа обслуживания вызова (ЭОВ);
2. номер входа, на который поступил вызов;
3. номера комплектов, задействованных для обслуживания вызова, и координаты СП к ним.

В БЗО хранятся номера абонентских комплектов (АК), для которых требу­ется подключить определенные комплекты (установить соединения в соответст­вующих направлениях связи). При успешном решении этой задачи номер АК, для которого найден свободный СП, переписывается из БЗО в Б3ППК. В противном случае номер АК, для которого свободный СП отсутствует, записывается в БЗКПЗ.

Функциональная схема поиска пути в 3-х звеном КП приведен на рисунке 1.



Алгоритм программы поиска пути в 3-х звеном КП приведен на рисунке 2.

Описание алгоритма программы поиска свободного СП

1. Чтение номера АК и номера направления из БЗО.
2. Чтение из массива МСПЛав слова состояния (Saв) из строки номер которой соответствует номеру коммутатора, к которому подключен заданный АК.



1. Чтение из массива МСН слова состояния выходов (Sн) из строки, номер которой равен номеру заданного направления.
2. Если в направлении отсутствуют не опробованные выходы, находящиеся в свободном состоянии (Sн-0), то перейти к п.13.

5. Поиск в Sн самой левой единицы (ПСЛЕ). Номер найденного разряда определяет номер коммутатора на звене С, через который можно установить соединение.

1. Чтение из массива МСПЛвс слова состояния Sвс из строки с номером, равным номеру коммутатора на звене С, через который можно установить со­единение.
2. Определить слово соединительного пути (ССП) путем логического умножения Saв и Sвс :ССП=Saв & Sвс
3. Если ССП=0, то записать нуль в разряд слова Sн и перейти к п. 4.
4. Осуществить операцию ПОЛЕ для ССП. Номер найденного разряда (самая левая единица) дает номер коммутатора на звене В, через который можно уста­новить соединение.

10. Записать в РВ для данного АК следующую информацию:

1. номер входа в КС;
2. номер ПЛaв;
3. номер ПЛвс;
4. номер выхода из КС.

11 Для отметки выбранного пути занятым заменить 1 на 0:

1. в массиве МСПЛaв — разряд в слове;
2. в массиве МСПЛвс — разряд в слове;
3. в массиве МСН — разряд в слове .
4. Записать заявку в БЗППК и перейти к п. 14.
5. Записать заявку в БЗКПЗ.
6. Если есть другие заявки в БЗО, то перейти к п. 1 .
7. Конец выполнения программы.

Пример поиска свободного СП. Пусть требуется найти свободный СП ме­жду 8-м входом и любым свободным выходом в направлении Н1. Решение этой задачи для КП, структура которого приведена на рисунке 3, осуществляется в со­ответствии с описанным алгоритмом. Содержимое массивов МСН, МСПЛав и МСПЛвс показано на рисунке 4.



1. Из массива МСПЛав читать содержимое 2-го слова, в котором отобража­ются состояния ПЛав, доступных АК8: Saв= 101.
2. Из массива МСН считать 1-е слово, в котором хранится информация о со стоянии выходов в направлении Н1-- Sн 011.
3. Номер 1-го разряда для самой левой единицы в Sн указывает на коммутатор С1 в котором имеется свободный выход в требуемом направлении связи.
4. Из массива МСПЛВС читать 1-е слово, в котором записаны состоянии
входов коммутатора С1 — Sвс= 010.
5. Наличие свободных СП определяем путем операции логического умно­жения над словами Saв и Sвс: ССП=101 & 010=000. Это говорит об отсутствии
свободных СП для установления соединения между АК8 и выбранным выходом в коммутаторе С1
6. В слове Sн самую левую единицу заменяем нулем и получаем Sн=001.
Наличие единицы в разряде с номером 2 говорит о том, что в направлении Н1, есть еще один свободный выход, относящийся к коммутатору С2.
7. Из массива МСПЛвс считать слово состояний входов коммутатора С2 Sвс=011.
8. Определяем ССП=Sас & Sвc =101 & 011 = 001. Поскольку ССП#0. самый левый разряд с номером 3 определяет номер коммутатора В3, через который
можно установить соединение от АК8 в требуемом направлении связи.
9. В регистр вызова (РВ) для АК8 записать координаты найденного пути:
10. номер входа в КС — 8 (номер АК);
11. номер свободной ПЛав — 8;
12. номер свободной ПЛвс — 8;
13. номер свободного выхода из КС — 7.

