

Лабораторная работа № 5

Изучение моделей взаимодействия распределено выполняющихся процессов

1. Цель работы: изучить алгоритмическое построение взаимодействия распределено выполняющихся процессов и средств их реализации с использованием подпрограмм библиотеки PVM.
2. Теоретическое введение.

Наряду с уже рассмотренными моделями взаимодействия распределенных процессов (взаимодействие «производитель-потребитель», «взаимодействие равных», «клиент-сервер») существует целый ряд важных моделей, к которым могут быть отнесены следующие:

- 1) модель «зонд-эхо»;
- 2) модель «распределенные семафоры»;
- 3) модель передачи маркера.

Топология вычислительного кластера (определяющая взаимодействие вычислительных процессов), построенного для распределенных вычислений, может быть представлена в виде графа с узлами – процессами, и ребрами – каналами передачи данных. «Зонд» - это сообщение, передаваемое узлами графа своему преемнику, «эхо» - это последующий ответ преемника родительскому узлу. Модель «зонд-эхо» используется как для построения топологии сети, так и для широковещательной рассылки сообщений по всем узлам (процессам) сети (процесс, выполняющийся на узле S должен разослать сообщения процессам, выполняющимся на всех остальных узлах сети).

Основным требованием при широковещательной рассылке является необходимость того, чтобы сообщение попадало на каждый узел (в каждый процесс) только один раз. Для обеспечения этого требования узел, выполняющий рассылку, должен иметь два специальных средства, обеспечивающих эту рассылку – это информация о топологии сети (взаимодействии распределено выполняющихся процессов) и остовное дерево сети. Топология может быть представлена в виде симметричной матрицы, где элемент $[i, j]$ равен 1, если узлы i и j связаны между собой каналами передачи данных, 0 – в противном случае.

Топология строится на основе выполнения широковещательных рассылок по сети и получения эхо-ответов (построение топологии рассматривается далее). Для эффективной рассылки узел должен построить и использовать остовное дерево сети. Корнем остовного дерева является узел, инициирующий рассылку. Остовное дерево отличается от топологии сети тем, что в нем исключены ребра топологии, дублирующие связи между узлами.

Отличие остовного дерева сети от ее топологии комментирует рис. 2.1. Остовное дерево интерпретируется соответствующей матрицей, построенной на основе матрицы топологии, из которой исключены единичные элементы, дублирующие связи между узлами. Через t обозначим остовное дерево сети.



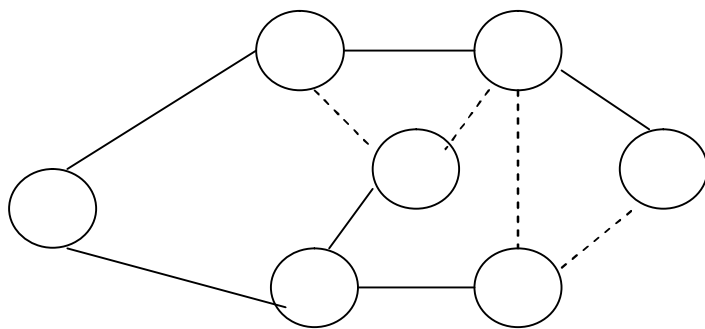


Рис. 2.1 Остовное дерево сети

- ребра основного дерева
- остальные ребра графа

Процесс рассылки сообщения инициируется корневым узлом остовного дерева. При этом соседям рассылается как само сообщение, так и остовное дерево. Рассылка остовного дерева вместе с сообщением необходима для того, чтобы другие узлы топологии знали, куда им необходимо далее пересылать сообщение. Т.о. формат передачи сообщения между узлами имеет следующий вид:

сообщение	Массив остовного дерева
-----------	-------------------------

Пересылка сообщения между узлами остовного дерева осуществляется с использованием уже существующих каналов передачи данных, созданных предварительно для обмена сообщениями между процессами. Построению остовного дерева предшествует процедура построения топологии сети, которая заключается в следующем.

