Основы программирования сетевых приложений

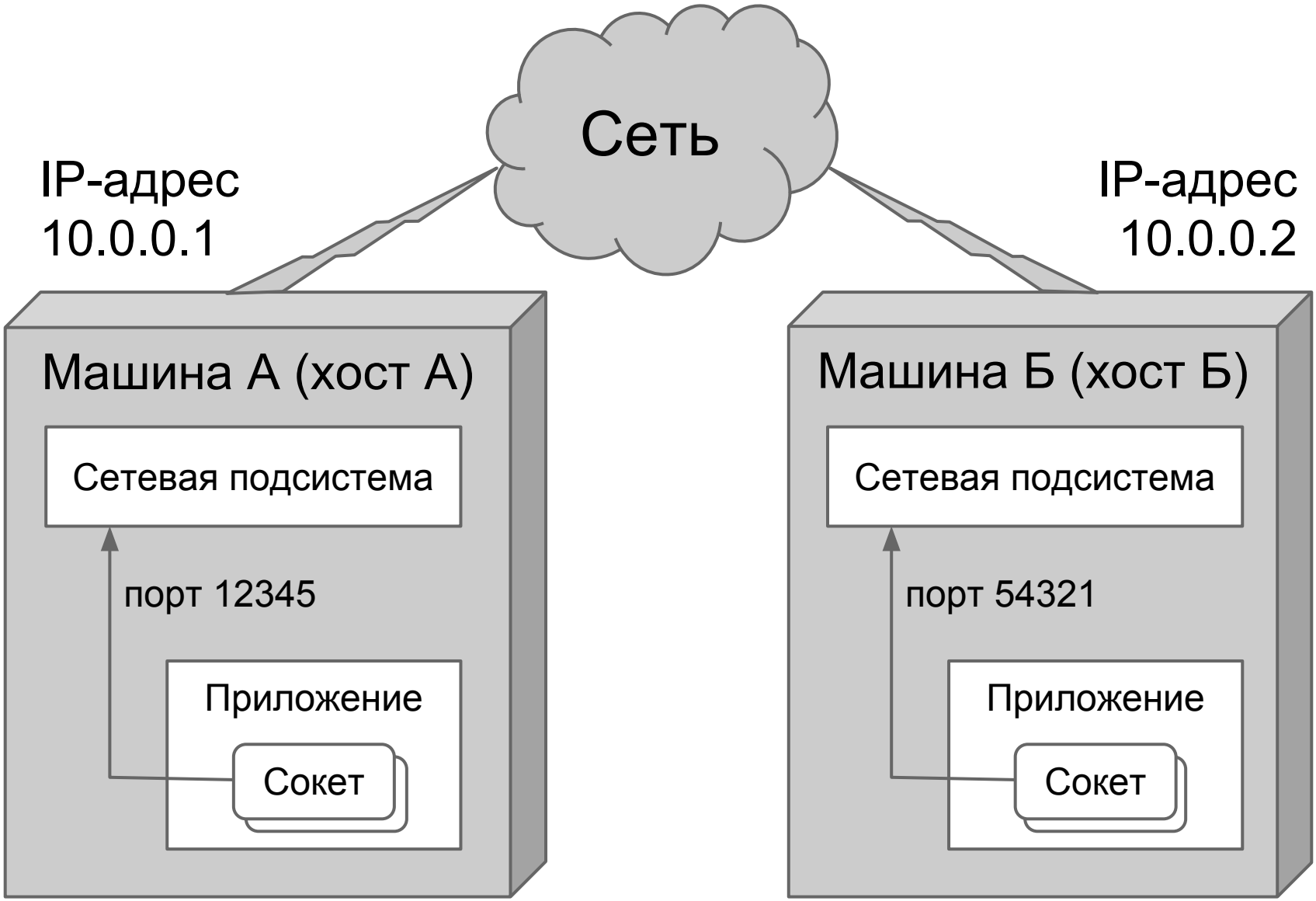
Курс «Информационные сети и телекоммуникации»

весенний семестр 2015 г.

кафедра Управления и информатики НИУ «МЭИ»

Сокет (socket)

- Сокет — это:
  - по-английски «розетка», «гнездо», «отверстие»;
  - идентификатор устройства для связи с удаленным узлом сети.
- В программе с сокетом связана переменная.
- Сокет — это:
  - НЕ соединение *(его может и не быть);*
  - НЕ особый вид файла,
    - но это особенное **устройство;**
  - НЕ общая сущность на связывающихся узлах *(2 машины, 2 программы, 2 переменные).*



Адрес узла (host) и порт (port)

- Хост идентифицируется адресом IP.
  - Адрес IP: 4 байта, записываются через точку
  - Особые адреса IP:
    - 0.0.0.0 — текущий (loopback)

      `INADDR_ANY` (`0x00000000`),
    - 255.255.255.255 — для рассылки всем узлам (broadcast)

      `INADDR_BROADCAST` (`0xFFFFFFFF`).
- У каждого узла 65536 портов
  - Порт идентифицирует приложение.
  - Порты 0 — 1023 зарезервированы для системы.
  - Сокет связан только с одним портом.
- Данные передаются между точками **(хост, порт)**.

