

Лабораторная работа №4

Исследование и моделирование взаимодействий распределенных процессов типа "клиент-сервер" с использованием подпрограмм библиотеки PUM.

1. Цель работы: изучить механизм взаимодействия распределенно выполняющихся параллельных процессов типа "клиент-сервер" и средств их реализации в библиотеке PUM.

2. Теоретическое введение.

Одной из типичных схем взаимодействия между процессами в параллельных (распределенных) программах является взаимосвязь "клиент-сервер". Процесс-клиент при этом запрашивает некоторый сервис, затем ожидает обработку запроса. Процесс-сервер многократно ожидает запрос (на использование ресурса), обрабатывает его и посылает ответ. При этом существует двунаправленный поток информации от клиента к серверу и обратно. Т.о. сервер – это процесс, постоянно обрабатывающий запросы от клиентских процессов, используя при этом асинхронную передачу сообщений. Введенная концепция поясняется на Рис 2.1.

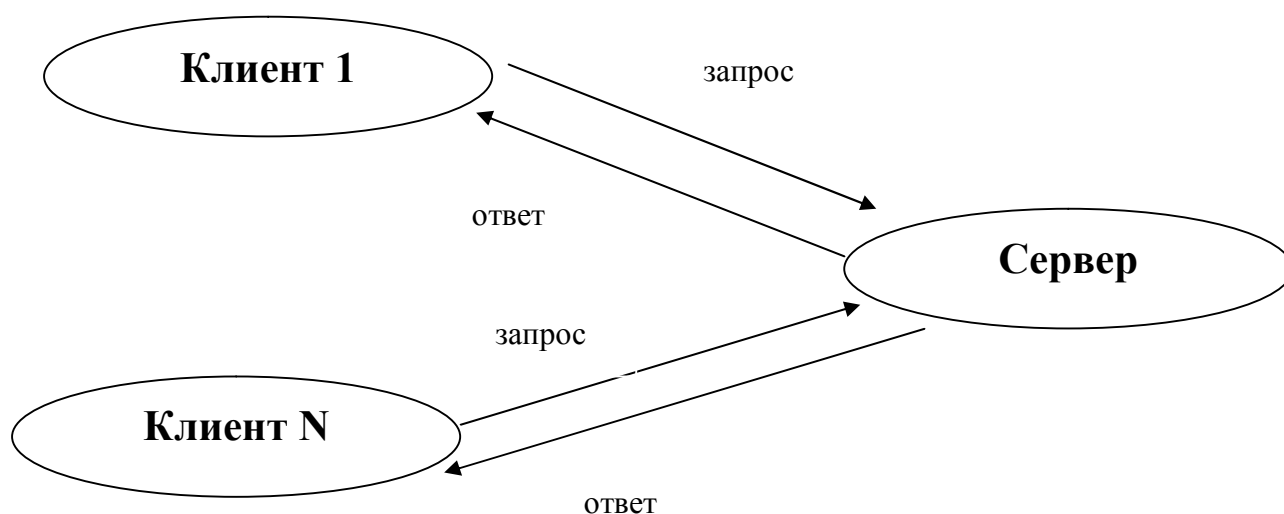


Рис 2.1- Схема взаимодействия типа "клиент-сервер"

В распределенных системах (системах с распределенной памятью) сервер реализуется набором подпрограмм, обеспечивающих доступ к ресурсу или структурам данных, представляющих собой ресурс.

Взаимодействие между клиентом и сервером реализуется вызовом соответствующих процедур доступа к ресурсу.

Серверы могут быть реализованы в соответствии со следующими схемами управления:

