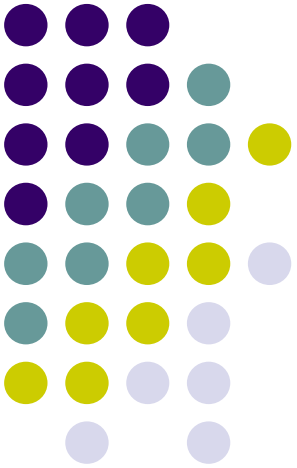
Sistemas Distribuídos

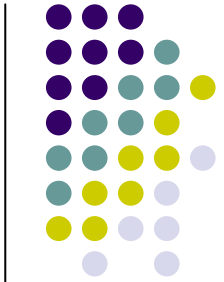
Universidade Federal do ABC

Turma:
Ciência da Computação

Prof. Dr. Francisco **Isidro** Massetto

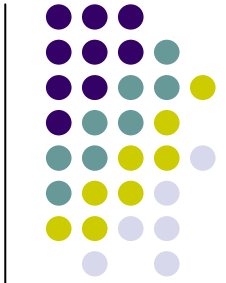


Introdução: Sockets



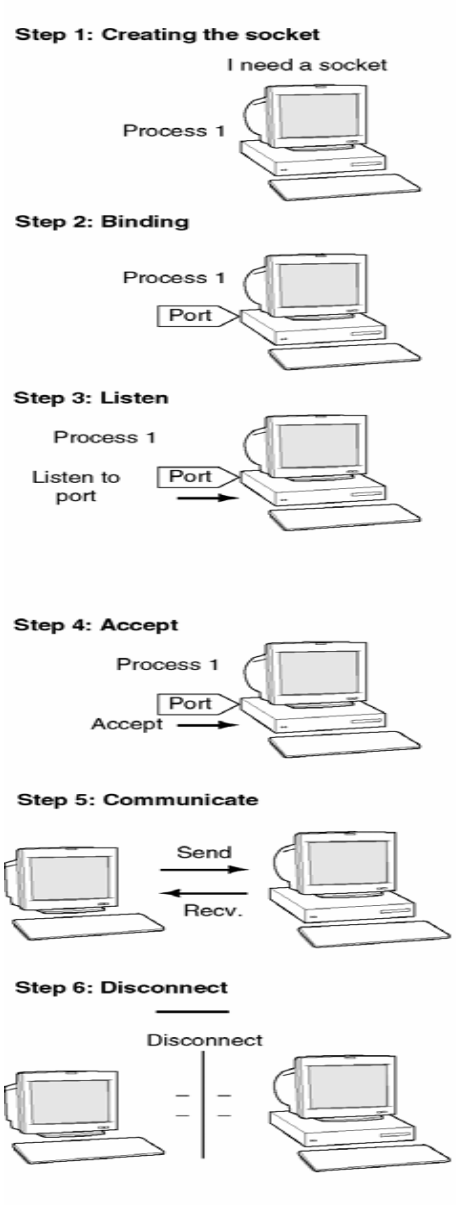
- Para estabelecer a Comunicação Interprocesso nos Sistemas Distribuídos, para permitir que processos se comuniquem na troca de dados ou acessos a recursos ou serviços em processadores remotos, se faz necessário o uso de um mecanismo de serviços de transporte;
- Um dos mecanismos mais utilizado é o Socket;
- Sockets é a maneira mais popular de utilizar as funcionalidades de comunicação TCP/IP;
- Todos os mecanismos Sockets são gerenciados pela camada de transporte;
- Existem diversas APIs Sockets (Application Program Interface) e as mais populares são do ambiente Unix, bem como a WinSock do Windows.