Простейшая схема работы

1) Создать сокет.
2) Передать данные:
   a) отправить:
      — указать адрес получателя,
      — выполнить отправку;
   b) принять:
      — указать собственный адрес,
      — выполнить прием,
      — *(можно)* определить адрес отправителя.
3) Закрыть сокет.

Интерфейс программирования

- Berkley Sockets (BSD sockets), 1989 г. — набор функций в BSD UNIX 4.2.
- POSIX sockets (UNIX, Linux, Mac — *nix) — стандарт на интерфейс программирования:
  - Является «наследником» BSD sockets.
  - Используется не только для сетей.
- Windows Sockets (winsock):
  - Не полностью соответствует стандарту POSIX.
  - Много дополнительной функциональности (элементы WSA*), связанной **с** Windows API.

Создание сокета

Тип возвращаемого значения: UNIX, Linux, Mac — **`int`**
Windows — SOCKET

```
auto channel = socket(
    AF_INET, SOCK_DGRAM, IPPROTO_UDP);
```

Семейство адресов:

в каких сетях расположены узлы, будут задаваться их адреса

Вид сокета:

как будет вестись работа с сокетом из программы

Протокол:

как система будет передавать данные по сети

Семейство адресов

- Определяет способ адресации узлов, с которыми будет осуществляться связь.
- Address family (AF_xxx), или protocol family (PF_xxx).
- Популярные семейства:
  - AF_INET — «интернет», адреса IPv4 (127.0.0.1)
  - AF_INET6 — «интернет», адреса IPv6 (::1)
  - AF_LOCAL — локальные сокеты ОС *nix
  - AF_BTH (Bluetooth), AF_IRDA (ИК) в Windows

