

# Introdução ao *POSIX Socket API*

**Volnys Borges Bernal**

**Depto de Eng. de Sistemas Eletrônicos  
Escola Politécnica da USP**



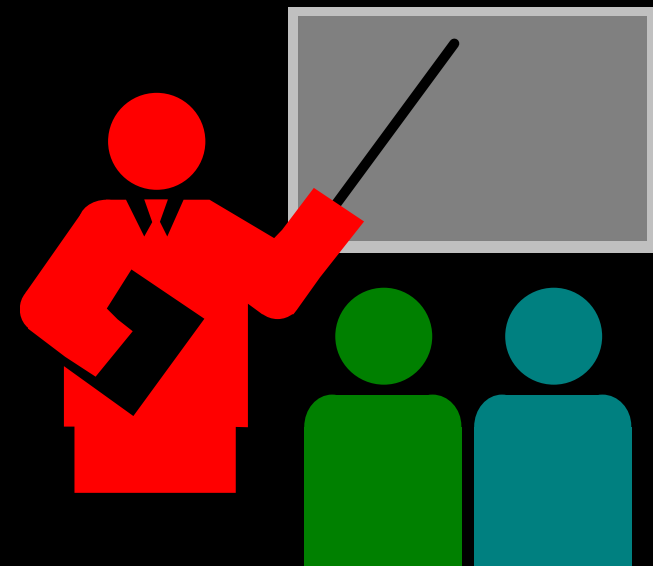


# Agenda

---

- ❑ **POSIX Socket API**
- ❑ **Socket Address**
  - ❖ Socket
  - ❖ Socket address
  - ❖ Par Socket
  - ❖ Estrutura sockaddr\_in
- ❑ **Utilização de sockets com a Pilha TCP/IP**
- ❑ **Estrutura de dados do sistema operacional**

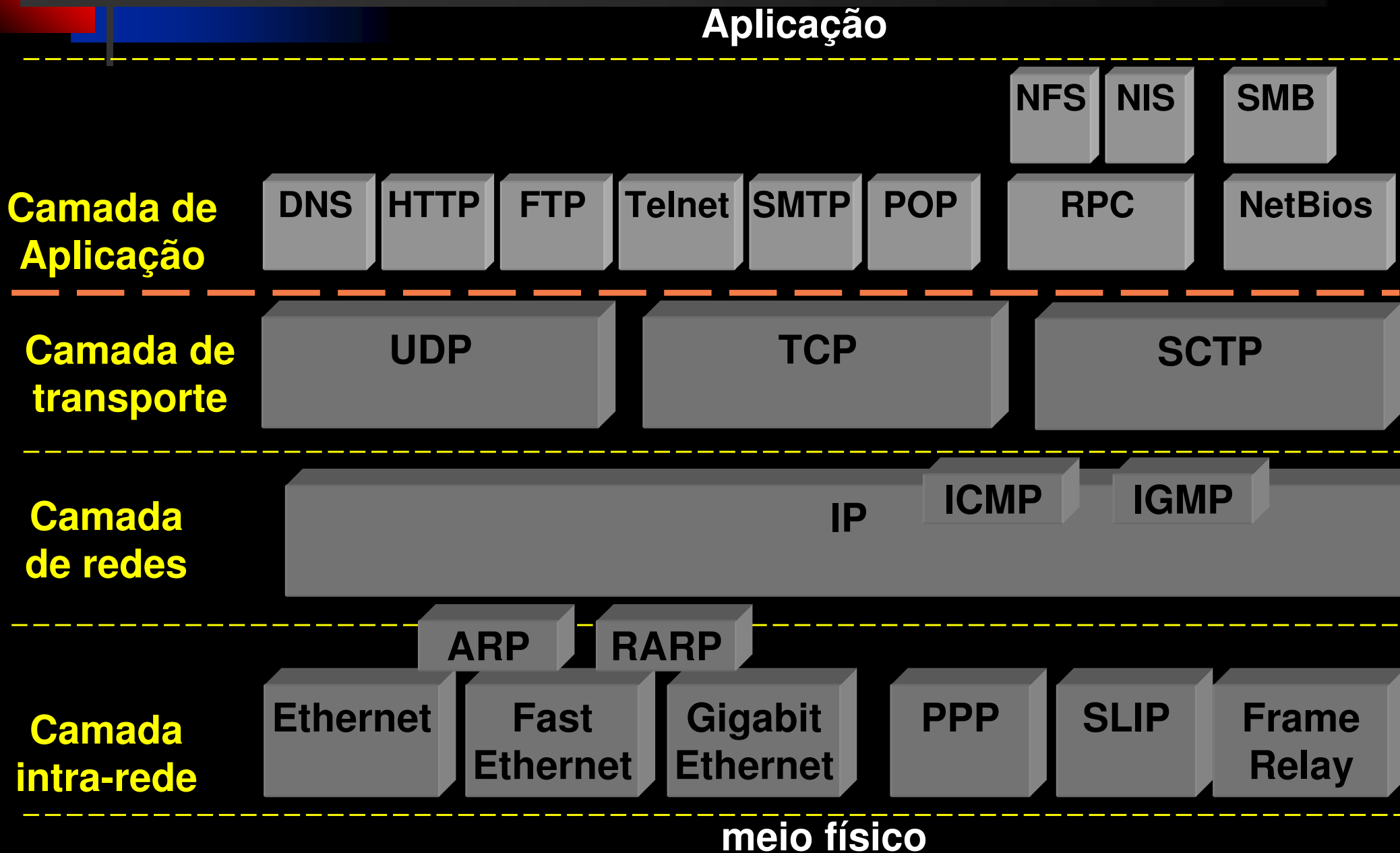
# POSIX Socket API



# POSIX Socket API

- ❑ **Originária do BSD Sockets**
- ❑ **Interface de programação mais usual para utilização dos serviços da pilha TCP/IP**
- ❑ **Permite à uma aplicação (processo) estabelecer:**
  - ❖ Sessão de comunicação UDP
  - ❖ Conexão TCP
  - ❖ Sessão de comunicação IP