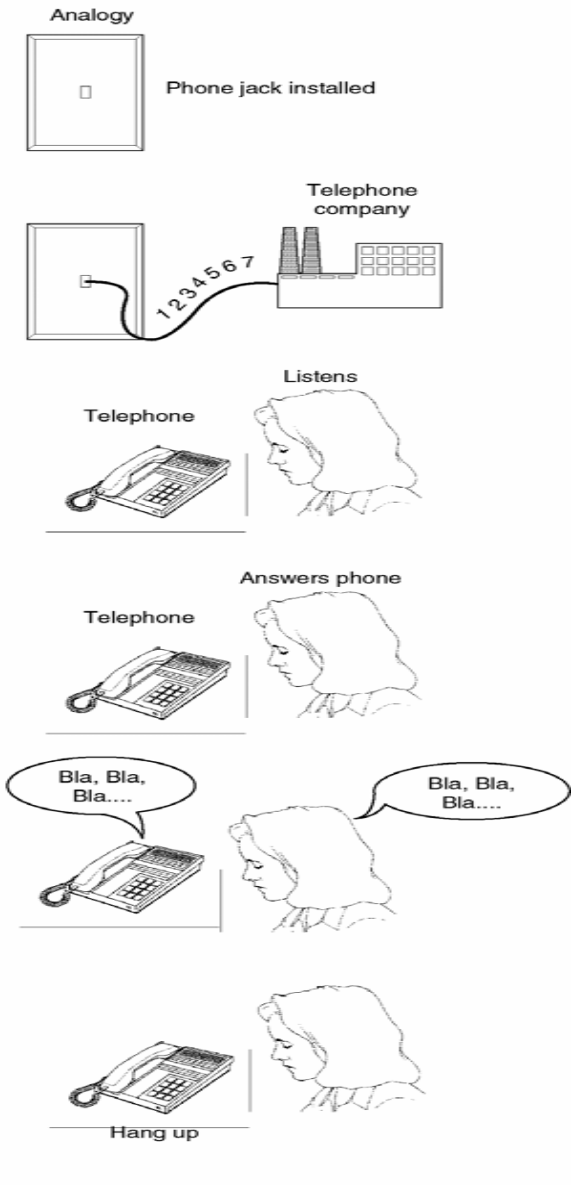
Socket: Definição



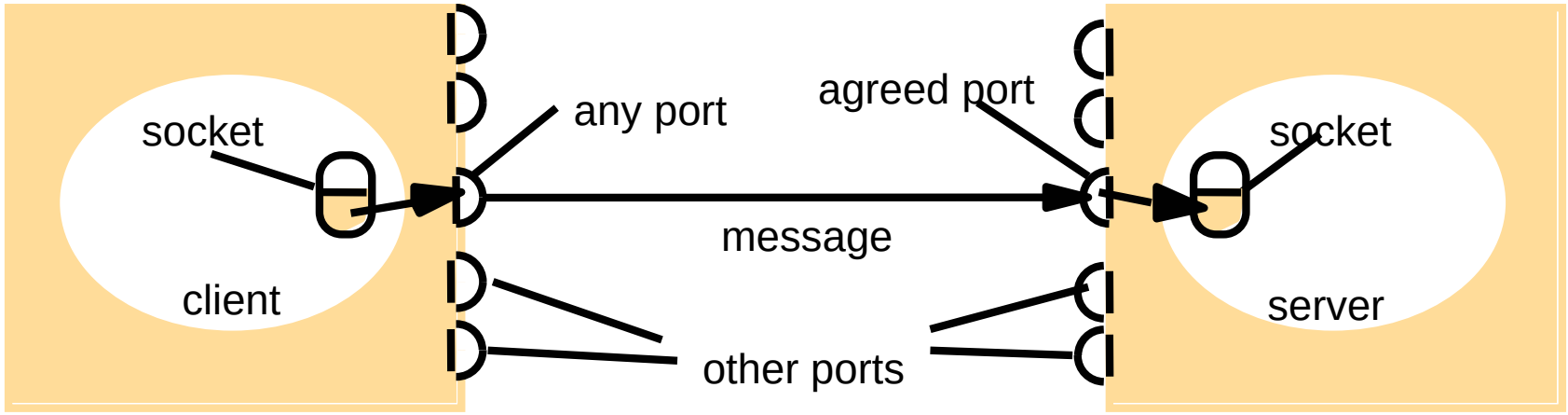
- Um Socket é um ponto final (endpoint) de um canal bidirecional de comunicação entre dois programas rodando em uma rede;
- Cada Socket tem os seguintes endereços de endpoint:
  - Endereço local (número da porta) que refere-se ao endereço da porta de comunicação para camada de transporte;
  - Endereço global (nome host) que refere-se ao endereço do computador (host) na rede.