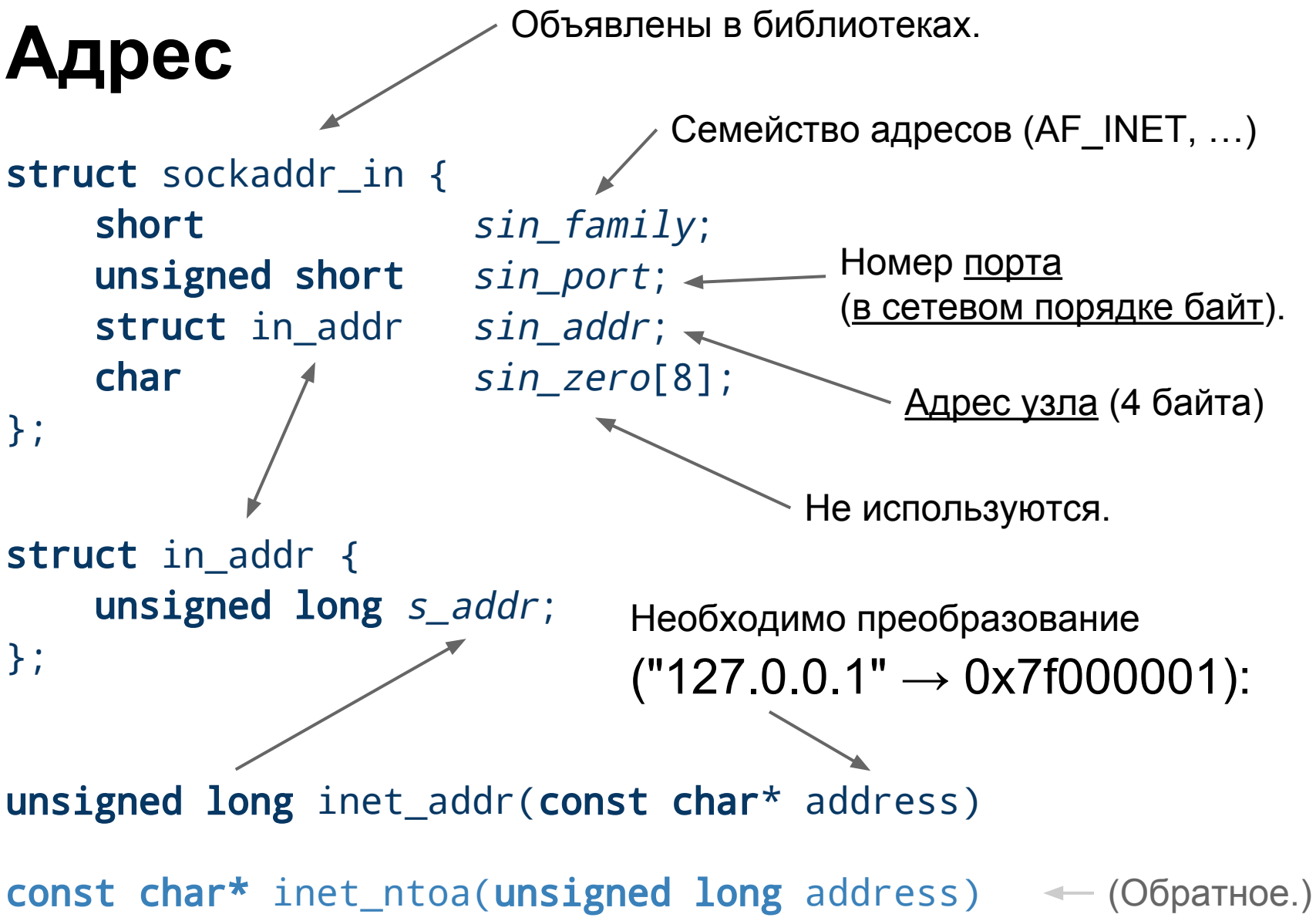
Вид сокетов

- Определяет, как приложению работать с сокетом.
- Пример: дейтаграммные (SOCK_DGRAM)
  - Соединения не устанавливается (connectionless)
  - Передача ведется блоками данных (datagram):
    - размер блока ограничен 512…4096 байтами;
    - блоки или приходят сразу целиком, или не приходят (не может прийти часть блока);
    - блоки могут прийти в ином порядке, чем были отправлены;
    - потери системой не фиксируются (unreliable — ненадежная доставка).

Протокол

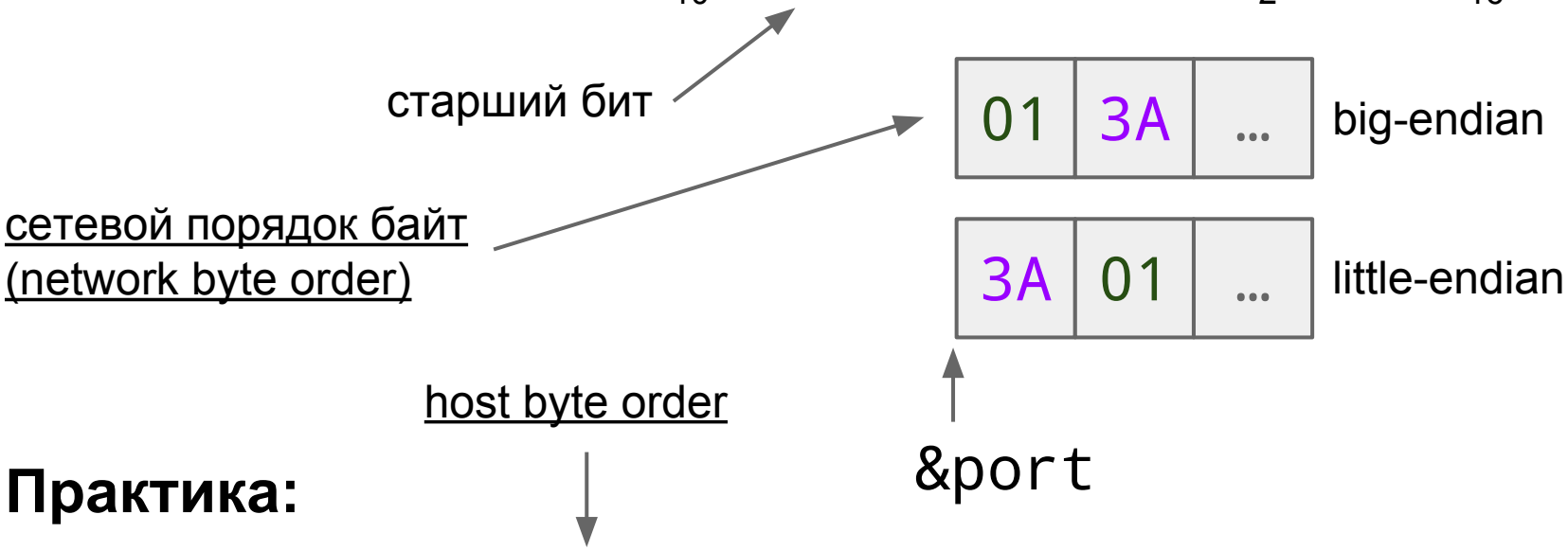
- Определяет, как система будет отправлять данные по сети
- Совместимы только определенные протоколы и виды сокетов:
  - SOCK_**DGRAM**, IPPROTO_**UDP**
  - SOCK_**STREAM**, IPPROTO_**TCP**
  - ~~SOCK_STREAM и IPPROTO_UDP~~

Адрес



Порядок байт (byte order, endianness)

unsigned short port = 314_{10} = 0000 0001 0011 1010_2 = 01 $3A_{16}$



Практика:

- Порядок байт на разных машинах отличается.
- `htons()`, `htonl()`, `ntohs()`, `ntohl()`

  h — host, n — network, s — short, l — long

Отправка данных

Целое со знаком:
`int` (Windows), `ssize_t` (*nix).