7. Отметить занятыми выбранные ПЛ, а также выбранный свободный выход в направлении Н1. С этой целью записываем 0:

1. в 3-й разряд 3-го слова массива МСПЛав;
2. в 3-й разряд 2-го слова массива МСПЛвс;
3. во 2-й разряд 1-го слова массива МСН.

8. Записать число 9 (номер АК) в БЗ ППК,

**ПРИЛОЖЕНИЕ В**

**Программы формирования и выдачи ППК**

Заявки для програм­мы формирования последовательности периферийных команд ППК подготавливает программа управления подключением комплектов. Эти заявки записываются в буфер (БЗ на формирование и выдачу ППК). Каждая заявка в этом буфере имеет свой номер и зак­реплена за определенным АК. Информационные связи между программами управления подключением комплектов, формирования ППК (ПрФ ПК*.)* и выдачи ПK (Пр.В ПК) показаны на рисунке1. Как видно на этом рисунке, взаимодействие между программами происходит с помощью буферов и полей зая­вок. Заявки в буфере представлены порядковым номером и номером найденного перед записью заявки комплекта, для подключения (или отключения) которого к АК необходимо сформировать ППК. Буферы образуют для тех программ, поток заявок к которым невелик.

Одна заявка в поле заявок занимает один разряд слова данных. По номеру слова и номеру разряда, где найдена единица, программа выдачи ПК выполняет пересчет этой заявки в номер буфера ПК (БПК), из которого должны быть выданы ПК.

Функции программы формирования ППК таковы:

1) поиск свободного буфера ПК (БПК),

1. чтение данных о координатах пути AK - комплект из PB,
2. формирование периферийных команд и запись их в найденный свободный БПК,
3. запись заявки в поле заявок на выдачу ПК*.*

Диспетчер данного приоритетного уровня системы коммутационных программ, которому подчинена программа формирования ППК, анализирует содержимое БЗ на формирование и выдачу ППК. Происходит  выбор очередной заявки и передача ее программе формирования ППК

Каждая заявка в БЗ закреплена за своим АК. Программе формирования ППК доступны все необходимые данные об обслуживаемом вызове, хранящиеся в регистре вызова (РВ), закрепленном за каждым АК.

Поиск свободного БПК выполняется по содержимому массива состояний БПК. В этом массиве за определенным разрядом данного слова закреплен конкретный БПК. Пусть логическая единица в некотором разряде этого массива соответствует свободности определенного БПК.

Программа отыскивает единицу, определяет номер свободного БПК, отмечает найденный буфер занятым в массиве состояний БПК. В найденный БПК записываются формируемые по содержимому РВ периферийные команды.

Если нет ни одного свободного БПК, то программа формирования ППК возвращает заявку, полученную от диспетчера, в БЗ на формирование и выдачу ППК Приналичии свободного БПК, заявка в БЗ на формирование и выдачу ППК аннулируется, чтобы одна и та же заявка не обслуживалась многократно.



Каждая ПК характеризуется форматом и содержимым, В формате ПК имеется три зоны: признак выдачи (ПВ), код операции (КО), адресная информация. ПВ занимает один разряд (старший) в слове ПК.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ПВ | КО | АПУУ | Y | X |

Значение ПВ=1 указывает на необходимость выдачи ПКв управ­ляемый объект (в частности - в ПУУ квазиэлектронной АТС).

Код операции выражается одним разрядом слова ПК. Значение КО=1 указывает на необходимость включения коммутационного элемента КП, КО=0 - выключения.

Адресная информация состоит из кода адреса блока ПУУ, адреса вертикали Y и адреса горизонтали коммутатора КП.



Программа формирования ПК формирует и записывает в БПК определенное количество ПК, зависящее от этапа обслуживаемого вызова.

После записи всех требуемых для обслуживания данного вызова ПК в БПК программа формирования ПК записывает заявку на вы­дачу этих команд в поде заявок. Место заявки в этом поле определяет­ся номером БПК, где записаны сформированные для обслуживания дан­ного вызова ПК.