# POSIX Socket API



# POSIX Socket API

Processos

Aplicação

POSIX Socket API

UDP

TCP

SCTP

Núcleo do  
Sistema  
Operacional

ICMP

IGMP

IP

Device Driver

ARP

Placa de Rede

Ethernet

PPP

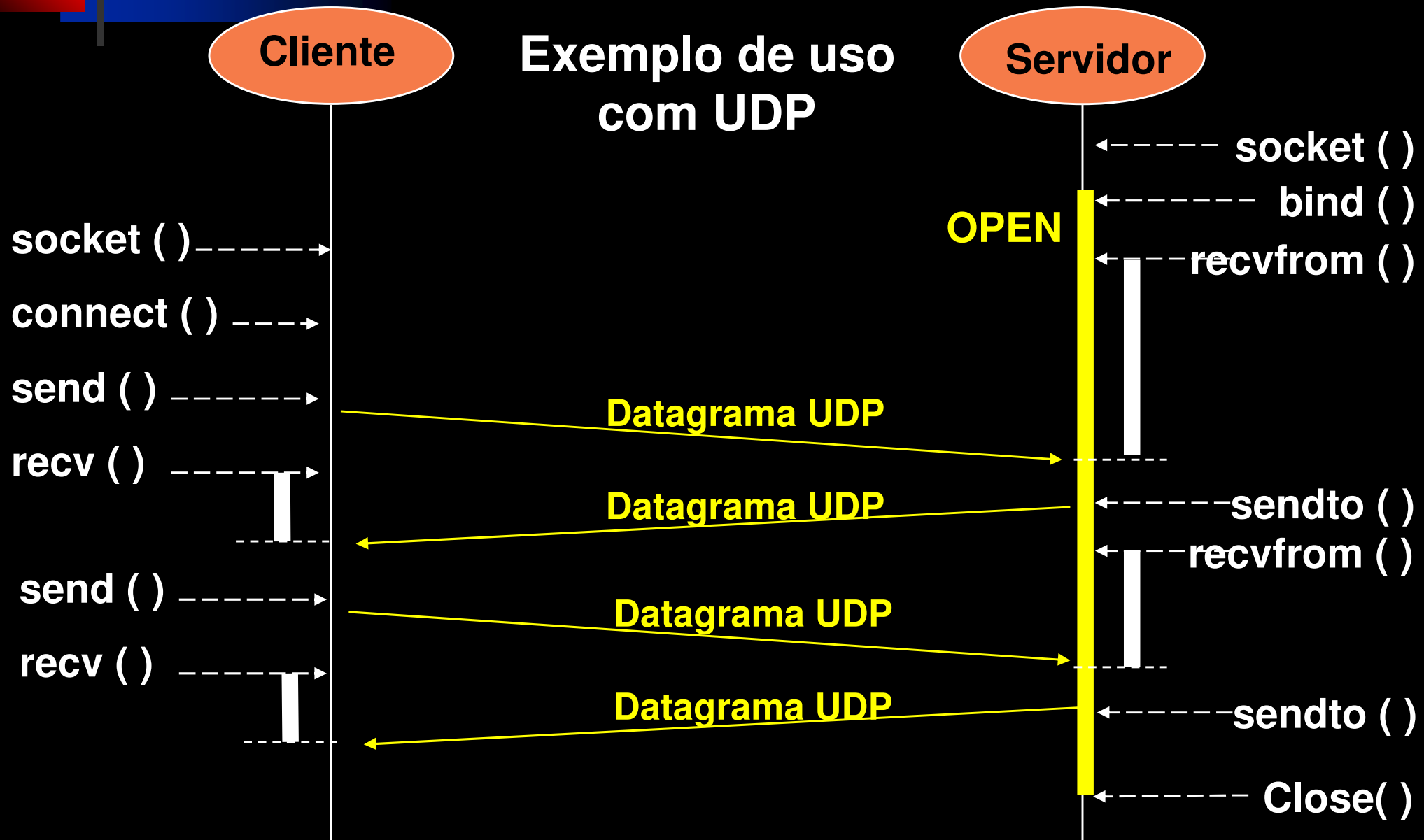
Meio Físico