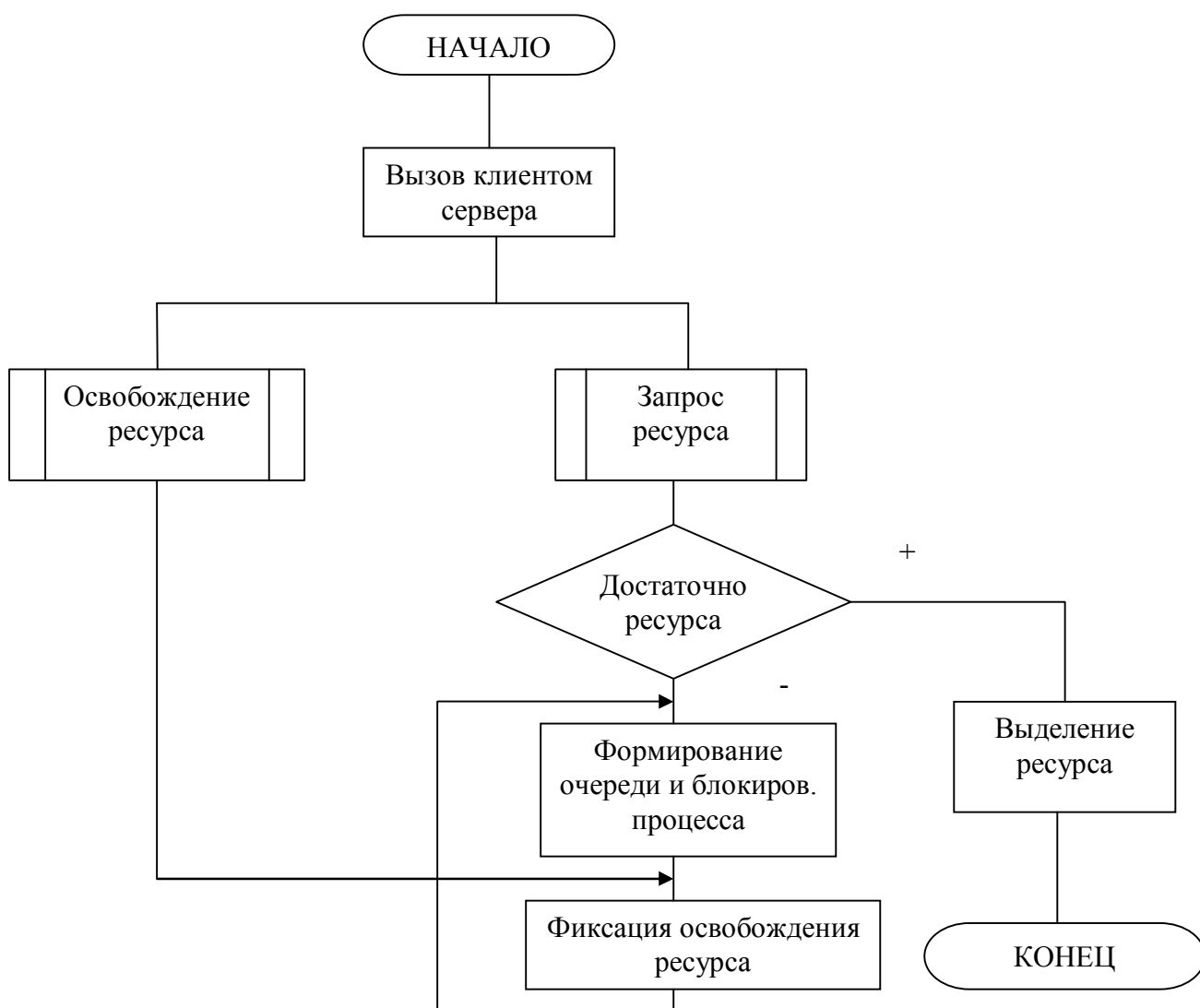
- 1) с помощью рассылки сообщений;
- 2) с использованием рандеву.

Взаимодействие клиента и сервера с помощью рассылки сообщений при выделении ресурсов инициируется следующим образом: клиентский процесс отправляет сообщение в канал запроса, а затем получает ответ (результат) из канала ответа. При этом каждому клиенту выделяется собственный канал ответа.

В структуре сообщения должно быть определено, какую опцию вызывает клиент (запрос или освобождение ресурса). Необходимо также передавать аргументные части сообщения (номер выделяемого ресурса и его запрашиваемое количество, если ресурс выделяется не целиком, а по частям). Когда нет доступных элементов ресурса, сервер не может ожидать, обслуживая запрос с выделением ресурса. Он запоминает запрос и откладывает посылку ответа (тем самым блокируя клиента). Когда ресурс освобождается, сервер возвращается к сохраненному запросу и передаем освободившийся элемент запрашивающему процессу. После отправки сообщения с запросом на выделение ресурса клиент ждет получения (элемента ресурса освобождая ресурс, клиента не ждет подтверждения его освобождения).