Заявки из поля заявок на выдачу ПК будут прочитаны программой выдачи ПК.

Алгоритм работы программы формирования ППК представлен на рисунке 2.

Функции программ выдачи ПК:

1) поиск заявок в поле заявок на выдачу ПК;

2) определение номера БПК, в котором имеются не выданные в ПУУ ПК;

3) отыскание в БПК команды с признаком ПВ=1;

1. считывание из этой ПКадреса блока ПУУ и определение его состояния по содержимому массива состояния ПУУ;
2. выдача ПК и коррекция признака ПВ (ПВ=0)

6) отметка в массиве состояния ПУУ занятости активного блока ПУУ, выполняющего данную ПК;

7)запись заявки на контроль выполнения ПК в поле заявок.

 Алгоритм работы программы выдачи ПК показан на рисунке 3.

Программа выдачи ПК,обнаружив заявку в каком-либо разряде поля заявок, обращается по номеру этого разряда к соответствующему БПК и последовательно выдает в ПУУ записанные там ПК. Если требуемое для выполнения ПК устройство занято, то эта команда не выдается и программа пытается выдать следующую в последовательности ПК. В том случае, когда ПУУ свободно и может принять ПК, команда выдается с одновременной отметкой занятия соответствующего ПУУ в массиве состояния ПУУ, отметкой выдачи этой команды в БПК и записью заявки на контроль выполнения ПК в поле заявок. Если все команды из ППК выданы в ПУУ, то заявка на выдачу ПК для данного БПК исключается из соответствующего поля заявок. Если какую-либо команду ППК из обрабатываемого буфера не удается выдать из-за занятости требуемого блока ПУУ, то она пропускается и выполняется попытка выдать другие команды.Оставшиеся не выданными из данного буфера ПК будут обслуживаться программой выдачи ПКпри последующем ее запуске диспетчером. Буфер ПК не освобождается до тех пор, пока не будут выданы все записанные в нем ПК. Программа выдачи ПК относится к периодическим программам.



 Для получения достоверных данных об окончании выполнения этапа обслуживания вызова необходимо контролировать правильность: исполнения ПК блоком ПУУ, комплектом или элементом КП. Для этогоиспользуется программа контроля выполнения ПК.

Функции этой программы таковы:

1) чтение заявки из поля заявок и определение номера БПК, в котором хранится (хранятся) команды с отметкой ПB=0 (выданы ПУУ),

2) запрос данных из определителя ПУУ, за которым закреплен подлежащий контролю блок ПУУ (каждый блок ПУУ имеет контрольные точки, по состоянию которых определяется правильность расшифровки и выполнения ПК),

3) отметка свободности блока ПУУ в массиве состояния ПУУпри правильном исполнении ПК,

1. исключение заявки из поля заявок на контроль, если осмотрены все ПКданного буфера ПК с признаком ПВ=0,
2. освобождение БПКи отметка его свободности в массиве состояний БПК, если все его ПКвыданы и исполнены верно,
3. формирование заявки в БЗ на окончание подключения КПН, если все ПКданного буфера выполнены верно,

7)формирование заявки к программе технического обслуживания (БЗ к СПТО), если какая-либо ПК не выполнена, с указанием но­мера БПК и номера ПК.

Алгоритм программы показан на рисунке 4.



Программа контроля выполнения ПКзапускается со сдвигом по отношению к программе выдачи ПК на величину времени выполнения ПУУ выданной в него ПК. Контроль выполнения ПК основан на анализе состояний КТ ПУУ, связанных со схемами контроля ПУУ, которые проверяют правильность приема ПК, ее дешифрации и формирования необходимых выходных сигналов.

При наличии заявки от программы выдачи ПК программа контроля выполнения ПК обращается к соответствующему БПК и поочередно проверяет состояние ПК, записанных в этом буфере.
Если проверяемая ПК имеет отметку о выдаче, то программа производит опрос КТ того ПУУ, в который была выдана эта
команда. При нормальном выполнении выданной команды соответствующее ПУУ отмечается свободным в массиве состояний
ПУУ и программа переходит к контролю выполнения следующей
по порядку ПК. Если выданная ПК не выполнена ПУУ, то программа записывает заявку к определенной программе технического обслуживания, сообщая ей номер БПК и порядковый номер невыполненной ПК, после чего переходит к контролю выполнения следующей ПК в БПК.