Socket: Uma analogia

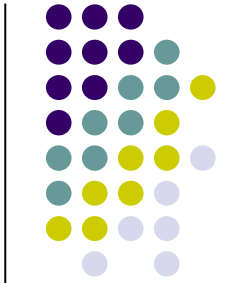




Sockets e Portas

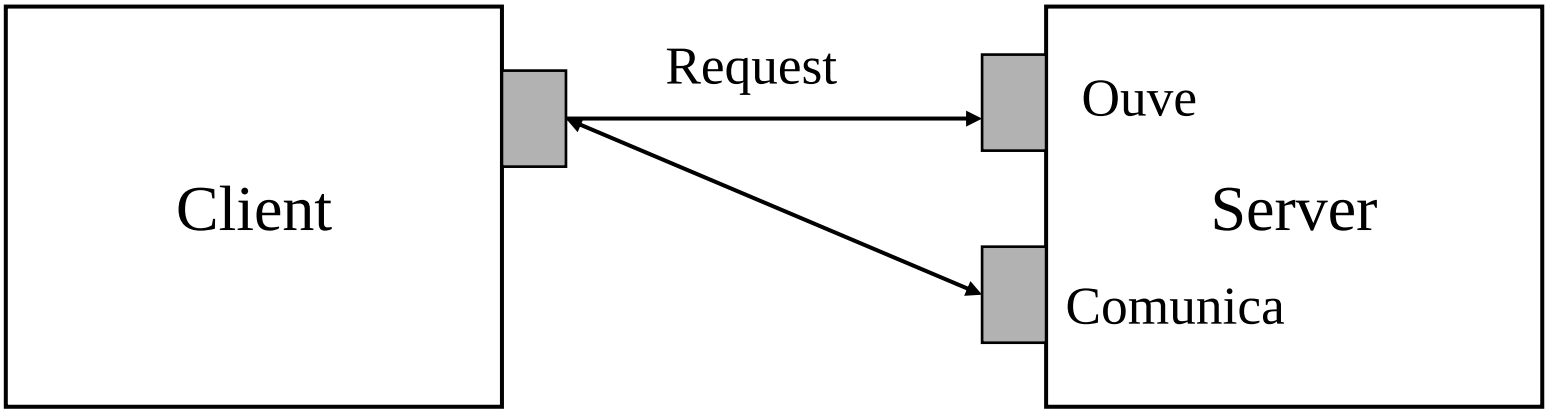


Socket: Conexão



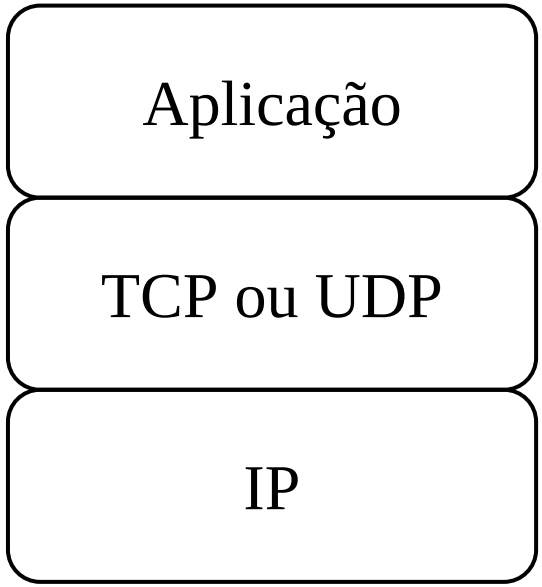
- O servidor apenas fica “ouvindo” o Socket aguardando um pedido de conexão do cliente;
- O cliente sabe o nome do host e qual porta está associada à aplicação servidora;
- Assim que o servidor aceitar a conexão, este cria um novo Socket (e conseqüentemente o associa a uma nova porta) e pode ficar esperando novas conexões no Socket original enquanto atende às requisições do cliente pelo novo Socket.

Socket: Conexão

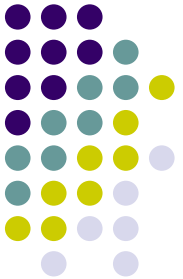


Socket: Protocolos TCP e UDP

- Protocolos de Transporte TCP e UDP;
- Ambos utilizam a camada IP como camada de Rede.