# POSIX Socket API

## □ Principais funções da interface de programação sockets

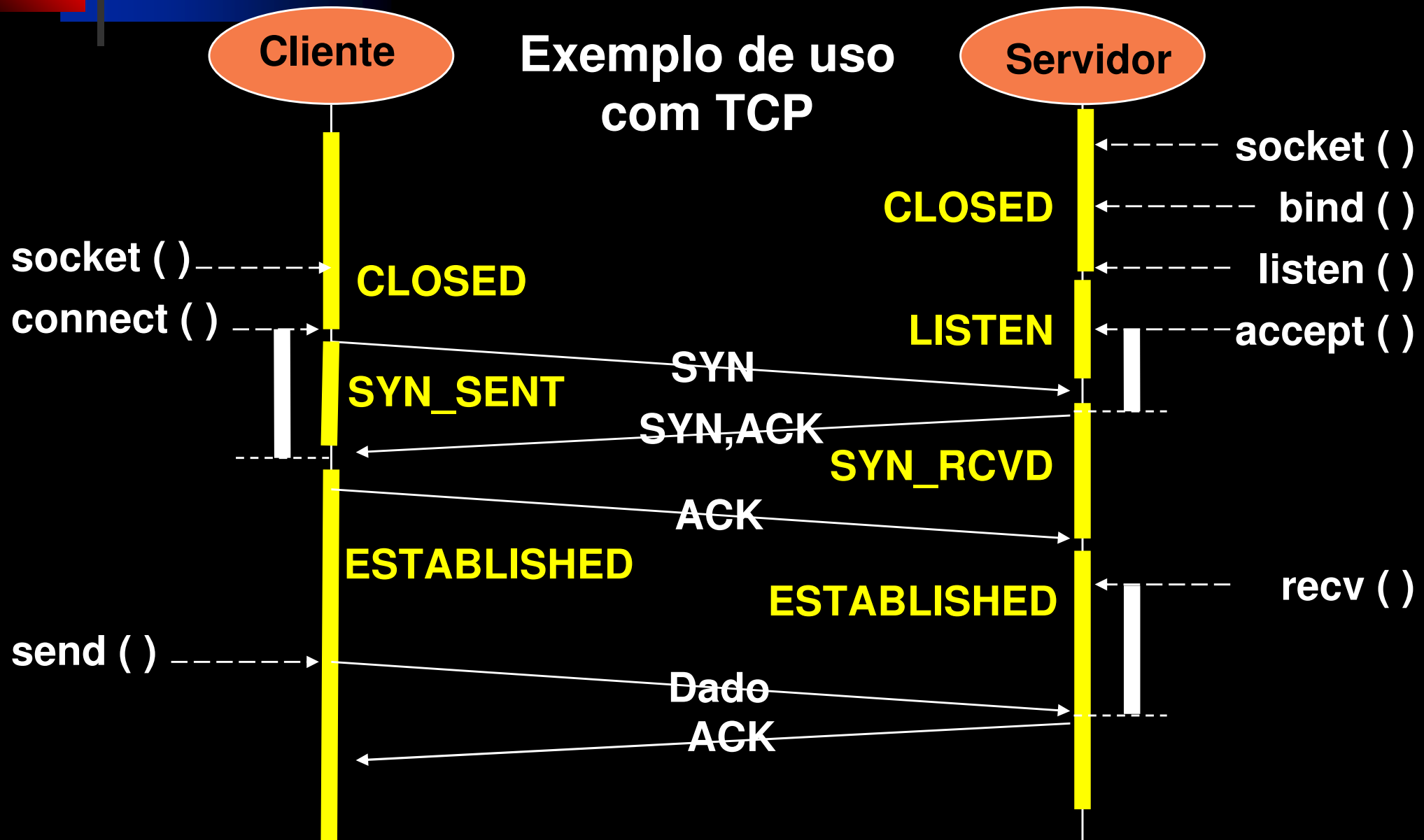
	UDP	TCP
socket	X	X
bind	X	X
listen		X
connect	X	X
accept		X
read	X	X
write	X	X
recv	X	X
send	X	X
recvfrom	X	X
sendto	X	X
close	X	X
shutdown	X	X