Каждому узлу известна лишь его локальная топология (т.е. связи с его соседями). Задача состоит в том, чтобы собрать воедино все локальные топологии и построить общую топологию сети. Особенности алгоритма построения топологии сети состоят в следующем:

- 1) каждый узел (начиная с инициатора рассылки) посылает зонд (запрос на топологию) своим соседям. В зонде указывается источник этого зонда. Таким образом, форма зонда следующий:

Тип сообщения (зонд)	Идентификатор источника
----------------------	-------------------------

- 2) узлы преемники, получив зонд, ретранслируют его далее, но не посылают эхо-ответ до тех пор, пока не придет эхо-ответ от их преемников (при ретрансляции указывается уже другой источник зонда);
- 3) так как источник зонда (начальный или последующий) знает сколько у него каналов рассылки зондов, из них он и будет ожидать возврат эхо-ответов;
- 4) процесс рассылки зонда продолжается до тех пор, пока он не достигнет узлов, не имеющих соседей, либо, если топология циклическая, на узел не придет пустой эхо-ответ, что определяет отсутствие необходимости дальнейшей ретрансляции зонда;

- 5) в топологии с циклами если узел получает последующие зонды во время ожидания эхо-ответа, он сразу отправляет пустой эхо-ответ на тот родительский узел, который послал этот зонд;
- 6) получив эхо от сыновних (соседних) узлов, узел объединяет эти ответы со своим множеством соседей и отправляет это сообщение родительскому узлу. Таким образом, суммарное сообщение должно содержать некоторую часть топологии сети, идентифицируемую массивом топологии. Формат эхо-ответов следующий:

Тип сообщения (эхо)	Источник ответа	Массив топологии
---------------------	-----------------	------------------

Т.е. массив топологии (используется его часть) заполняется узлами по мере движения снизу вверх по сети;

- 7) инициировавший рассылку узел обобщает эхо-ответы, поступившие от соседей, строит дерево смежности, остовное дерево и пересылает широкоэвентально сообщение по сети.

Модель «распределенные семафоры» позволяет осуществлять разделение общего ресурса между распределенными процессами при их выполнении. При реализации механизма распределенных семафоров можно не только разграничивать доступ к ресурсу, но и формировать очередь процессов, ожидающих этого доступа. Возможность организации очереди связана с понятием логических часов.

Логические часы – это целочисленный счетчик, увеличивающийся при реализации события. У каждого процесса есть свой логический счетчик, имеющий нулевое начальное значение. В каждом передаваемом сообщении, используемом в алгоритме распределенных семафоров, выделено поле для временной метки. Изменение значения счетчика осуществляется в соответствии со следующими правилами:

- 1) передавая сообщение, процесс присваивает его метке текущее значение логических часов (переменная l с) и увеличивает l с на 1;
- 2) получая сообщение с меткой времени ts , процесс присваивает своей переменной l с значение, максимальное из l с и $ts+1$, затем увеличивает свою переменную l с на 1.

По аналогии со стандартными семафорами распределенные семафоры оперируют с P и V операциями. При синхронизации доступа к ресурсу процессы обмениваются сообщениями следующего формата:

Идентификатор процесса	Дескриптор типа операции (P или V)	Метка времени
------------------------	---	---------------

При рассылке сообщений, процесс их генерирующий, сохраняет их временные метки. Т.к. очереди операций (P и V) всеми процессами обрабатываются одновременно, каждый процесс знает, когда он может занять общий ресурс (т.е. когда в очереди обрабатывается именно его операция P).