После проверки всех ПК, содержащихся в БПК, программа
исключает заявку на контроль выполнения ПК для этого буфера
из поля заявок. При этом, если все ПК из БПК выполнены, то
программа освобождает БПК, делая об этом отметку в массиве состояний БПК.

Рассмотрим пример формирования и выдачи ПК*.* Пусть известна структура и параметры КП(смотри рисунок 5). В выхо­ды КП включены следующие комплекты:

 КПН1, КПН2, КСГ1, КСГ2, КПВ1, ККПВ1, ШК1.



На первом этапе обслуживания вызова (ЭОВ1) необходимо подключить КПН (комплект приема номера) и КСГ (комплект сигнала "Готово" или "Ответ станции"), для этого необходимо сформировать четыре ПК (по две для каждого комплекта т.к. КП двухзвенное).

На ЭОВ2 необходимо отключить КСГ т.к. уже начат набор номера. Для этого формируются две ПК.

На ЭОВ3 необходимо отключить КПН, а также подключить КПВ (комплект посылки вызова) и ККПВ (комплект контроля посылки вызова). Для этого формируются шесть ПК.

На ЭОВ4 необходимо отключить КПВ и ККПВ, а также подключить ШК который подключается и к абоненту А и к абоненту Б. Для этого необходимо сформировать восемь ПК.

На ЭОВ5 необходимо отключить ШК и подключить КПЗ (комплект посылки “Занято”). Для формируются шесть ПК.

На ЭОВ6 необходимо отключить КПЗ. Для этого формируются две ПК.

 Пусть в БЗ на формирование и выдачу ППК записано три заявки. В соответствии с первой заявкой необходимо подключить АК2 к КПН1 и КСГ1 (ЭОВ1). Вторая заявка требует обслуживания АК7 на втором ЭОВ2. Третья заяв­ка требует обслуживания АК5на 4-ом ЭОВ.

В РВ1, РВ2 и РВ3 записаны следующие данные:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| РВ1 |  | РВ2 |  | РВ3 |
| АК(А)2 |  | АК(А)7 |  | АК(А)6 |
| КПН1 |  | КПН2 |  | АК(В)14 |
| ПЛ1 |  | ПЛ5 |  | КПВ1 |
| КСГ1 |  | КСГ2 |  | ПЛ12 |
| ПЛ2 |  | ПЛ6 |  | ККПВ1 |
| ЭОВ1 |  | ЭОВ2 |  | ПЛ8 |
|  |  |  |  | ШК1-А |
|  |  |  |  | ПЛ7 |
|  |  |  |  | ШК1-Б |
|  |  |  |  | ПЛ11 |
|  |  |  |  | ЭОВ4 |

Эти данные используются программой формирования ПК для определения значений всех зон формата ПК. В БЗ на формирования и выдачу ППК записано 3 заявки

|  |
| --- |
| БЗ на формирование и выдачу ПК |
| Заявка №1 | Заявка №2 | Заявка №3 |
| АК2 | АК7 | АК6 |

Начиная обслуживать заявку №1, программа формирования ППК находит свободный буфер БПК. Пусть им будет БПК3.

В БПК3 будут записаны следующие четыре ПК.

БПК3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ПВ | КО | АПУУ | Y | X |  |
| 1 | 1 | 1 | 2 | 1 |  |
| 1 | 1 | 4 | 2 | 1 | Подключение КПН1 |
| 1 | 1 | 1 | 2 | 2 |  |
| 1 | 1 | 5 | 2 | 1 | Подключение КСГ1 |

Признак выдачи везде равен 1,т.к. ПК еще не выдавалась. КО везде равен 1,т.к. необходимо на ЭОВ1 подключить КПН и КСГ к АК с помощью КЭ КП. Значения зоны адреса объекта определяются согласно структуры КП (Рисунок 2) и содержимого РВ.

Пусть для обслуживания второй заявки найден свободным БПК 5.