# POSIX Socket API

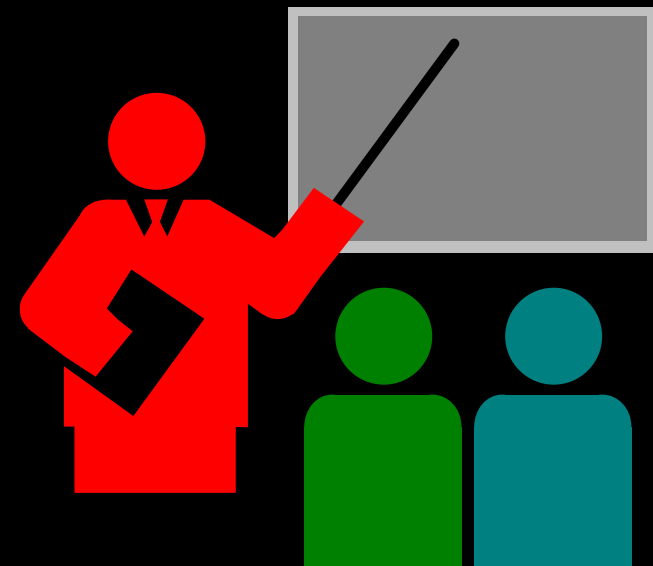




# POSIX Socket API



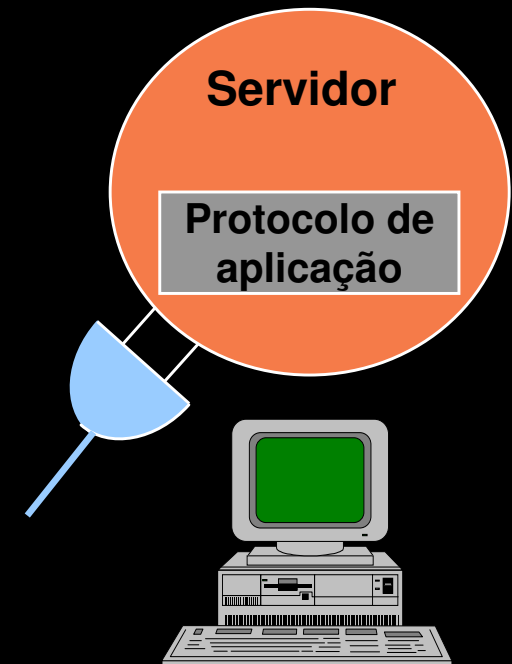
# Socket Address



# Socket address

## □ Socket

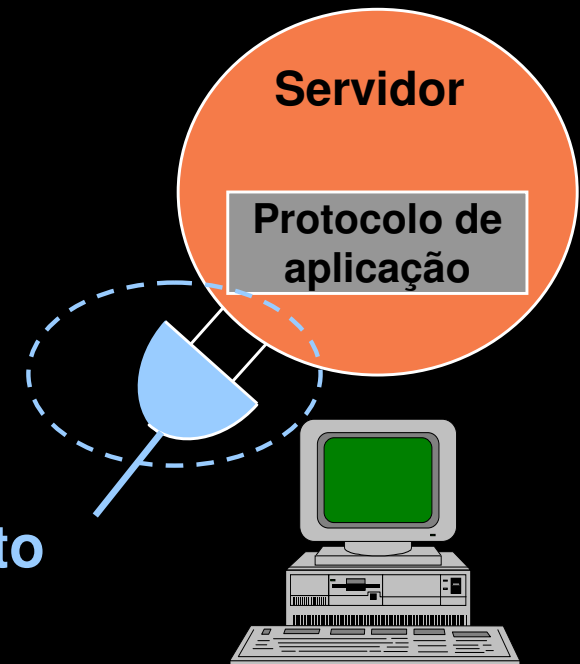
- ❖ Socket = Soquete = Plug
- ❖ Neste caso se refere a um “soquete de comunicação”
- ❖ Extremo da comunicação (*end point*) entre duas aplicações



# Socket address

## □ Socket address

- ❖ Endereço que permite identificar um soquete de comunicação
- ❖ Socket address para IPv4 contém:
  - Endereço IP
  - Porta do protocolo de transporte

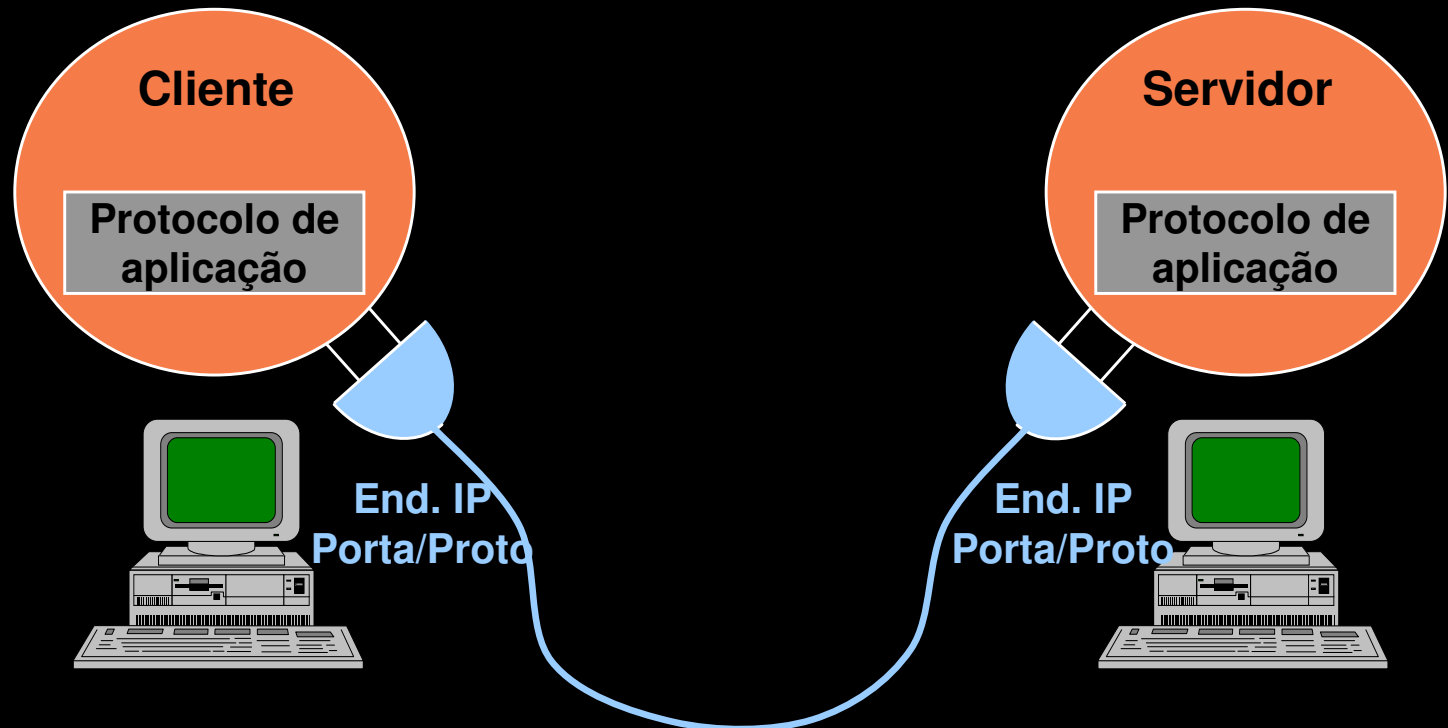


**Socket Address = End. IP + Porta/Proto**

# Socket address

## □ Par Socket

- ❖ Identifica uma sessão de comunicação
- ❖ Identificado por dois “Socket Address”