Алгоритм синхронизации процессов с помощью распределенных семафоров состоит из следующих шагов:

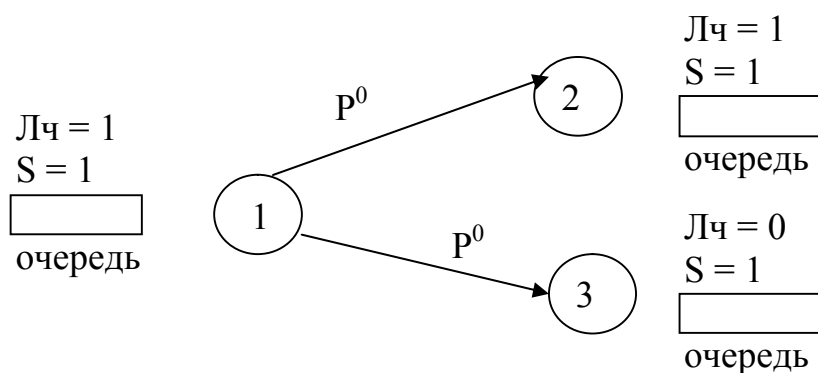
- 1) при необходимости доступа к ресурсу процесс рассылает всем остальным процессам сообщение с дескриптором P операции и меткой

времени. Это же сообщение получает и процесс – источник рассылки. Отсылка сообщения приводит к изменению значения логических часов на источнике.

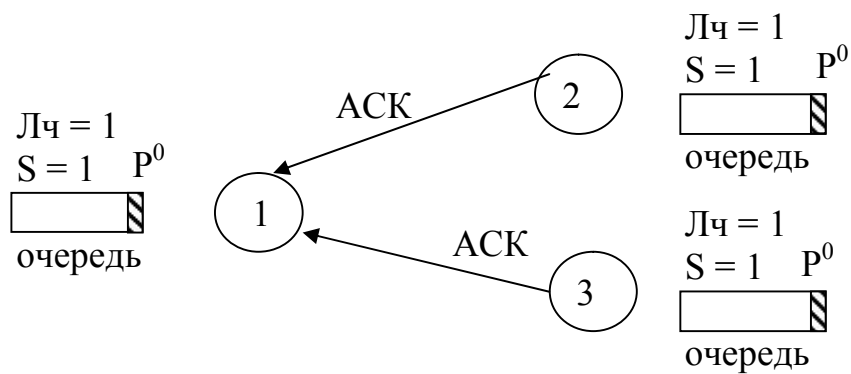
- 2) Принятие широковещательного сообщения (с P операцией) приводит к изменению логических часов на всех хостах (за исключением источника). Одновременно сообщение (с P операцией и временной меткой) устанавливается в общую очередь обработки. Очередь сообщений должна быть упорядочена в соответствии с временной меткой. Приемники широковещательного сообщения (P или V) подтверждают его принятие, отсылая источнику сообщение АСК, которое не изменяет временных меток ни на одном из хостов.
- 3) Разослав, а также выполнив принятие сообщения АСК, хосты кластера обрабатывают очередь сообщений. Для синхронизации доступа на каждом хосте выделяется локальная переменная S , определяющая доступность ресурса. Если в очереди встретилась операция V , она немедленно выполняется, переводя значение S в единичное состояние. P операции в очереди обрабатываются в соответствии с порядком поступления (временной метрикой).
- 4) Если переменная $S=1$, то всеми хостами выполняется первая P -операция, которая переводит значение S в 0. Далее эта операция извлекается (удаляется) из очереди. Тот процесс, который сгенерировал эту операцию (в соответствии с временной метрикой), получает доступ к ресурсу. В случае $S=0$ процесс доступа к ресурсу не получает и может продолжить свое выполнение до следующего сообщения с V операцией.

Реализовать алгоритм взаимодействия процессов с использованием распределенных семафоров поясняет рис. 2.2.