Серверу приходится сохранять ожидающий запрос, а также впоследствии проверять очередь ожидающих процессов.

Обобщенные алгоритмы функционирования клиента и сервера представлены на Рис 2.2.



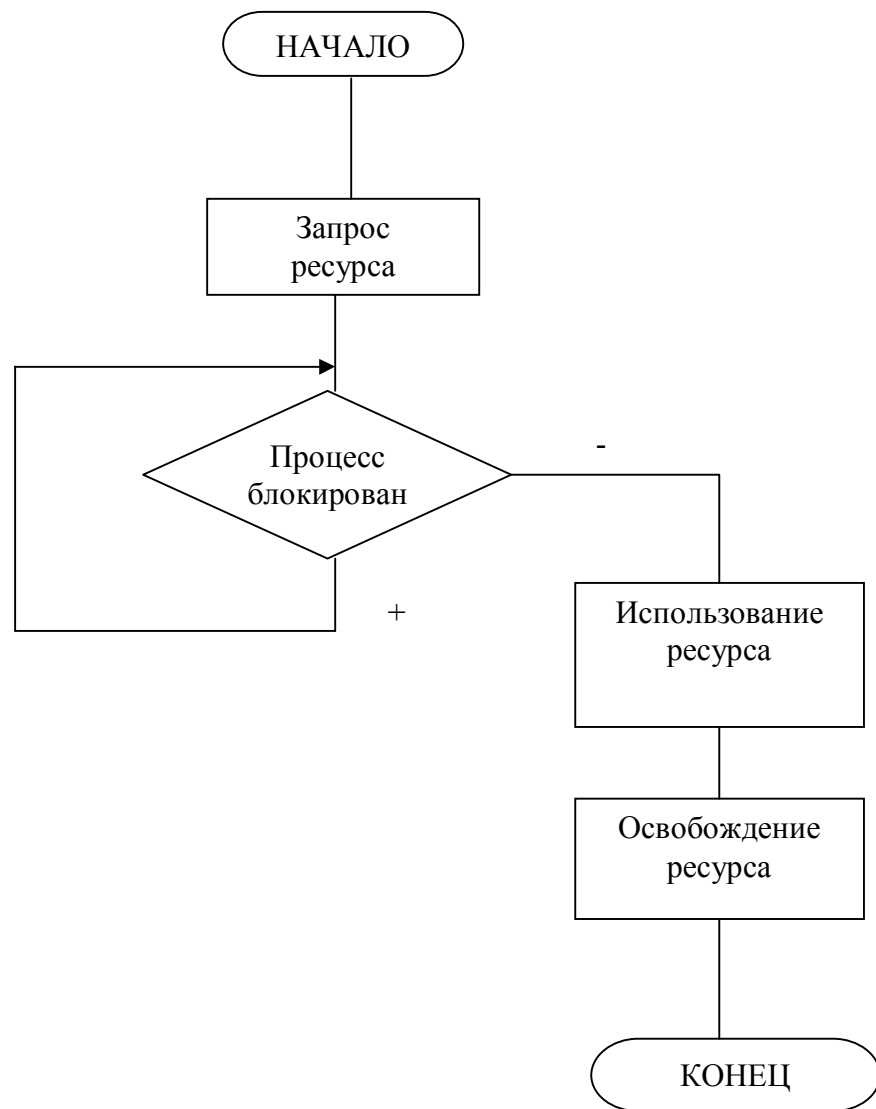


Рис 2.2 – Алгоритмы процедур функционирования сервера и клиента.

Т.о. при взаимодействии клиента и сервера при разделении ресурсов осуществляется обмен между ними тремя сообщениями (запрос, подтверждение, освобождение) и выделение ресурса клиенту, если он свободен.

Концепция рандеву предполагает инициализацию выполнения некоторой процедуры, доступ к которой регулируется сервером, в ответ на запрос клиента. Действия этой процедуры и клиентов необходимо синхронизировать (клиенты-поставщики информации для обработки этой процедурой). Т.о. процедуре и клиенту необходимо рандеву, которое синхронизируется сервером.