# Socket address

## □ Estrutura sockaddr

- ❖ Estrutura genérica para representar o endereço de um socket.

```
struct sockaddr
```

```
{  
    u_short sa_family;  
    char    sa_data[14];  
}
```

struct sockaddr

sa\_family



sa\_data



# Socket address

## □ Address Family

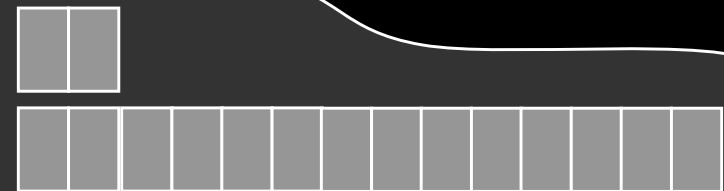
- ❖ Identifica o tipo de endereço utilizado
- ❖ Exemplo de identificadores para address family:
  - AF\_INET - para endereços IPv4
  - AF\_INET6 - para endereços IPv6

```
struct sockaddr  
{  
    u_short sa_family;  
    char    sa_data[14];  
}
```

struct sockaddr

sa\_family

sa\_data



# Socket address

## □ Estrutura sockaddr

- ❖ Apesar de existir uma estrutura genérica para representar o endereço de um socket, cada protocolo possui uma forma própria de representar o endereço de um socket.
- ❖ Por este motivo, existem definições de estruturas de dados específicas para cada protocolo que são mapeadas sobre a estrutura sockaddr.



# Socket address

## ❑ Estrutura sockaddr\_in

- ❖ Pode ser usada para endereço IPv4 (address family=AF\_INET)
- ❖ Endereço do soquete pode ser representado utilizando a estrutura “*socket address Internet*”.

### Formato geral

```
struct sockaddr
```

```
{  
    u_short sa_family;  
    char    sa_data[14];  
}
```

Pode ser  
mapeado  
sobre ...



### Formato específico para IPv4

```
struct sockaddr_in
```

```
{  
    u_short  sin_family;  
    u_short  sin_port;  
    u_long   sin_addr  
    char     sin_zero[8];  
}
```

# Socket address

## □ Definição de `sockaddr_in` nos sistemas Linux atuais:

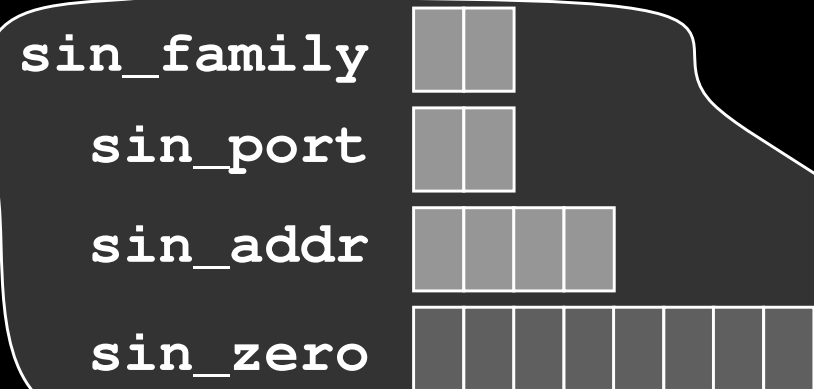
```
struct sockaddr_in
{
    sa_family_t    sin_family; // address family: AF_INET (IPv4)
    in_port_t      sin_port;   // porta em network byte order
    struct in_addr sin_addr;    // endereço IP
};

struct in_addr
{
    uint32_t       s_addr;     // endereço IP em network byte order
};
```

# Socket address

**Estrutura customizada para  
socket address internet**

**struct sockaddr\_in**



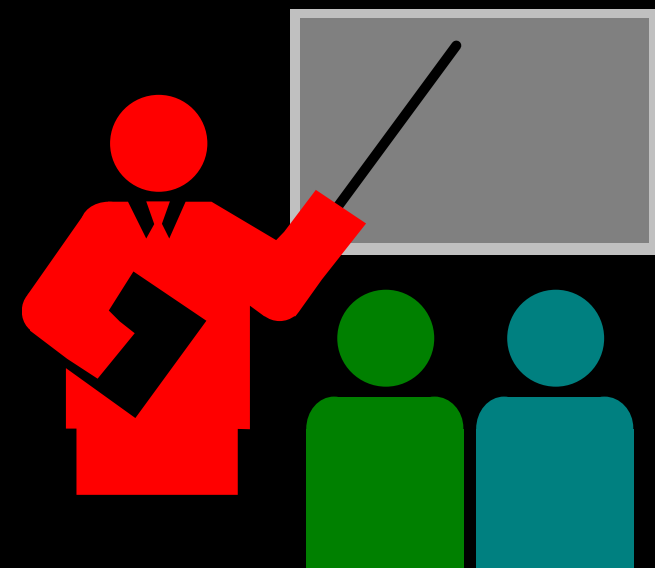
**Estrutura geral  
socket address**

**struct sockaddr**



Mapeada em ...