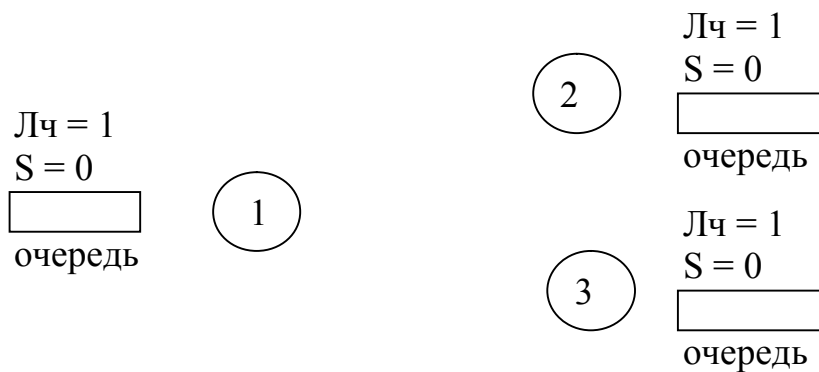
1. запрос захвата ресурса первым процессом



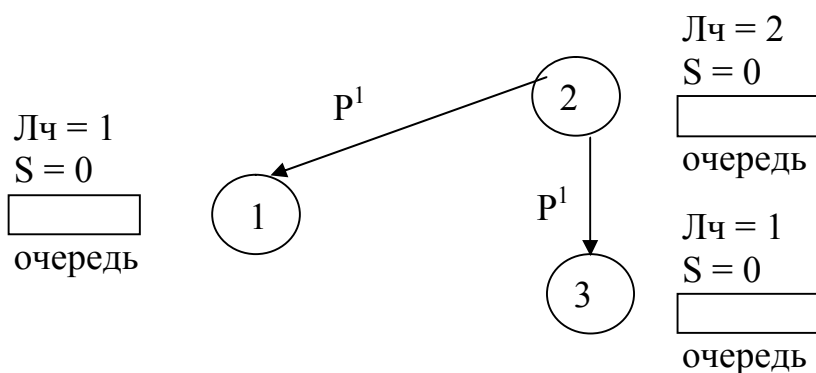
2. размещение сообщения P^0 в очереди и отсылка подтверждающих сообщений АСК (логические часы при принятии P сообщения увеличиваются на 1)



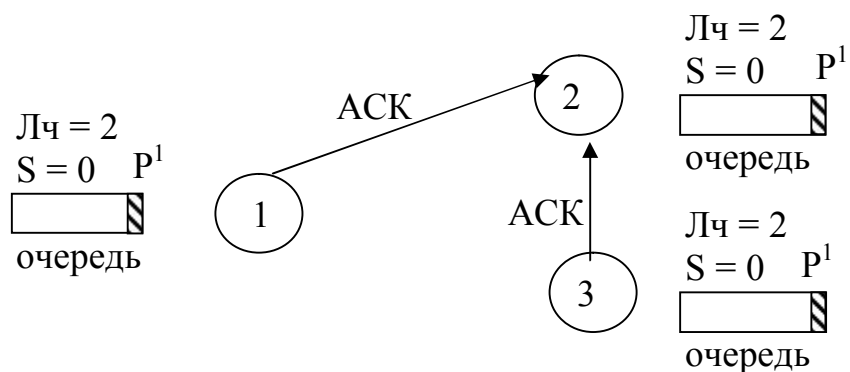
3. обработка очереди (P операция), удаление P операции из очереди, изменение значения переменной S



4. запрос процессом 2 ресурса. Отправка сообщения с идентификатором P-операции.

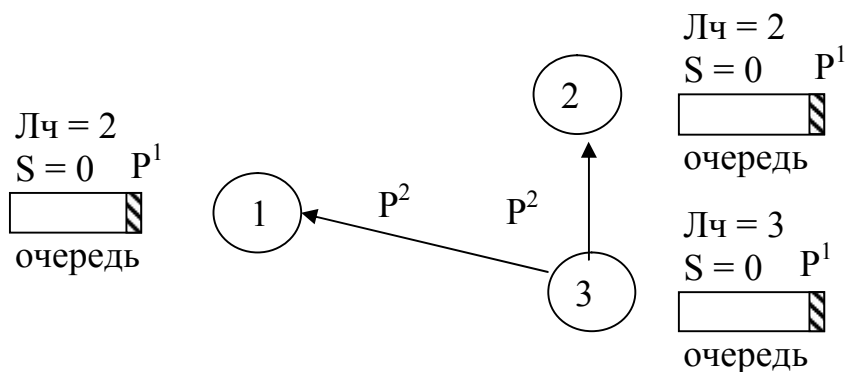


5. размещение полученного сообщения в очереди, подтверждение приемниками получения сообщения сигналом ACK.