В отличие от первой схемы взаимодействия клиент, запрашивая ресурс, ждет от него непосредственных результатов обработки, и сообщение о завершении этой обработки отправляет серверу сама вычислительная процедура. Как и в описанном выше случае сервер обслуживает любой процесс, запросивший ресурс и клиента должен ждать освобождения ресурса.

Существует два способа решения задачи диспетчеризации при организации серверов:

- 1) отделение сервера от ресурса. В этом случае клиент осуществляет вызов сервера для получения доступа к ресурсу (запрос ресурса) и в случае получения доступа непосредственно обращается к ресурсу. В этом случае используется пять сообщений.
- 2) Сервер является посредником между клиентом и ресурсом (процессом обслуживания ресурса). Клиенты вызывают сервер, указывая в нем требуемый вид обработки ресурса, далее сервер сам обращается к ресурсу, разграничивая доступ клиентов.

В данном случае инициируется передача 4 сообщений. На Рис 2.3 приведены указанные схемы обмена сообщениями.

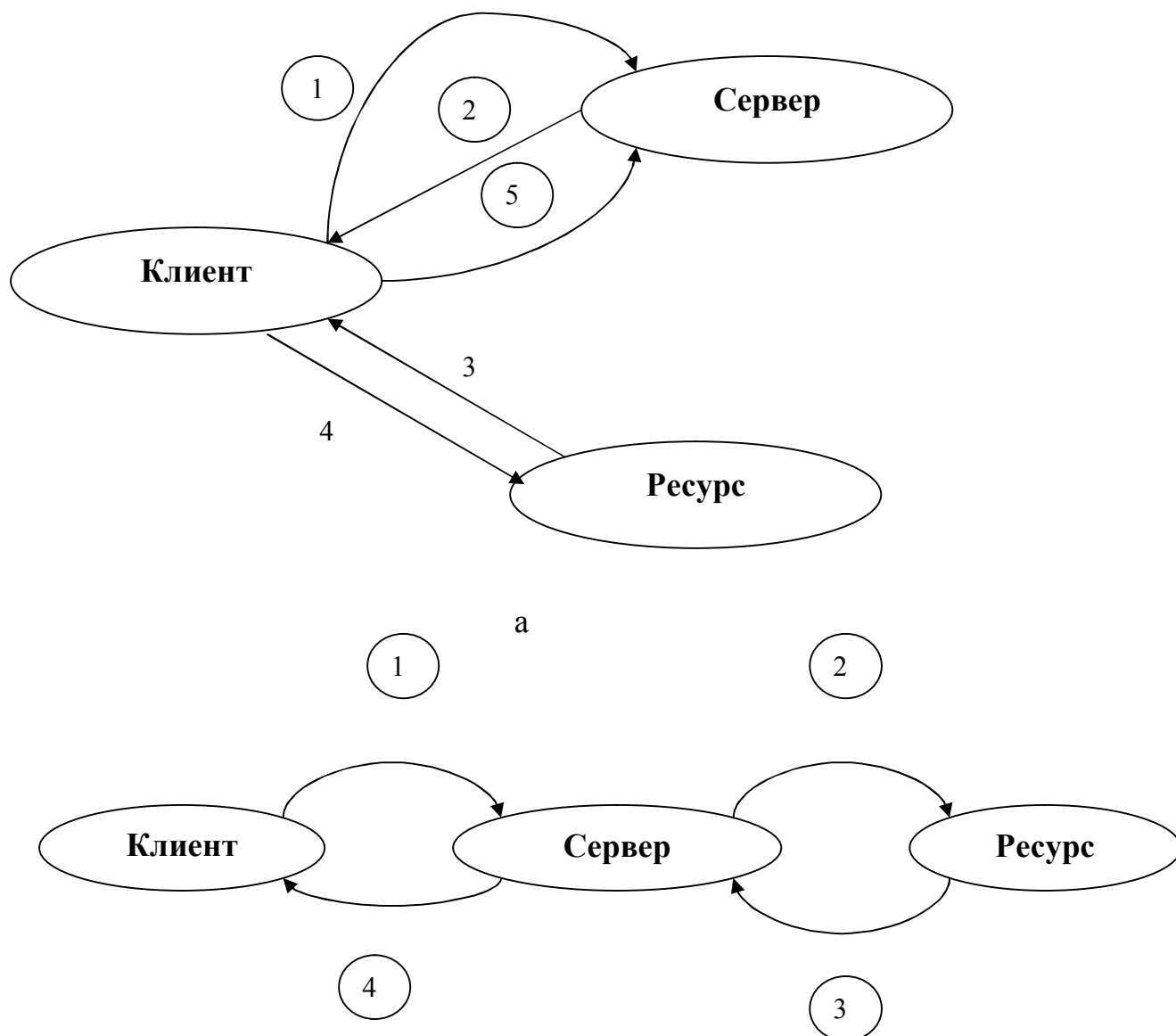


Рис 2.3 – Схемы организации доступа клиентов к ресурсу.