# Utilização de sockets com a Pilha TCP/IP



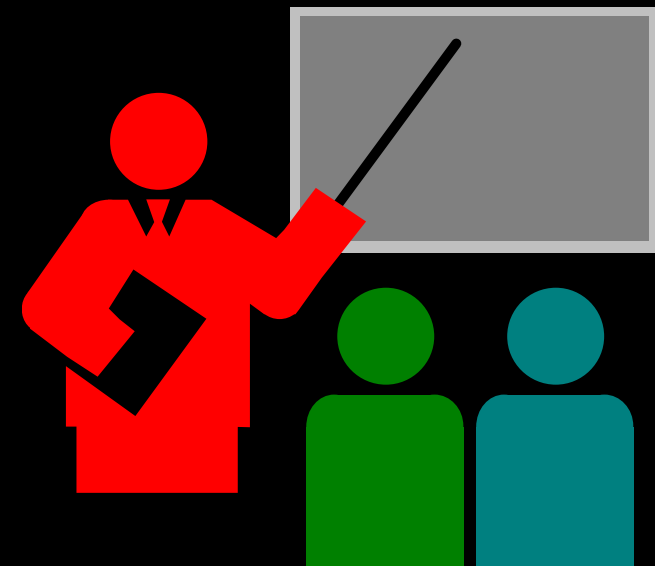
# Sockets com a Pilha TCP/IP

- ❑ Quando a chamada `socket()` é ativada para uma comunicação sobre a pilha TCP/IP, o sistema operacional aloca uma estrutura `socket`, contendo inicialmente somente informações a respeito de:
  - ❖ Família de protocolo
  - ❖ Tipo de serviço
  - ❖ Protocolo de interação (IP, UDP, TCP, ...)
  
- ❑ Existem diversas pilhas de protocolos de rede (além da pilha TCP/IP), cada uma com diferentes protocolos e formatos de endereço.
  
- ❑ Para cada uma destas, existem diferentes valores que podem ser utilizados para identificar a Família de protocolos, o tipo de fluxo e o protocolo de interação

# Sockets com a Pilha TCP/IP

- No caso da utilização da pilha TCP/IP com IPv4 os valores possíveis são:
  - ❖ **Protocol Family** (PF)
    - Possibilita identificar o tipo de protocolo utilizado
    - Deve ser utilizado PF\_INET
  - ❖ **Address Family** (AF)
    - Possibilita identificar o tipo de endereço utilizado
    - Deve ser utilizado AF\_INET
  - ❖ **Protocol**
    - Protocolo de interação
    - Valores usuais:
      - Camada de transporte: TCP, UDP, STCP
      - Camada de redes: IP
  - ❖ **Type**
    - Tipo de serviço oferecido
    - Valores possíveis: SOCK\_STREAM, SOCK\_DGRAM ou SOCK\_RAW

# Estruturas de dados do sistema operacional

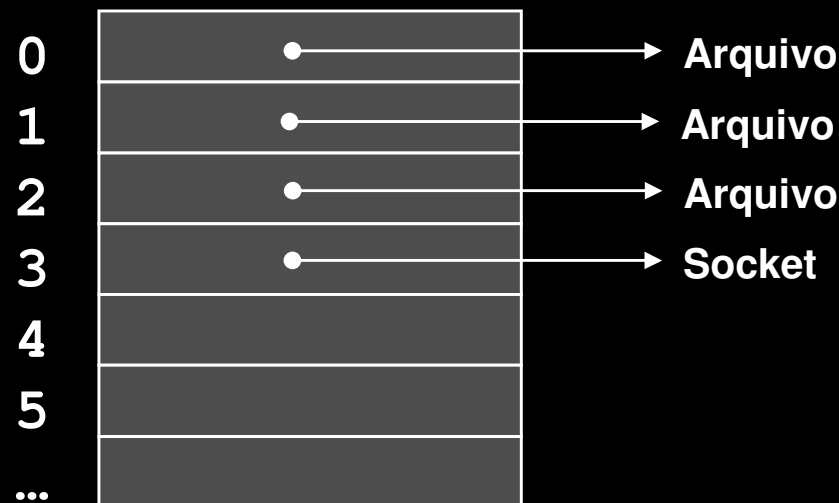


# Estruturas de dados do sistema operacional

## □ Tabela de descritores de arquivos abertos

- ❖ Ao abrir um arquivo ou um socket, é alocada uma entrada da tabela de descritores de arquivos abertos.
- ❖ Cada entrada da tabela de descritores contém um índice para outra tabela que contém informações a respeito do arquivo ou socket
- ❖ O índice da tabela (chamado de descritor) é utilizado pela aplicação para identificar o arquivo ou socket em operações futuras

### Tabela de descritores de arquivos abertos





# Estruturas de dados do sistema operacional

## □ Estrutura socket:

- ❖ Para cada sessão de comunicação UDP, TCP, SCTP ou IP é necessário alocar uma estrutura socket
- ❖ Esta e outras estruturas internas ao sistema operacional mantém informações a respeito da sessão de comunicação (TCP ou UDP) completa ou embrionária

protocol family

service

local

socket address

remote

socket address

...

...

*Estrutura interna do SO (socket)*

PF\_INET

SOCK\_STREAM

*Observação:*

- *Visão simplificada da estrutura.*
- *O programador nunca manipula a estrutura socket diretamente.*

# Estruturas de dados do sistema operacional

## □ Estrutura socket

- ❖ Os valores da estrutura socket interna ao sistema operacional são complementados a cada chamada da POSIX socket API.