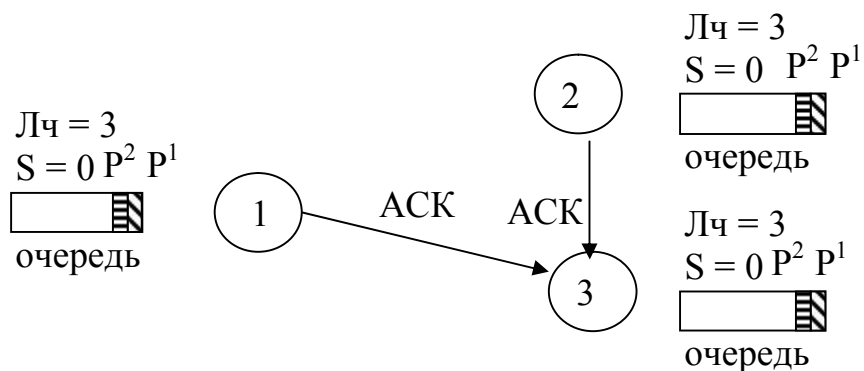


Т.к. $S = 0$, то процесс 2 блокируется в ожидании разрешения доступа к ресурсу. Сообщение P^2 остается в очереди.

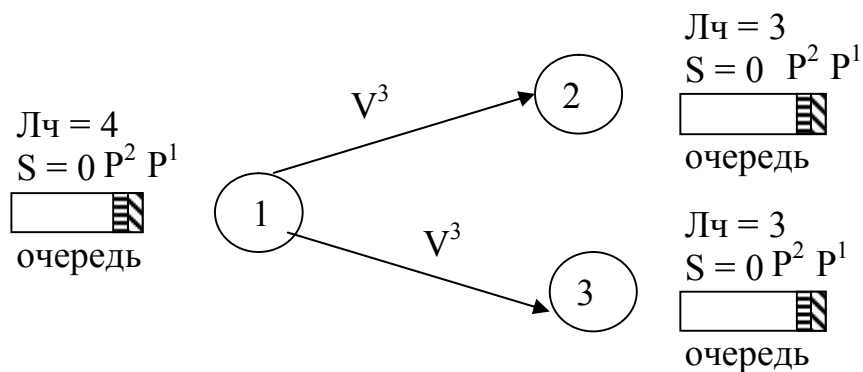
6. запрос процессом 3 ресурса. Отправка сообщения с идентификатором Р-операции.



7. размещение Р сообщения в очереди. Подтверждение принятия Р сообщения



8. обработка процессом 3 значения переменной S. Посылка процессом 1 сообщения V об освобождении ресурса.



9. обработка очереди: изменение значения переменной S на 1, выборка операции P^1 из очереди, изменение переменной S в значение 0. Разрешение доступа процессу, сгенерировавшему операцию P (процесс 2) доступа к ресурсу. Удаление операции P^1 из очереди.

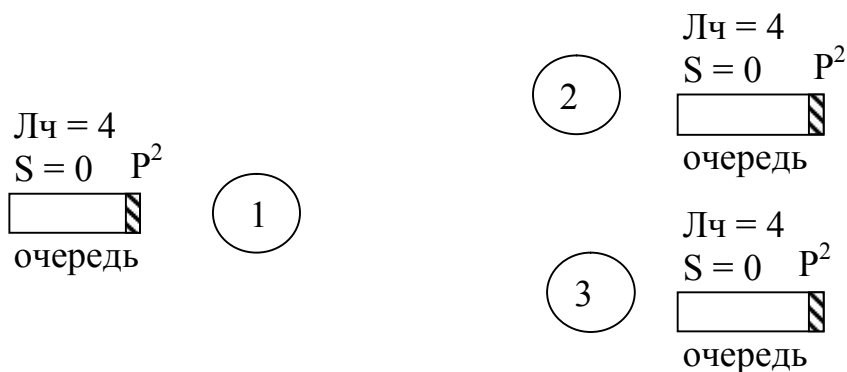


Рис. 2.2 Пример реализации алгоритма взаимодействия процессов с использованием распределенных семафоров

