**Реферат на тему «Сетевые технологии» - Каспер Наталья**

**Сервер –** компьютер сети, который предоставляет свои программные и аппаратные ресурсы юзерам сети для хранения данных, выполнения программ и др.

**Клиент –** компонент архитектуры «клиент-сервер»», который пользуется услугами сервера. Чаще всего клиент = программы, которые имеют доступ к информационным ресурсам или устройствам сервера. Для подключения юзер должен получить собственное регистрационное имя и пароль.

Преимущества архитектуры «клиент-сервер»:

* Централизованное управление ресурсами сети
* Безопасность доступа
* Скорость доступа

**Роль сервера** – обеспечение централизованной защиты и управление трафиком, а также предоставление клиентам ресурсов.

В клиент-серверной среде в роли клиентов выступают ПК. Как правило, клиент использует собственные вычислительные мощности для обработки инфы, полученной от сервера, но полагается на сервер для предоставления необходимых данных. Такое распределение ролей в обработке инфы – **клиентская и серверная обработка.**

**Серверная ОС** концентрируется на управлении ресурсами, а **клиентская** – на выполнении задания с максимальной скоростью и эффективностью. Добиться максимально возможной скорость можно с помощью «трех м»: *многопоточности, многозадачности и многопроцессорности.*

**Многопоточность**

Это метод оптимизации производительности, который основан на том, что процессор работает с высокой скоростью, измеряемой тактами. Такты выполняются независимо от того, занят процессор или нет.

При такой обработке процесс подразделяется на потоки, каждый из потоков выполняется микропроцессором по отдельности.

**Многозадачность**

Выгодна для повышения производительности. Сущность: каждому процессу предоставляется отдельное время, а кажется, что процессы идут параллельно. Такой эффект достигается за счет высокой скорость работы процессора и способности перемешивать выделенные интервалы времени. Также, нужно заметить, что для компьютера и ОС время идет намного быстрее, чем для людей.

Если многозадачность реализована на клиенте – обеспечивается лучшая координация взаимодействия сервера с клиентом и управлением им с большей эффективностью.

**Многопроцессорность**

Необходима для того, чтобы система успешнее справлялась с задачами за счет того, что нагрузка перекладывается на много параллельных процессоров.

Многопроцессорная обработка мб:

* Симметричная – любой процесс мб поручен любому, свободному в данный момент, процессору
* повышается отказоустойчивость
* улучшенная балансировка нагрузки
* Асимметричная – нагрузка распределяется так, что 1/неск процессоров обслуживают только ОС, а остальные работают с приложениями.

Для работы в лок.сети создаются проекты для клиента и сервера раздельно. При этом сервер дб запущен первым.

Рассмотрим наиболее важные функции:

|  |  |
| --- | --- |
| **Наименование**  **функции** | **Назначение** |
| **accept** | Разрешить подключение к сокету |
| **bind** | Связать сокет с параметрами |
| **closesocket** | Закрыть существующий сокет |
| **connect** | Установить соединение с сокетом |
| **listen** | Переключить сокет в режим прослушивания |
| **recv** | Принять данные по установленному каналу |
| **recvfrom** | Принять сообщение |
| **send** | Отправить данные по установленному каналу |
| **sendto** | Отправить сообщение |
| **socket** | Создать сокет |

Для работы данных методов необходима библиотека Ws\_32.lib.

#pragma comment(lib, "Ws2\_32.lib")