БПК5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ПВ | КО | АПУУ | Y | X |  |
| 1 | 0 | 2 | 2 | 2 |  |
| 1 | 0 | 5 | 3 | 2 | Отключение КСГ2 |

В БПК 5 записаны ПК с КО=0,т.к. на ЭОВ2 необходимо отключить КСГ после начала набора абонентом А номера абонента Б.

Пусть для обслуживания третьей заявки найден свободным БПК8.

БПК8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ПВ | КО | АПУУ | Y | X |  |
| 1 | 0 | 2 | 1 | 4 |  |
| 1 | 0 | 7 | 2 | 2 | Отключение ККПВ1 |
| 1 | 0 | 3 | 4 | 4 |  |
| 1 | 0 | 7 | 3 | 4 | Отключение КПВ1 |
| 1 | 1 | 2 | 1 | 3 |  |
| 1 | 1 | 6 | 1 | 2 | Подключение ШК-А |
| 1 | 1 | 3 | 4 | 3 |  |
| 1 | 1 | 6 | 2 | 3 | Подключение ШК-Б |

На ЭОВ4 необходимо сформировать 4ПКдля управления отклю­чением ККПВ и КПВ, которые были подключены к АК-А и АК- Б на ЭОВЗ и 4ПК для управления подключением ШК.

Программа формирования ПКзаписывает заявки в разряды поля заявок на выдачу, закрепленные за 3,5,8 БПК.

Программа - диспетчер периодически запускает программу выдачи ПК, которая отыскивает заявку, определяет соответствующий ей номер БПК и управляет выдачей ПК. Не все ПК данного БПК могу быть выданы при первой попытке. Например, из БПК3 нельзя выдать ПК3 т.к. ПК1 занимает ПУУ1. Программа выдачи будет запускаться до тех пор, пока не будут выданы все ПК.

**ПРИЛОЖЕНИЕ Г**

**ПРИНЦИПЫ И ОРГАНИЗАЦИЯ** **ДИСПЕТЧИРОВАНИЯ**

 Существует два основных принципа очередности запуска программ:

* принцип запуска программ по заявкам
* принцип приоритетности заявок и программ.

*Принцип запуска программ по заявкам* означает, что программа запускается только при наличии заявки на ее выполнение.

В Управляющей системе (УС) могут одновременно иметься несколько заявок на выполнение различных программ.

*Принцип приоритетности заявок* устанавливает приоритеты между заявками в зависимости от требуемой срочности их выполнения.

Назовем набор правил, устанавливающих соответствие между приоритетами заявок и очередностью запуска соответствующих программ, *дисциплиной обслуживания заявок,* или *алгоритмом диспетчирования программ.*

Применяются дисциплины обслуживания заявок:

* с абсолютными,
* относительными
* смешанными приоритетами.

При *абсолютным*  *приоритете* заявка с более высоким приоритетом может прервать выполняющуюся в данный момент программу с менее высоким приоритетом. После выполнения программы с более высоким приоритетом продолжается выполнение прерваннойпрограммы (рисунок 1.а).

При *относительном приоритете*, если выполняется программа с низким приоритетом, то если поступают заявки прерывания не происходит, а после ее выполнения запускается программа с более высоким приоритетом, даже если заявка от нее поступила позже (рисунок 1.б).

При *смешанном приоритете* все программы разбиваются на группы. Между группами организуется абсолютный приоритет, а внутри группы относительный приоритет.



Рисунок 1 – Временная диаграмма обслуживания заявок

Для простоты будем считать, что все программы обслуживания вызовов следует разделить на две группы:

1. программы высокой срочности. Это программы приоритетного уровня (ППУ).;
2. программы нормальной срочности. Это программы основного уровня (ПОУ).

Чтобы программы выполнялись со строгой периодичностью, время работы управляющей системы (УС) разбивают на фиксированные интервалы времени, которые называются первичными периодами (τ1). Например:

* Программа 1 - П1 - 10 мс
* Программа 2 – П2 К - 20 мс
* Программа 3 – П3 - 100мс и т.д.

