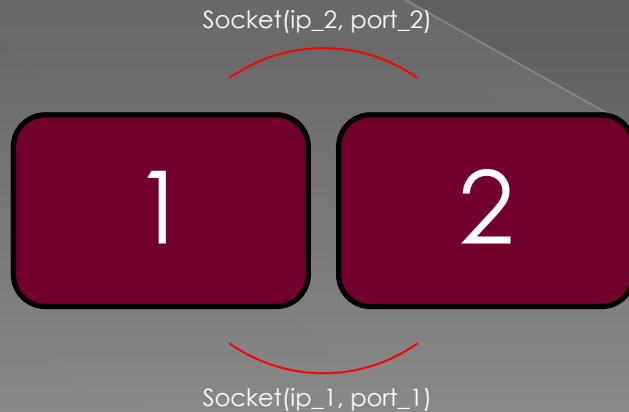


Сетевое взаимодействие через сокеты

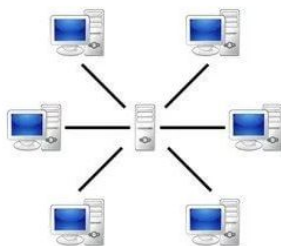
Сокеты

- Средство межпроцессного взаимодействия (на одной или разных машинах)
- Возможна передача данных по разным протоколам (TCP, UDP, RAW)
- Доступны и в UNIX-like, и в Windows

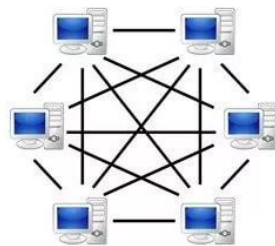


Клиент и сервер

- Клиент — участник межсетевого взаимодействия, который формирует запросы (сообщения) и принимает ответы на эти запросы
- Сервер — участник межсетевого взаимодействия, принимающий и обрабатывающий запросы от клиента
- P2P-сеть (peer-to-peer) — децентрализованная архитектура сети, в которой каждый участник взаимодействия является одновременно и клиентом, и сервером