Алгоритм передачи маркера также предназначен для синхронизации доступа распределения выполняющихся процессов к общему ресурсу. В данном случае маркер – это сообщение специального вида, которое предназначено для передачи разрешения на доступ к ресурсу.

Особенностью построения алгоритма передачи маркера является то, что с каждым из распределено выполняющихся процессов, реализующих вычисления, сопоставлен специальный вспомогательный процесс, реализующий передачу маркера. Вспомогательные процессы образуют кольцо, т.е. процесс 1 передает маркер процессу 2, процесс 2 – процессу 3 и т.д., процесс N – процессу 1. Алгоритм распределения ресурса с использованием передачи маркера комментирует рис. 2.3.

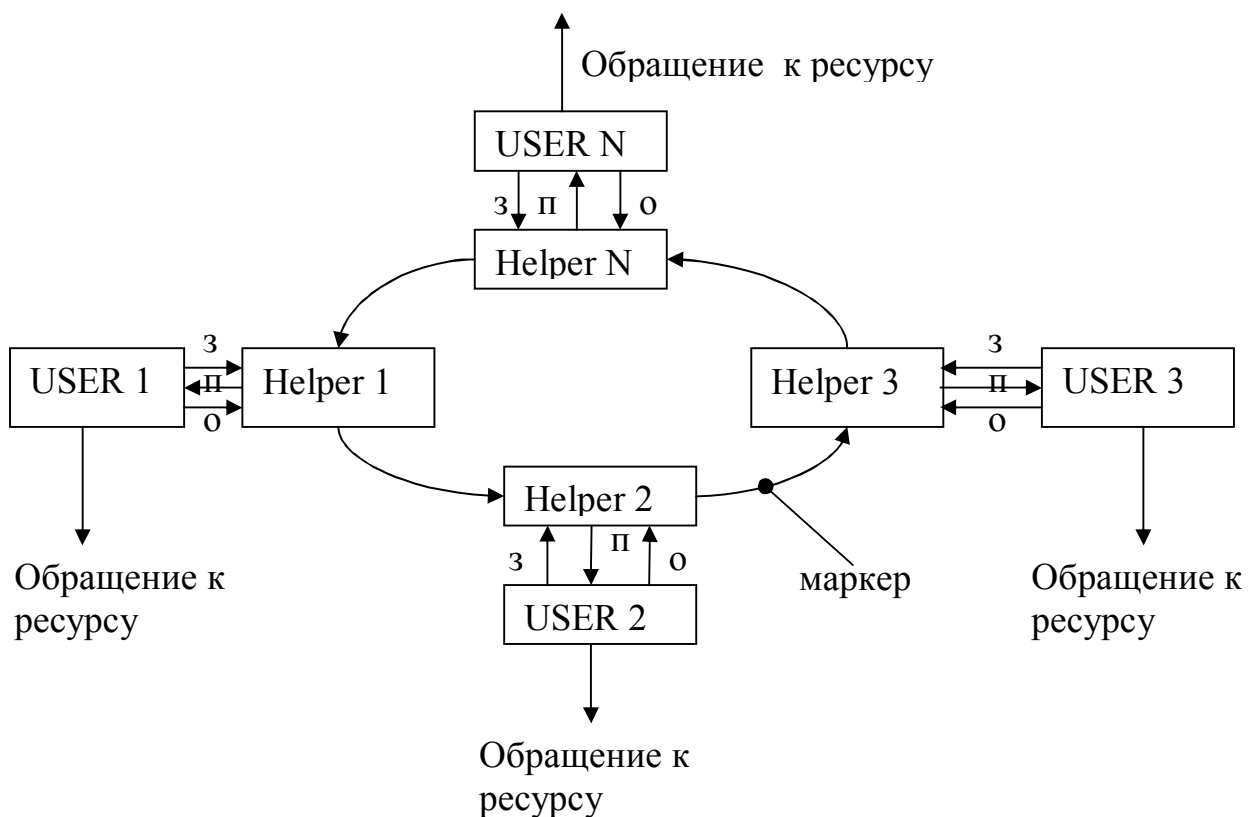


Рис. 2.3 Пример реализации алгоритма передачи маркера
 USER – вычислительный процесс, Helper – вспомогательный процесс, 3 – запрос ресурса, П – подтверждение возможности использования ресурса, О – сигнал освобождения ресурса

Алгоритм управления доступом процессов к ресурсу с использованием передачи маркера состоит из следующих шагов:

- 1) приняв маркер, процесс Helper [i] проверяет не осуществил ли вычислительный процесс запрос доступа к ресурсу. Если процесс User [i] выполнил запрос, то процесс Helper [i] далее маркер не передает, выдает процессу User [i] разрешение на использование ресурса и ждет от него сигнала об освобождении ресурса.
- 2) В том случае, если процесс User [i] не осуществил запрос на использование ресурса, процесс Helper [i] передает маркер (т.е. безаргументное сообщение) процессу Helper [i+1].

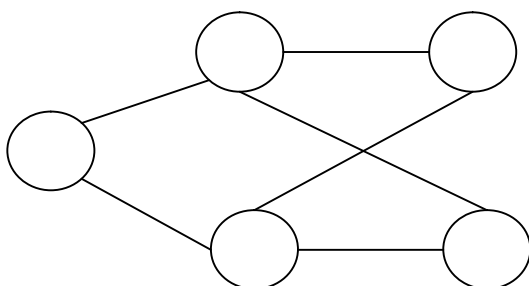
Таким образом, реализуется механизм исключительного доступа к ресурсу лишь одного из группы процессов.