Величина первичного периода выбирается из минимального периода запуска одной из программ (τ1=10 мс)

После этого составляется таблица расписаний, где число столбцов равно числу программ, а число строк числу первичных периодов тогда:

* Программа 1 - П1 – будет запускаться каждый первичный период
* Программа 2 – П2 К - будет запускаться каждый второй первичный период
* Программа 3 – П3 - будет запускаться каждый десятый первичный период

Число строк определяется как наименьшее общее кратное (НОК) множества запускаемых программ (в данном случае 10)

Таблица расписаний:

 1 2 3 … n - номер программ

|  |  |  |  |
| --- | --- | --- | --- |
| 1  | 1 | 1 | 0 |
| 2 | 1 | 0 | 0 |
| 3 | 1 | 1 | 0 |
| 4 | 1 | 0 | 0 |
|  | 1. | ... | 0 |
| 10 | 1 | 0 | 1 |

 номер ПП

 Будем считать, что диспетчеры включают в себя:

1. диспетчер ППУ (программ приоритетного уровня)
2. диспетчер ПОУ (программ основного уровня)
3. диспетчер прерывания (ДП), который координирует работу ДППУ и ДПОУ.

 В основу работы каждого диспетчера (ДППУ или ДПОУ) положено соответствующее расписание. Функциональная схема показана на рисунке 2 и включает в себя следующие элементы: счетчик первичных периодов (СПП), таблицу расписаний (ТР), слово активности программ (САП), слово выполняемых программ (СВП), таблицу начальных адресов программ (ТНАП), и модули программ П1,П2, ...,П.

Номер текущего первичного периода задается счетчиком первичных периодов (СПП), который наращивается на единицу при поступлении сигнала прерывания. После прохождения всех первичных периодов текущего цикла осуществляется переход снова к первому.

Элементами ТР являются двоичные переменные, принимающие значения 0 или 1. Наличие "1" означает необходимость запуска программы в данном первичном периоде "0"-пропуск этой программы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | J | СПП | +1 при поступлении сигнала  |
|  |  | Счетчик первичных периодов |  |  | прерывания |
|  |  |  | 1 | 2 | 3 | 4 |  | n |  |  |  |
|  |  | 1 | 1 | 1 | 0 | 1 | ... | 1 |  | ТР |  |
|  |  | 2 | 1 | 0 | 0 | 0 | ... | 0 | Таблица расписаний |
|  |  | 3 | 1 | 0 |  |  |  |  |  |  |  |
|  |  | … | 1 | … | 0 | 1 | ... | 0 | ТРj | Строка ТР |
|  |  | 10 | 1 | 0 | 1 | 0 | ... | 0 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | САП | 1 | 1 | 0 | 1 | ... | 0 |  | & |
| Слово активности программ |  |  |  |  |  |  |  |  |
|  |  | СВП | 1 | 0 | 0 | 1 | ... | 0 |  |  |  |
| Слово выполнения программ |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | просмотр  | СВП |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 1 | НА П1 (начальные адреса) |  |  |  |
|  |  | 2 | НА П2 (начальные адреса) |  |  | ТНАП |
|  |  | 3 |  |  |  | Таблица начальных |
|  |  | 4 | НА П4 (начальные адреса) |  |  | адресов программ  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | П1 | ... | Пi | ... | П4 |  |  |

Рисунок 2 - Функциональная схема работы диспетчера

 Для блокировки выполнения некоторых второстепенных программ, при перегрузке управляющей системы (УС) служит слово активности программ САП. Конкретный набор программ запускаемых в данный первичный период определяется словом выполнения программ (СВП), как: <СВП>=<САП>&<ТРi>.

 Начальные адреса всех программ сведены в таблицу начальных адресов программ ТНАП, где каждая строка соответствует своей программе. Последовательность действий выполняемых при поступлении сигнала прерывания, когда управление получает ДППУ показаны на следующем рисунке 3.а.

 В отличие от ППУ, запуск программ ПОУ осуществляется в случайные моменты времени. По отношению к действиям ДППУ в работе ПОУ можно выделить следующие отличия:

1. Запуск любой программы ПОУ осуществляется при условии, что имеется заявка на выполнение программы. : <СВП>=<ПЗ>&<ТРi> (где ПЗ - поле заявок).
2. Переход с одной строки на другую в ТР осуществляется по мере выполнения всех программ текущего первичного периода (СПП=СПП+1)
3. при поступлении сигнала прерывания программа основного уровня ПОУ прерывается. Чтобы в дальнейшем обеспечить возможность продолжить выполнение программы, информацию о состоянии программы на момент прерывания, сохраняют в специальной зоне памяти. (АПР).