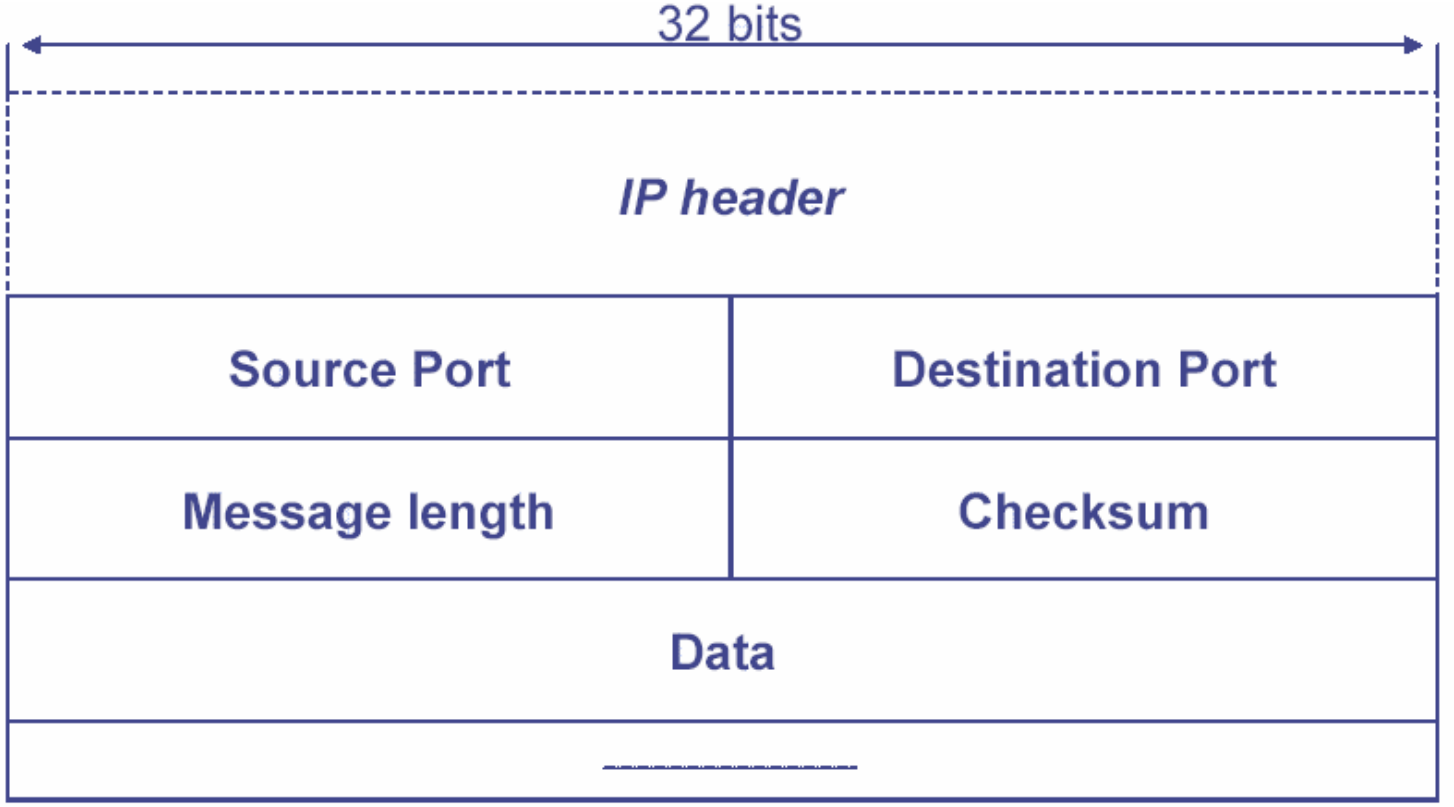
Socket: Protocolos UDP



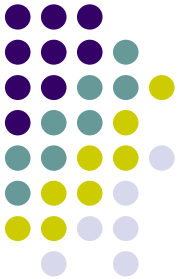
- User Datagram Protocol (UDP):
- Protocolo não orientado à conexão;
- Não há garantia de entrega dos dados (não há mensagens de confirmação);
- Perdas durante as transmissões não são tratadas por este protocolo;
- Usado em redes com alta confiabilidade, onde as taxas de perda são baixas;
- Exemplos:
  - TFTP, BOOTP, ...