В случае, если ресурс представляет собой совокупность дискретных элементов и выделяется клиентом не целиком, а по частям, сервером должны быть использованы некоторые информационные структуры:

1) Таблица ресурсов, каждый элемент которой должен содержать следующие поля (Рис 2.4):

№ ресурса	Информация	
	Количество свободного ресурса $V_{св}$	Ссылка на очередь клиентов

Рис 2.4 – Вид таблицы ресурсов сервера.

а) количество свободного ресурса, которое сравнивается с количеством требуемого клиенту ресурса, передаваемом в запросе. В случае превышения свободного количества над требуемым ресурс выделяется в использовании клиентом. В противном случае клиент блокируется. При выделении ресурса клиенту вносятся изменения в количество свободного ресурса.

2) Очередь ожидающих данный ресурс клиентов. Это очередь может представлять собой массив структур, каждая строка которого соответствует номеру ресурса, а каждый элемент которого имеет вид, представленный на Рис 2.5.

Идентификатор процесса	Требуемое количество ресурса
------------------------	------------------------------

Рис 2.5- Вид элемента очереди к ресурсу

В этом случае действия сервера при выполнении запроса и освобождения ресурсов могут быть определены в виде алгоритмов, представленных на Рис 2.6

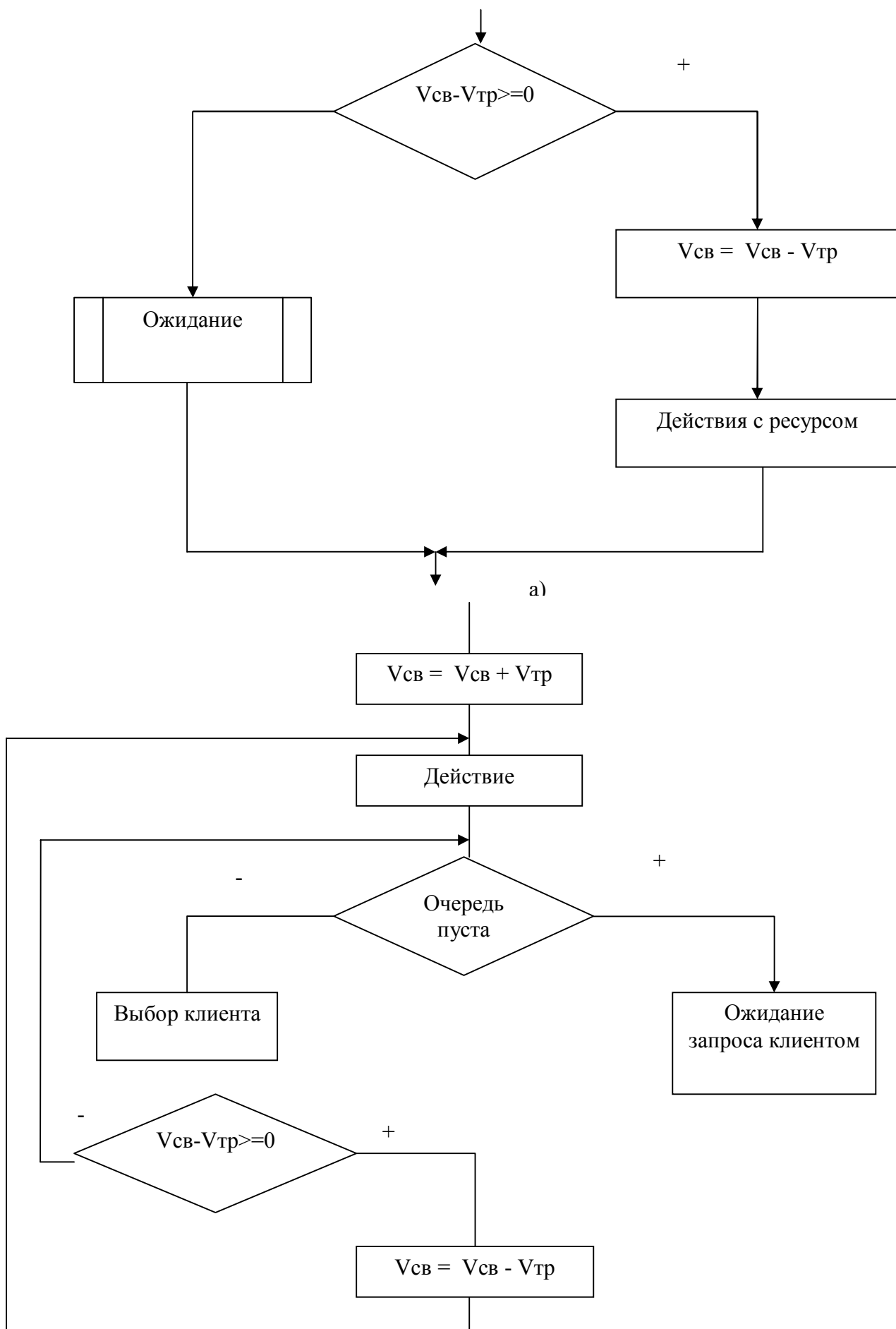


Рис 2.6 – Алгоритм контроля информационных структур сервера при запросе и освобождении ресурса

3. Задание на работу

задание выбирается в соответствии с вариантом, назначенным преподавателем

1 Вариант

В состав вычислительного кластера входит три хоста, один из которых реализует функции сервер, два – клиенты.

Сервер разграничивает доступ к трем общим ресурсам – нерешенным, хранящим общую вырученную сумму от продажи товаров, общее количество товаров и остальных товаров. Доступ к ресурсам осуществляется в произвольном порядке, все ресурсы разделяются между клиентами по отдельности. Реализована процедура выделяющая ресурсы (путем передачи сообщения) в использование клиентам. Реализовать серверный процесс, который разграничивает доступ клиентов к этой процедуре (процедурам) и к ресурсам. Реализацию выполнять в соответствии со схемой управления, использующую рассылку сообщений.