Алгоритм работы программ - диспетчеров основного уровня ДПОУ показан на рисунке 3. б.



Рисунок 3 – Алгоритм работы ППУ и ПОУ

Реализация дисциплины обслуживания заявок или алгоритма диспетчирования со смешанными приоритетами осуществляется с помощью специальных аппаратурных и программных средств, образующих иерархическую структуру организации запуска программ рисунке 4.



Рисунок 4. Структура операционной системы

На верхнем уровне управления находятся блок прерывания программ и диспетчер прерываний. Блок прерывания программ и диспетчер прерываний осуществляют анализ приоритетов поступающих в виде сигналов прерывания заявок и запуск управляющей программы - диспетчера приоритетного уровня, соответствующего наиболее приоритетной заявке.

Необходимость в специальной программе-диспетчере для приоритетного уровня вызвана тем, что принятый сигнал прерывания может сигнализировать о поступлении на данный приоритетный уровень нескольких различных заявок или по принятому сигналу прерывания может быть выполнен запуск нескольких различных программ.

Выделяются следующие приоритетные уровни и связанные с ними программы-диспетчеры:

* приоритетный уровеньдля системы программ технического обслуживания СПТО;
* приоритетный уровень для программ обслуживания ввода-вывода (ПОВ-В),;
* приоритетный уровень Н для периодических программ, запускаемых по сигналу прерывания от датчика времени и требующих строгого соблюдения периодичности запуска;
* приоритетный уровень L для периодических программ, запускаемых по сигналу прерывания от датчика времени и допускающих некоторые отклонения от заданной периодичности запуска;
* приоритетные уровень В, называемый также основным уровнем, для программ, которые поэтому имеют самый низкий абсолютный приоритет.

Приоритетные уровни расположены в порядке убывания их приоритетов.



Рисунок 5 – Диаграмма разделения времени работы программ ППУ и ПОУ

**Пример решения задачи 4**.

Изобразить схему алгоритмов работы диспетчеров программ приоритетного уровня (ППУ) и программ основного уровня (ПОУ).

Составить таблицу расписания для запуска ППУ согласно данных таблицы 4, определить СВП для ППУ в N-ом первичном периоде, если задано слово активности программ САП.

Таблица 4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № Вар. | П1 | П2 | П3 | П4 | П5 | N | САП |
| х | 4пп | 6пп | 3пп | 2пп | 1пп | 2 | 01101 |

Изобразить диаграмму работы программ ППУ и ПОУ в N-ом первичном периоде. Определить адрес прерывания и номер прерванной ПОУ, если задано СВП ПОУ и время работы ДП, ДППУ, ДПОУ составляет 0,2 мс.

Сигнал прерывания поступает через каждые 10 мс. Время выполнения ППУ, время выполнения ПОУ и НА ПОУ заданы в таблице 5.

Таблица 5

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № Вар. |  | П1 | П2 | П3 | П4 | П5 | СВП ПОУ |
|  | Тв ППУ | 2.2 | 1.4 | 1.7 | 2.3 | 1.2 |  |
| х | Тв ПОУ | 3.7 | 4.4 | 2.2 | 4.3 | 5.7 | 11010 |
|  | НА ПОУ | 100 | 200 | 300 | 400 | 500 |  |

**Решение задачи №4**

Все программы обслуживания вызовов делятся на две группы:

1. Программы высокой срочности. Это программы приоритетного уровня (ППУ).
2. Программы нормальной срочности - программы основного уровня (ПОУ). На их выполнение остается время, свободное от выполнения программ приоритетного уровня.

Необходимо распределить моменты запуска приоритетных программ (ППУ) по различным первичным периодам. Данная задача решается путем составления таблиц расписаний (ТР) запуска программ.