Socket: Protocolos UDP

- Header UDP:



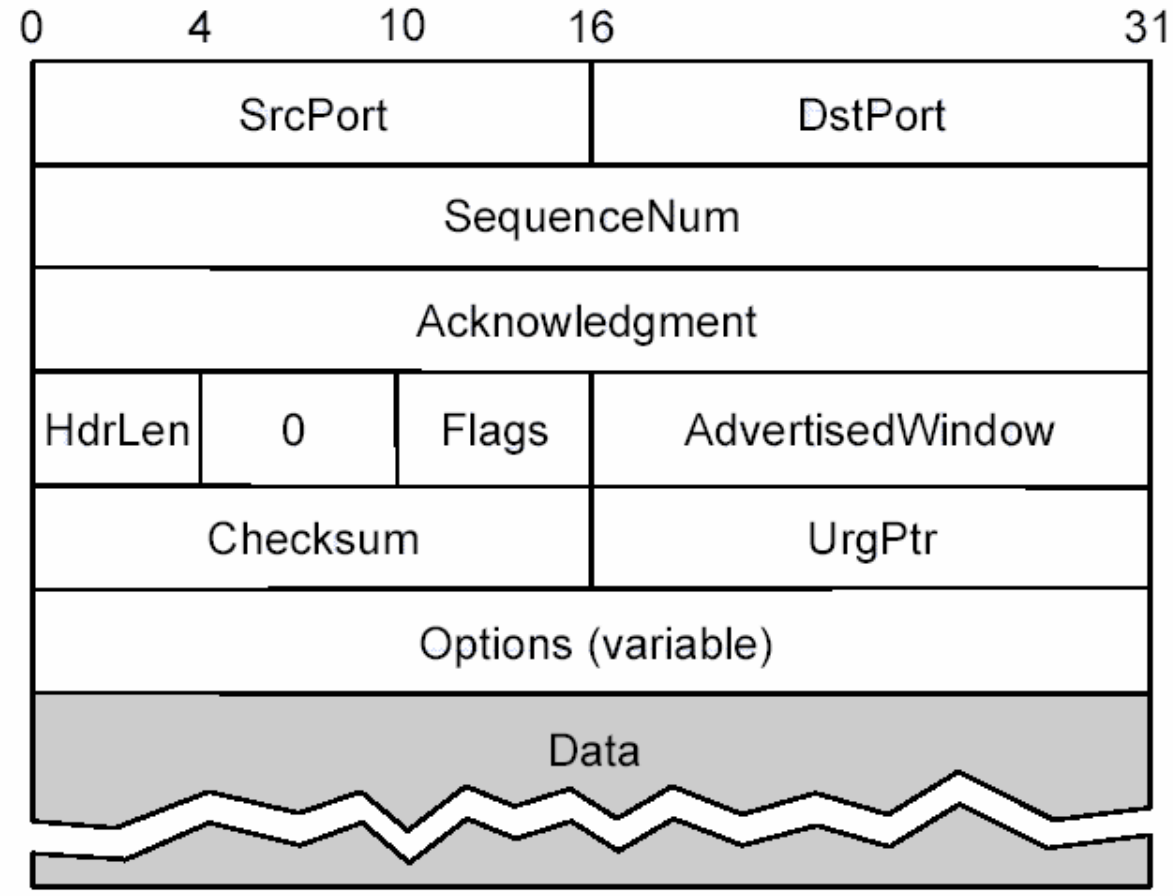
Socket: Protocolos TCP



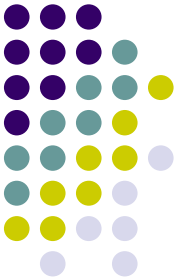
- Transmission Control Protocol (TCP):
- Protocolo orientado à conexão;
- Para haver a transmissão dos dados, uma fase de conexão entre as duas entidades que se comunicam precisa ser feita;
- Fase de Conexão $\rightarrow$ Fase de Transmissão dos Dados $\rightarrow$ Fase de Desconexão;
- Exemplos:
  - TELNET, Web Browser, ...