Количество отправленных байт
`(0 <= result <= size)`
при успехе;
`SOCKET_ERROR` (Windows)
или -1 (*nix) при ошибке.

```
auto result = sendto(
    channel,
    buffer,
    size,
    0,
    (const sockaddr*)(&destination),
    sizeof(destination));
```

Ранее созданный сокет.

Адрес данных для отправки.

Количество байт для отправки.

Флаги настроек (см. документацию).

Адрес получателя и его размер.

Привязка сокета (указание собственного адреса)

Необходима для приема, необязательна для отправки. Отменить нельзя.

Успех: `0`,
ошибка: `SOCKET_ERROR` (Windows) или `-1` (*nix).

```
int result = bind(
    channel,
    (const sockaddr*)&address,
    sizeof(address));
```

Ранее созданный сокет.

Структура-адрес `sockaddr_in`, на который должно быть отправлено сообщение, чтобы сокетом `channel` можно было его принять.

Размер структуры-адреса.

Прием данных (SOCK_DGRAM)

Размер принятой датаграммы, если она уместилась в буфер (`result <= size`).

`SOCKET_ERROR` или `(-1)`

Код ошибки:

`WSAEMSGSIZE` — датаграмма была длиннее `size`, буфер заполнен, остаток утерян.

другой — прием не удался.

```
auto result = recvfrom(
    channel,
    buffer,
    size,
    0,
    nullptr,
    nullptr);
```

Буфер под принятые данные.

Количество байт, которое можно записать в буфер.

Флаги настроек (см. документацию).

Нужны для получения адреса отправителя (необязательные параметры).

Прием данных (SOCK_DGRAM): получение адреса отправителя

```
sockaddr_in source;
int source_size = ...;
auto result = recvfrom(
    channel, buffer, size, 0,
    (sockaddr*)(&source), &source_size);
```

Структура-адрес.

Размер в байтах структуры `sockaddr_in` или переменной `source`.

Windows; `socklen_t` в *nix (беззнаковый).

Адрес переменной, содержащей:

перед вызовом — количество байт, которое можно записать в структуру-адрес;

после вызова — количество байт, записанных в структуру-адрес.

N. B.: вызовы блокирующие:

функции не вернут управление программе, пока не произойдет некоторое событие.

- `recvfrom()` выполняется, пока не будут получены данные по сети.
- `sendto()` выполняется, пока ОС не примет данные для отправки.
  - Ожидание только при загруженности системы.
  - Отправка может произойти не сразу или не произойти вообще (unreliable!).

Закрытие сокета

- Функция:
  - Windows: `closesocket(channel);`
  - *nix: `close(channel);`
- Сокеты — дефицитный ресурс ОС!
  - По исчерпании доступных сокетов становится невозможно создать новые.
  - Процессу выделяется несколько тысяч.

    (Сетевые программы могут работать непрерывно, долго и под большой нагрузкой.)

Настройка сокета

Значение опции: 1 — включить (разрешить), 0 — выключить (запретить).

```
int value = 1;

setsockopt(
    channel,
    SOL_SOCKET,
    SO_BROADCAST,
    (const char*)&value,
    sizeof(value));
```

Описание опций и возможных значений — в документации (MSDN или man setsockopt).

Опция задается для указанного сокета (не для сеанса, подсистемы и т. п.).

Указатель на переменную со значением опции и её размер.

Диагностика ошибок

- Можно получить код последней ошибки:
  - Windows:
    - `int error_code = WSAGetLastError();`
    - расшифровка кодов — в MSDN;
  - *nix:
    - переменная `errno`;
    - расшифровка кодов — `man errno`;
    - проверять нужно *сразу* после ошибки.
- Возвращаемое значение при ошибках:
  - Windows: `SOCKET_ERROR` (есть исключения)
  - *nix: всегда `-1`

Используемые библиотеки

- Windows:
  - `#include <winsock.h>`
  - компоновать с `ws2_32.lib`
    - Visual Studio:
      `#pragma comment(lib, "ws2_32")`
    - Code::Blocks:
      Project → Build options… → Linker
- *nix:

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

Запуск и останов библиотеки сетевой подсистемы (Windows)