3. Задание на работу.

Задание выбирается в соответствии с вариантом, назначаемым преподавателем.

1 Вариант

Осуществить построение топологии сети требуемого вида (рис. 3.1); выполнить широковещательную рассылку вводимого с клавиатуры сообщения от узла S на все остальные узлы.



На узле, инициирующем рассылку, выводить (в виде матрицы) топологию сети и остовное дерево, на остальных хостах сети после получения сообщения выводить номер хоста и сам текст сообщения.

Рис. 3.1

Схема сети для рассылки сообщений

2 Вариант

Распределено выполняются три процесса, каждый из которых вводит данные (целые числа) из локального файла по 5 элементов, выполняет суммирование 5 элементов и помещает полученную сумму в выходной файл. При этом все три процесса используют общую процедуру суммирования, в которую передают массив элементов, обратно получают сумму. Выполнить синхронизацию доступа трех процессов к общему ресурсу (процедуре суммирования), используя модель распределенных семафоров.

3 Вариант

Реализовать задание второго варианта, используя модель передачи маркера. Выводить сообщения комментирующие синхронизацию каждого вычислительного процесса (запрос и получение ресурса, освобождение ресурса), а также получение маркера в кольце.

4. Контрольные вопросы.

4.1 Особенности реализации и использования рассматриваемых моделей взаимодействия распределенных процессов.

4.2 Понятие топологии сети и остовного дерева. Форматы рассылаемых по сети сообщений. Алгоритмы построения топологии сети и остовного дерева, алгоритм рассылки.

4.3 Алгоритм реализации механизма «Распределенных семафоров». Назначение логических часов и очереди сообщений. Форматирование сообщений.

4.4 Алгоритм реализации модели «передачи маркера».