# Estruturas de dados do sistema operacional

## □ Estruturas internas ao sistema operacional

*Observação:*

- *Visão simplificada*
- *O programador não manipula estas estruturas diretamente.*

Tabela de descritores

0	
1	
2	
3	•
4	
5	
...	

protocol family  
service  
local  
socket address  
remote  
socket address

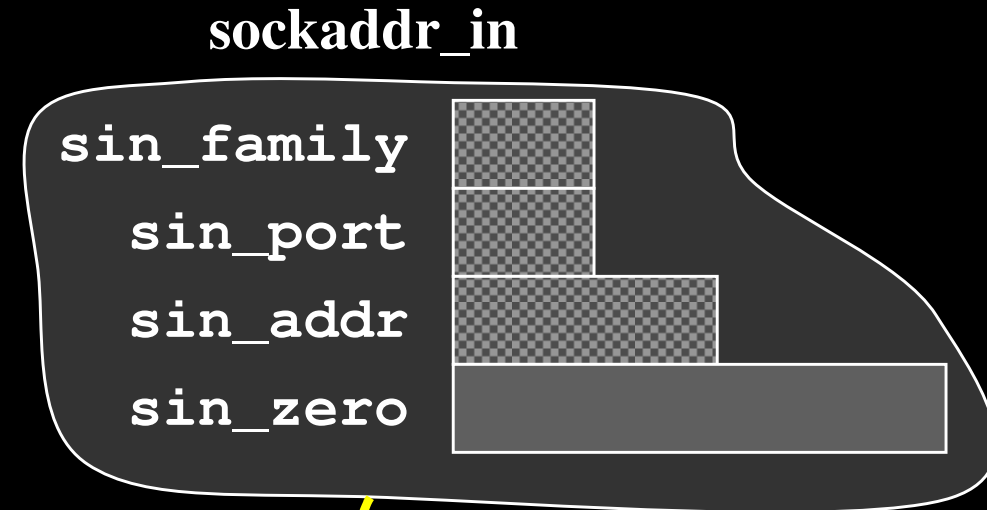
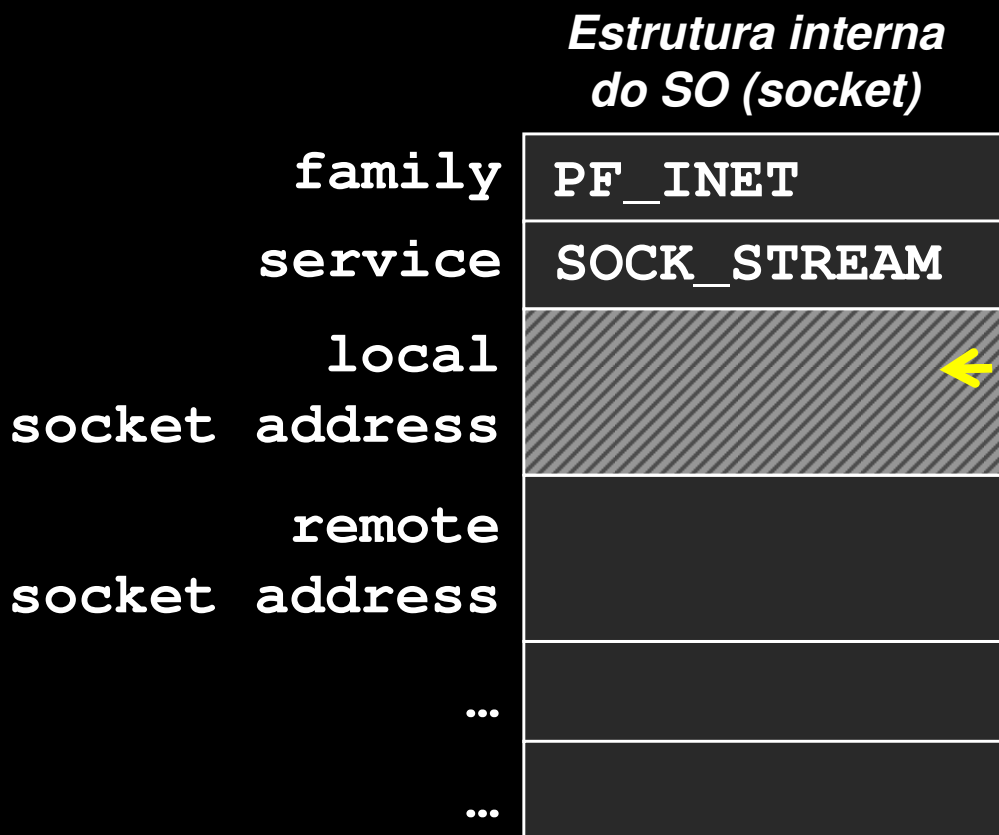
Estrutura interna do SO (socket)

PF_INET
SOCK_STREAM
...
...