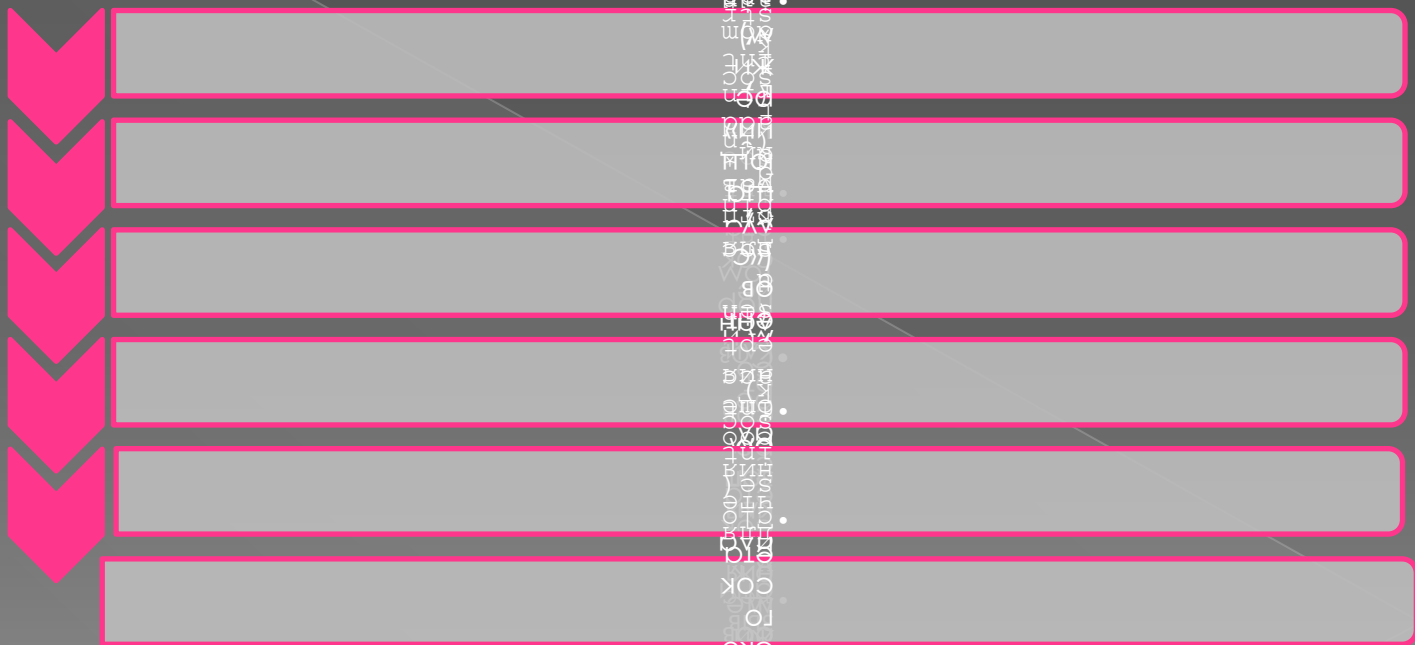


Server-based

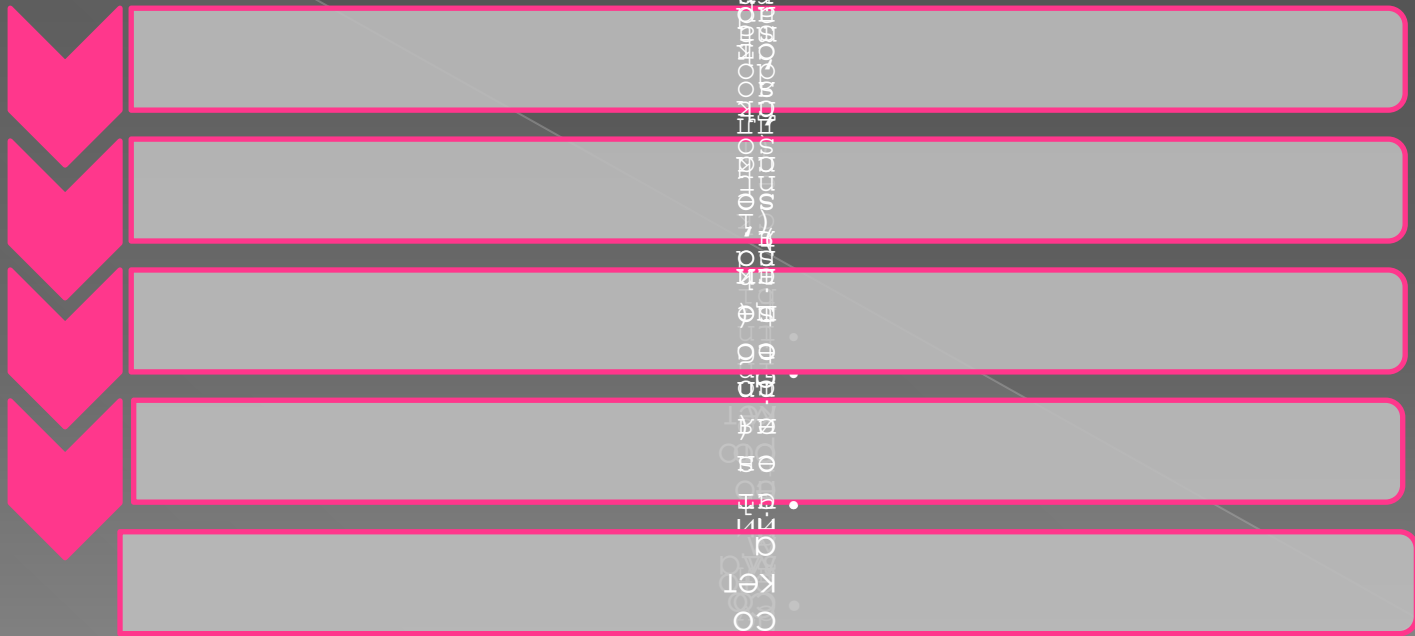


P2P-network

Алгоритм работы сервера



Алгоритм работы клиента



Заголовочные файлы в UNIX

- ◉ `#include <sys/types.h>`
- ◉ `#include <sys/socket.h>`
- ◉ `#include <netinet/in.h>`
- ◉ `#include <arpa/inet.h>`

Функции для работы с сокетами в Linux

- ◉ `int socket (int domain, int type, int protocol)`
 - > `domain = {AF_INET, AF_UNIX, AF_INET6, AF_IPX, ...}`
 - > `type = {SOCK_STREAM, SOCK_DGRAM, SOCK_RAW}`
 - > `protocol` - чаще всего 0
- ◉ `int bind (int sock, struct sockaddr * addr, int addrlen)`
 - > `Sock` - дескриптор сокета
 - > `Addr` - структура для хранения адреса
 - > `Addrlen = sizeof(addr)`
- ◉ `int connect(int sock, struct sockaddr * addr, int addrlen)`
 - > `Sock` - дескриптор сокета
 - > `Addr` - структура для хранения адреса (адрес сервера)
 - > `Addrlen = sizeof(addr)`
- ◉ `int listen(int sock, int queuesize)`
 - > `Sock` - дескриптор сокета, который будет переводиться слушающий режим
 - > `Queuesize` - размер входных подключений (размер очереди)
- ◉ `int accept(int sock, void * addr, int * addrlen)`
 - > `Sock` - дескриптор слушающего сокета
 - > `Addr` - структура для хранения адреса клиента (можно NULL)
 - > `Addrlen = sizeof(addr) (NULL)`
- ◉ `int close(int sock)`
 - > `Sock` - дескриптор сокета, который будет закрыт

Функции чтения и записи

- ◉ `int send(int sockfd, const void *msg, int len, int flags)`
 - > `Socketfd` – сокет, куда шлем
 - > `Msg` – сообщение
 - > `Len` – размер сообщения
 - > `Flags` – флаги (можно NULL)
- ◉ `int recv(int sockfd, const void *msg, int len, int flags)`
 - > `Socketfd` – сокет, откуда читаем
 - > `Msg` – буфер сообщения
 - > `Len` – размер буфера
 - > `Flags` – флаги (можно NULL)

Заголовочные файлы в Windows

- `#include <winsock2.h>`
- Линковать с `ws2_32` (`-lws2_32`)

Функции для работы с сокетами в Windows

- ◉ `int WSASStartup (WORD Version, LPWSADATA lpWSADATA)`
 - > `Version = MAKEWORD(2, 2)`
 - > `lpWSADATA` – почти бесполезная структура, которую нужно в начале создать
- ◉ `SOCKET socket (int domain, int type, int protocol)`
 - > `domain = {AF_INET, AF_UNIX, AF_INET6, AF_IPX, ...}`
 - > `type = {SOCK_STREAM, SOCK_DGRAM, SOCK_RAW}`
 - > `protocol` – чаще всего 0
- ◉ `int WSACleanup ()`
- ◉ Остальное идентично функциям в Linux

