

Сокеты предоставляют весьма мощный и гибкий механизм межпроцессного взаимодействия. Сокеты обеспечивают двухстороннюю связь типа ``точка-точка" между двумя процессами. Они являются основными компонентами межсистемной и межпроцессной связи. Каждый сокет представляет собой конечную точку связи, с которой может быть совмещено некоторое имя. Он имеет определенный тип, и один процесс или несколько, связанных с ним процессов.

С каждым сокетом связываются три атрибута: домен, тип и протокол. Эти атрибуты задаются при создании сокета и остаются неизменными на протяжении всего времени его существования. Для создания сокета используется функция `socket`, имеющая следующий прототип: **`int socket(int domain, int type, int protocol);`**

Домен определяет пространство адресов, в котором располагается сокет, и множество протоколов, которые используются для передачи данных. Тип сокета определяет способ передачи данных по сети. Последний атрибут определяет протокол, используемый для передачи данных. Часто протокол однозначно определяется по домену и типу сокета. В этом случае в качестве третьего параметра функции `socket` можно передать 0, что соответствует протоколу по умолчанию. Тем не менее, иногда требуется задать протокол явно. Прежде чем передавать данные через сокет, его необходимо связать с адресом в выбранном домене. Для явного связывания сокета с некоторым адресом используется функция `bind`.

Основные типы сокетов

Поточный - обеспечивает двухсторонний, последовательный, надежный, и недублированный поток данных без определенных границ. Тип сокета - `SOCK_STREAM`, в домене Интернета он использует протокол TCP.

Датаграммный - поддерживает двухсторонний поток сообщений. Приложение, использующее такие сокеты, может получать сообщения в порядке, отличном от последовательности, в которой эти сообщения посылались. Тип сокета - `SOCK_DGRAM`, в домене Интернета он использует протокол UDP.

Сокет последовательных пакетов - обеспечивает двухсторонний, последовательный, надежный обмен датаграммами фиксированной максимальной длины. Тип сокета - `SOCK_SEQPACKET`. Для этого типа сокета не существует специального протокола.

Простой сокет - обеспечивает доступ к основным протоколам связи.

Все сокеты обычно ориентированы на применение датаграмм, но их точные характеристики зависят от интерфейса, обеспечиваемого протоколом.

Установка соединения на стороне сервера состоит из четырех этапов. Сначала сокет создаётся и привязывается к локальному адресу. На следующем шаге создаётся очередь запросов на соединение. При этом сокет переводится в режим ожидания запросов со стороны клиентов. Всё это выполняет функция `listen`. Функция `accept` создаёт для общения с клиентом новый сокет и возвращает его дескриптор. Один из параметров, `sockfd` задаёт слушающий сокет. После вызова он остаётся в слушающем состоянии и может принимать другие соединения. На стороне клиента для установления соединения используется функция `connect`. После того как соединение установлено, можно начинать обмен данными. Для этого используются функции `send` и `recv`. В Unix для работы с сокетами можно использовать также файловые функции `read` и `write`, но они обладают меньшими возможностями, а кроме того не будут работать на других платформах.

После окончания обмена данными, сокет закрывают с помощью функции `close`. Это приводит к разрыву соединения. Также можно запретить передачу данных в каком-то одном направлении, используя функцию `shutdown`. В зависимости от переданного параметра можно запретить запись в сокет, чтение из него или и то и другое.

Использование сокетов TCP/IP при большом количестве соединений

При использовании большого количества подключений неэффективно создавать по одному потоку на каждое подключение по сокету, т.к. значительное время потоки будут простаивать в

ожидании появления данных. Для более рационального использования времени процессора используются подходы:

- Использование функций, позволяющих опрашивать несколько сокетов одновременно (такие как `poll()` и `select()`). Таким функциям передается набор сокетов, и их вызов ждет появления указанного события на одном из переданных сокетов или истечения тайм-аута.
- Использование RT-сигналов. В программе при помощи функции `fcntl()` устанавливается получение некоторым потоком сигнала при приходе данных на указанный сокет. Поток в цикле вызывает функцию `sigwaitinfo()` для синхронного извлечения сигналов из очереди и определения сокета, связанного с сигналом. С одним потоком можно связать сигналы для нескольких сокетов.