Socket Descriptor

# Estruturas de dados do sistema operacional

- ❑ Lado servidor com protocolo IPv4 (Internet)

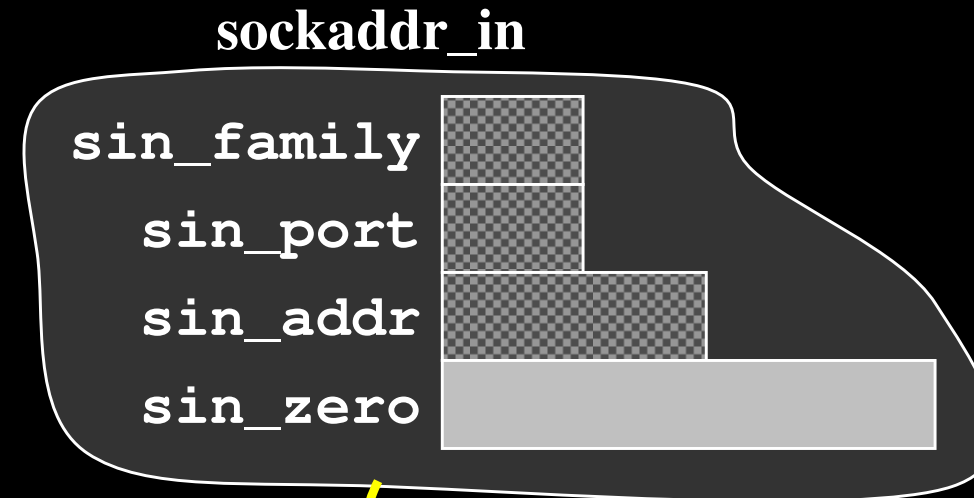
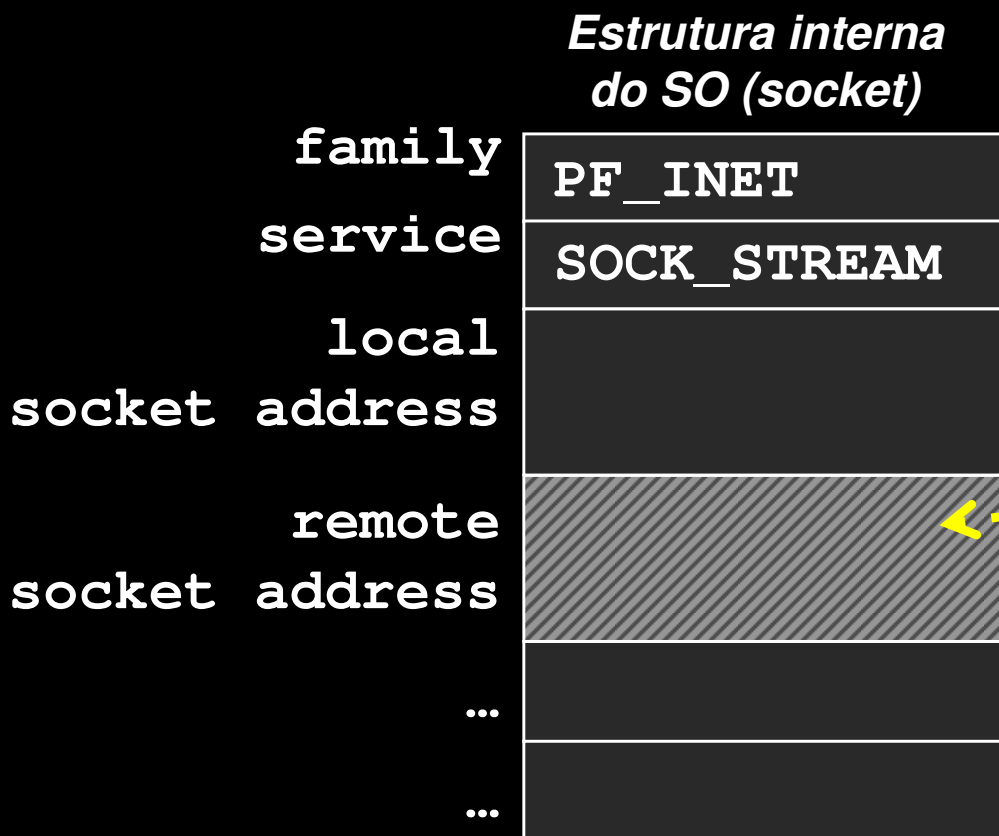


**bind ( )**

*Utilizado no lado servidor.  
Define o endereço e a porta local para o socket*

# Estruturas de dados do sistema operacional

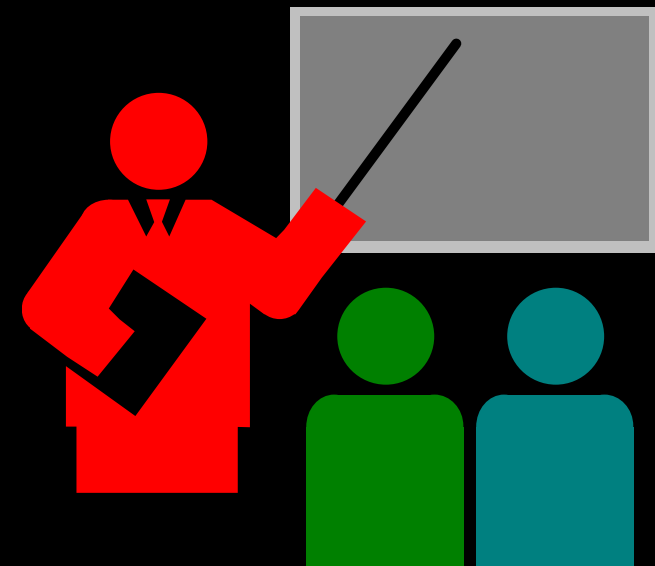
- ❑ Lado cliente com protocolo IPv4 (Internet)



**Connect ( )**

*Utilizado no lado cliente. Define o parceiro remoto de comunicação*

# Referências Bibliográficas



# Referências Bibliográficas

- **COMMER, DOUGLAS; STEVENS, DAVID**
  - ❖ Internetworking with TCP/IP: volume 3: client-server programming and applications
  - ❖ Prentice Hall
  - ❖ 1993