Socket: Protocolos TCP

- Header TCP:

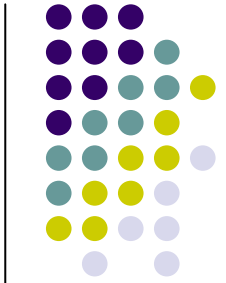


Socket: Comunicação C/S



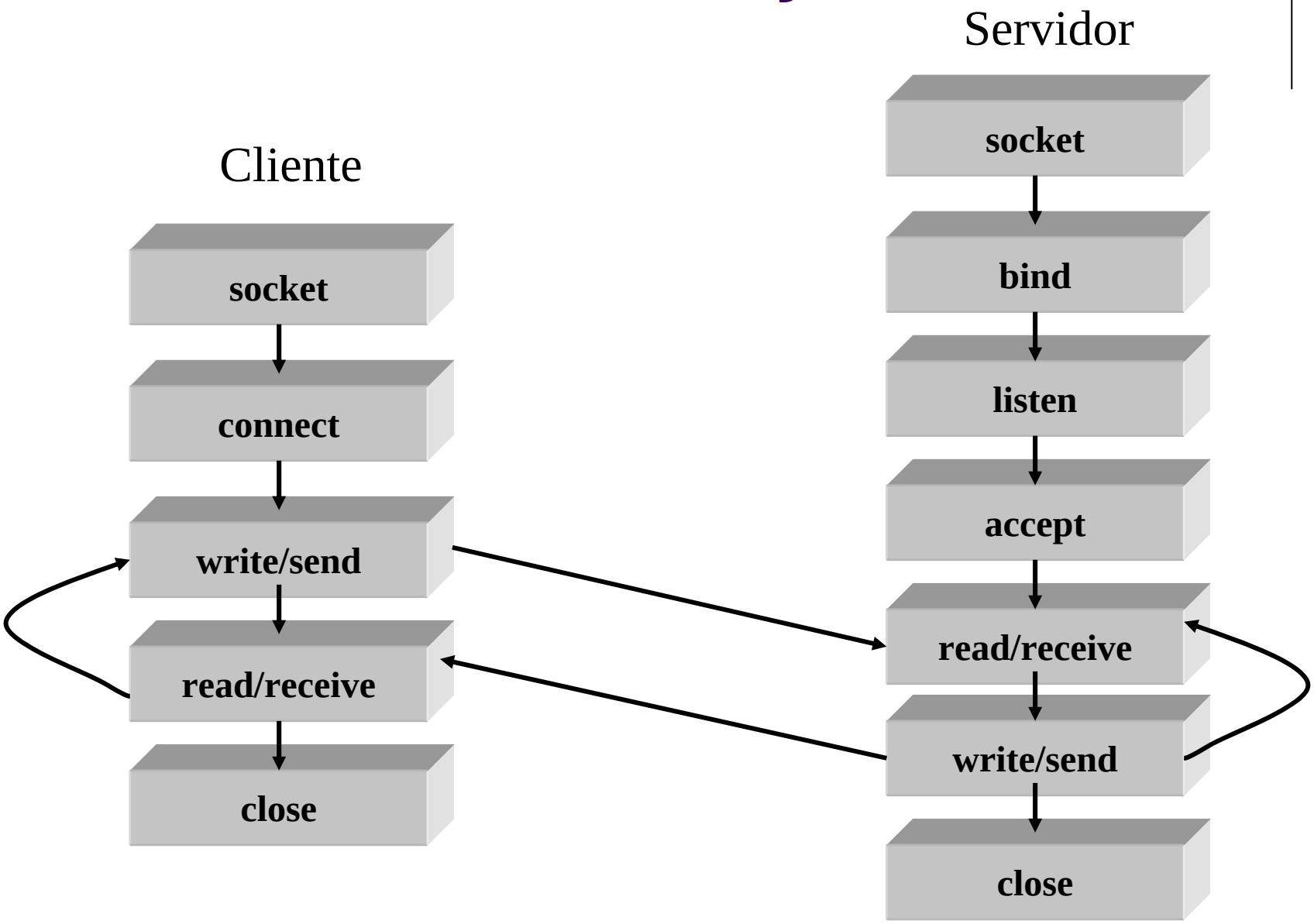
- Servidor:
  - Efetua a criação de um Socket;
  - Associa o Socket a um endereço local;
  - Aguarda por conexões da parte cliente;
  - Aceita conexões;
  - Lê requisições;
  - Opcionalmente envia resposta;
  - Fecha o Socket.