 1 2 i n - номер программ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1  | 0 | 1 | ... | 1 | ... | 0 |
| 2 | 1 | 0 | ... | 0 | ... | 0 |
|  | ... | ... | ... | ... | ... | ... |
| j | 1 | 1 | ... | 1 | ... | 0 |
|  | ... | ... | ... | ... | ... | ... |
| m | 1 | 0 | ... | 0 | ... | 1 |

 номер ПП

 Число столбцов ТР соответствует числу программ (в данном примере их пять). Число строк - числу первичных периодов, количество которых определяется как наименьшее общее кратное (НОК) множества запускаемых программ (для данного примера наименьшее общее кратное (НОК) между 4, 6, 3, 2 и 1 равно – 12, т.е. это минимальное число которое делится без остатка на 4, 6, 3, 2, 1). Элементами ТР являются двоичные переменные, принимающие значения 0 или 1. Наличие "1" означает необходимость запуска программы в данном первичном периоде, "0"-пропуск этой программы. Частота расположения единиц в столбцах ТР определяется периодом запуска соответствующей программы (каждый первичный период – 1пп, каждый второй первичный период – 2пп, каждый третий первичный период – 3пп и т.д.) смотри таблицу 4.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | П1 | П2 | П3 | П4 | П5 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 | 0 | 1 |
| 5 | 0 | 0 | 1 | 1 | 1 |
| 6 | 0 | 0 | 0 | 0 | 1 |
| 7 | 0 | 1 | 0 | 1 | 1 |
| 8 | 1 | 0 | 1 | 0 | 1 |
| 9 | 0 | 0 | 0 | 1 | 1 |
| 10 | 0 | 0 | 0 | 0 | 1 |
| 11 | 0 | 0 | 1 | 1 | 1 |
| 12 | 1 | 0 | 0 | 0 | 1 |

 Согласно алгоритма работы программ приоритетного уровня (рисунок 3 а):

1. счетчик первичных периодов (СПП) увеличивается на «1» и становится равным N, значение которого берется из таблицы 8 (N=2);
2. формируется слово выполнения программ (СВП) для второго первичного периода (N=2) - <СВП>=<САП>&<ТР2>=01101 & 00101 = 00101,

где САП - слово активности программ (из таблицы 8), а ТР2 – это содержимое строки таблицы расписаний для второго первичного периода.

1. Из СВП определяются номера программ, которые должны запускаться во втором первичном периоде (в данном примере это программы П3 и П5), каждая из этих программ выполняется в течении времени Тв (смотри таблицу 5). Например, для выполнения программы П3 требуется 1,7 мс, а для П5-1,2 мс. Остальное время отводится для программ ПОУ.
2. Требуется изобразить диаграмму работы программ ППУ и ПОУ в N-ом первичном периоде. Определить адрес прерывания и номер прерванной ПОУ, если задано СВП ПОУ (11010 – из таблицы 5) и время работы всех программ ПОУ (смотри таблицу 5), а также время работы диспетчеров ДП, ДППУ, ДПОУ которое составляет 0,2 мс.



 Из диаграммы видно, что при поступлении сигнала прерывания ДП передает управление ДППУ (время работы диспетчеров по 0,2 мс), который запускает программы ППУ П3 (Тв=1,7 мс) и П5 (Тв=1,2мс) согласно СВП, относящееся ко второму первичному периоду. После окончания выполнения этих программ ДППУ обращается к ДП, который отдает управление ДПОУ (время работы диспетчеров по 0,2 мс), до поступления следующего сигнала прерывания выполняется программа ПОУ П1 (Тв=3,7мс) и начинает выполняться программа П2 (Тв =4,4) (Слово выполнения программ ПОУ задано в таблице 5). В 10мс поступает сигнал прерывания и программа основного уровня П2 будет прервана. Необходимо определить адрес прерывания.

Начальные адреса всех программ ПОУ сведены в таблицу 5, из которой видно, что каждая программа занимает по 100 ячеек памяти ЗУ. Вся программа состоит из микрокоманд. В каждой ячейке хранится одна микрокоманда (МК). Необходимо определить:

1. Сколько микрокоманд программы П2 успеет выполниться в данном первичном периоде до поступления сигнала прерывания.

На выполнение программы П2 всех 100 микрокоманд необходимо время 4,4 мс, а в данном первичном периоде программа П2 будет выполняться только 2,2 мс. Можно определить сколько микрокоманд программы П2 успеет выполниться по формуле:

 Nмк=2,2\*100/4,4=50

1. Определим адрес прерывания, зная что программа П2 начинается с начального адреса (НА) 200: Апр=НА+Тмк= 200+50=250