2 Вариант

С разграничивает доступ двух клиентов к общей процедуре. Клиент запрашивает у сервера доступ к процедуре и передает ей данные для расчета. Процедура выполняет простейшие операции суммирования введенных элементов.

При выполнении задания в основу построения сервера положить концепцию рандеву с оповещением процедурой сервера об ее освобождении при обработке данных. Реализовать две схемы построения сервера с отделением сервера от ресурса и сервера посредника между клиентом и ресурсом. Результаты работы процедуры возвращаются в вызванные клиентские процессы.

3 Вариант

Сервер разграничивает доступ со стороны двух клиентов к общей базе данных (массиву структур). При этом указанный ресурс рассматривается как дискретный, т.е. по запросу клиента ему может быть выделено столько записей, сколько необходимо. Информация из БД считывается обрабатывающей программой, доступ которой к БД также разграничивается сервером. Общее количество записей в БД равно N. Для синхронизации доступа к БД используются целые переменные P1 И P2 – определяющие первую занятую ячейку и первую пустую ячейку, P3 – число заполненных ячеек. Возможно использование следующих сигналов: запрос требуемого количества ресурса, освобождение ресурса после считывания, "буфер пуст" для потребителя и "буфер полон" для производителя. При построении сервера реализовать концепцию передачи сообщений. При реализации отделить сервер от ресурса, а также реализовать сервер как посредник между клиентами и ресурсом.

4. Контрольные вопросы.

- 4.1 Назначение серверного процесса при взаимодействии с клиентами;
- 4.2 Схемы управления при реализации сервера. Содержание и особенности схем управления;
- 4.3 Обобщенный алгоритм функционирования сервера;
- 4.4 Структура сообщения при обмене между клиентом и сервером;
- 4.5 Схема реализации доступа клиентов к ресурсу через сервер;
- 4.6 Информационные структуры сервера при разделении доступа к дискретным ресурсам.