**Распред.прилож**.предполагает наличие одной проги сервера, которая одновременно обслуживает несколько программ клиентов. Управление сервером осущ.с помощью спец.проги, которая называется **консолью упр-ния.**

Серверы, одновременно обслуживающие несколько клиентов, по методу обслуживания подразделяются на **итеративные** и **параллельные.**

Итеративный (исп., если предполаг.короткие запросы от К, не требующие больших затрат на обработку и длинные ответы С) – как правила работают над UDP, когда нет необх-сти создавать отдельный канал связи для каждого клиента.

1. Ожидание запроса от клиента
2. Обработка запроса
3. Отправка результата запроса
4. Возврат в ждущее состояние 1.

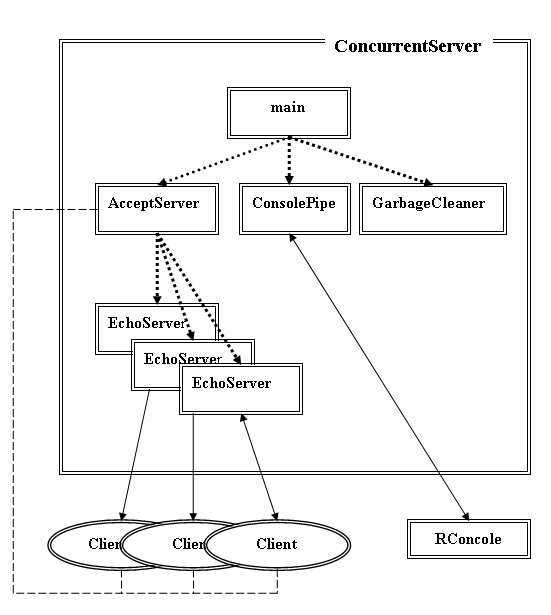
Параллельные (порождает новые сервера, кот.занимаются обработкой запросов клиентов) – исп-ть если предполагается наличие относительно длительного сеанса связи между К и С. Как правило, работают над TCP.

1. Ожидание запроса от клиента
2. Запуск нового сервера для обработки тек.запроса
3. Возврат в ждущее состояние 1.

**Паррал.С** – в рамках одного сервера работает несколько процессов, работающих параллельно и выполняющих специфические функции в рамках сервера.

**Процесс севера** – часть проги сервера работающая || с другими частями независимо от способа реализации.

Критическим по времени для параллельного сервера является момент подключения клиента. Сервер не должен тратить много времени на подключение клиента, т.к. в этот момент могут осуществлять подключение другие клиенты, которые из-за занятости сервера могут получить отказ. Поэтому целесообразно в сервере выделить отдельный процесс (и желательно, чтобы он имел наивысший приоритет), который бы был занят только подключением клиентов к серверу.



Назначение параллельного сервера – одновременное обслуживание нескольких клиентских программ. Обслуживание заключается в получении от клиента по установленному TCP-соед.посл-сти символов и возврате этой последовательности обратно. Сервер может выполнять команды введенные с консоли управления

Пунктивные линии – создание (запуск) одного процесса другим.

Сплошная линия – перемещение данных (между echo и клиентами, а также между RConsole и процессов ConsolePipe, кот.реализ.серверную часть консоли упр-ния)

\***Main** – главный процесс, кот.получает управление от ОС. Создает и запускает след.3 процесса; запускает, иниц и завершает работу сервера.

\***AcceptServer** – создает процессы EchoServer; предназначен для выполнения процедуры подключения клиентво к серверу для исполнения команд консоли управления, а ткже для запуска процессов EchoServer, кот.обслуж.запросы клиентов. Также создает список поделючений и кождый раз хранит инфу о состоянии данного подключения.

\***ConsolePipe** – сервер ИК, по которому осуществляется связь между Rconsole и параллельным сервером.

\***GarbageCleaner –** удаление элемента из списка подключений после отключения команды клиента (!нельзя одновременно записывать и удалять д-е о подключении – разрушение списка)

\***EchoServer** – созда.тся AcceptServer по одному для каждого успешного подключения клиента. Назначение – прием даннных по созданному каналу + отправка этих же д-х клиенту без изменений

\***Client –** для пересылки данных серверу и полсения ответа; окончание – семгент 0ой длины

\***Rconsole** – для ввода команд управления сервером и вывода диагностических сообщений полученных от сервера. Является клиентом ИК.

\***ListContact** – для хранения инфы о каждом подключении; создается пустым при иниц.процесса AcceptServer. Тут же осущ.добавление элементов списка для каждого подключения.