Socket: Comunicação C/S

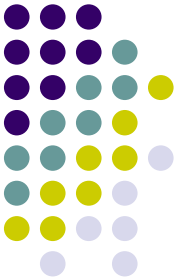


- Cliente:
  - Efetua a criação do Socket;
  - Estabelece a conexão;
  - Envia a requisição;
  - Opcionalmente aguarda resposta;
  - Fecha o Socket.

Socket: Comunicação C/S

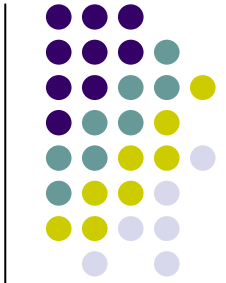


APIs Sockets: Socket



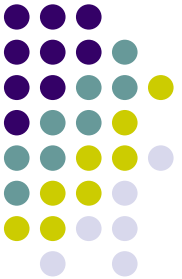
- Socket:
  - Cria um Socket e retorna um descritor;
  - O descritor é a referência para que as outras funções utilizem o Socket criado.
- result = socket(pf, type, protocol)
  - pf: Família de protocolos a ser usada com Socket .
    - PF_INET, PF_APPLETALK
  - type: Tipo de conexão a ser utilizada.
    - SOCK_STREAM, SOCK_DGRAM
  - protocol: Tipo de protocolo a ser utilizado.
    Requer grande conhecimento dos serviços oferecidos pelo protocolo.

APIs Sockets: Bind



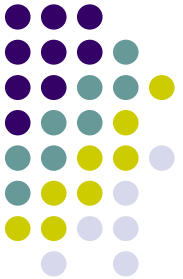
- Bind:
  - Prove o número da porta que servidor espera contato;
  - Função utilizada apenas pelo servidor, uma vez que associa um determinado endereço IP e porta TCP ou UDP para o processo servidor.
- bind(socket, localaddr, addrlen)
  - socket: Socket associado para ser registrado;
  - localaddr: Endereço local para vincular o Socket;
  - addrlen: Valor inteiro que determina o tamanho do endereço em bytes.

APIs Sockets: Listen



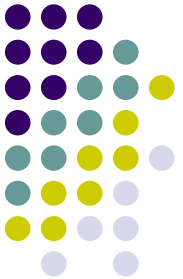
- Listen:
  - Indica ao sistema operacional para colocar o Socket em modo de espera (passivo) para aguardar conexões de clientes.
- listen(socket, queue)
  - socket: Socket que ficará em modo passivo aguardando por conexões;
  - queue: Tamanho máximo da fila de conexões que serão aceitas pelo Socket.

APIs Sockets: Accept



- Accept:
  - Cria um novo Socket a partir do estabelecimento de uma conexão para iniciar a comunicação (leitura e escrita).
- newsock = accept(socket, addr, addrlen)

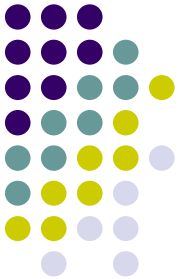
APIs Sockets: Read e Write



- Read:
  - Lê o conteúdo do buffer associado ao Socket.
- read(socket, buffer, lenght)

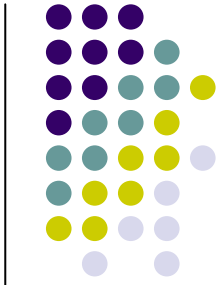
- Write:
  - Escreve dados em um buffer associado ao Socket.
- write(socket, buffer, lenght)

APIs Sockets: Close



- Close:
  - Informa ao sistema operacional para terminar o uso de um Socket.
- close(socket)

APIs Sockets: Outras Primitivas



- gethostbyname(host)
  - Extrai o endereço IP a partir do nome de um Host.
- getprotobyname(protocol)
  - Extrai o código correspondente ao protocolo a partir de uma string que o define.
- htons(addr)
  - Converte um endereço para o padrão de rede (Big endian).