

COMUNICAÇÃO ENTRE PROCESSOS

Sockets

Sistemas Distribuídos

Sockets - Conceitos Básicos

- Sockets são uma forma de IPC (InterProcess Communication) fornecida pela 4.3 BSD que fornecem comunicação entre processos residentes em sistema único ou processos residentes em sistemas remotos.

Sistemas Distribuídos

Conceitos Básicos

- Sockets criados por diferentes programas usam nomes para se referenciar
- Esses nomes geralmente devem ser traduzidos em endereços para uso
- Um endereço é especificado por um domínio

Sistemas Distribuídos

Tipos de Sockets

- STREAM SOCKET - Provê sequenciamento e fluxo bidirecional. Este tipo de socket transmite dados sobre uma base confiável e com capacidade de transmissão de dados expressos.
- No domínio UNIX, o SOCKET_STREAM trabalha igual a um pipe. No domínio INTERNET este tipo de socket é implementado sobre TCP/IP.

Sistemas Distribuídos

Tipos de Sockets

- SOCK_DGRAM - Suporta fluxo de dados bidirecional mas não oferece um serviço confiável como STREAM_SOCKET.
- Mensagens duplicadas, perdidas, e em ordem diferente (não sequenciadas) são problemas que podem aparecer neste tipo de socket.

Sistemas Distribuídos

Tipos de Sockets

- RAW_SOCKET - permite o acesso a interface de protocolos de rede. Disponível para usuários avançados e que possuam autoridade de usuário root
- Permite que uma aplicação acesse diretamente protocolos de comunicação de baixo nível.
- Permite a construção de novos protocolos sobre os protocolos de baixo nível já existentes
- Normalmente orientados a datagrama

Sistemas Distribuídos

Domínios e Protocolos

- O espaço no qual o endereço é especificado é chamado de domínio
- Domínios básicos:
 - INTERNET - AF_INET - os endereços consistem do end. de rede da máquina e da identificação do no. da porta, o que permite a comunicação entre processos de sistemas diferentes
 - Unix: AF_UNIX - os processos se comunicam referenciando um pathname, dentro do espaço de nomes do sistema de arquivos

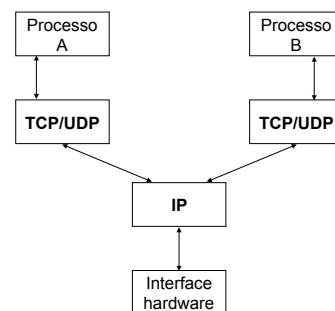
Domínios e Protocolos

- Domínio Internet
 - Implementação Unix do protocolo TCP/UDP/IP
 - Consiste de:
 - end. de rede da máquina
 - identificação do no. da porta
 - Permite a comunicação entre máquinas diferentes
 - Conexões sob a forma de sockets do tipo stream e do tipo datagramas

Tipos de Sockets

- Socket Stream
 - Conexões confiáveis
- Datagramas
 - Não fornecem segurança

Protocolos TCP/IP

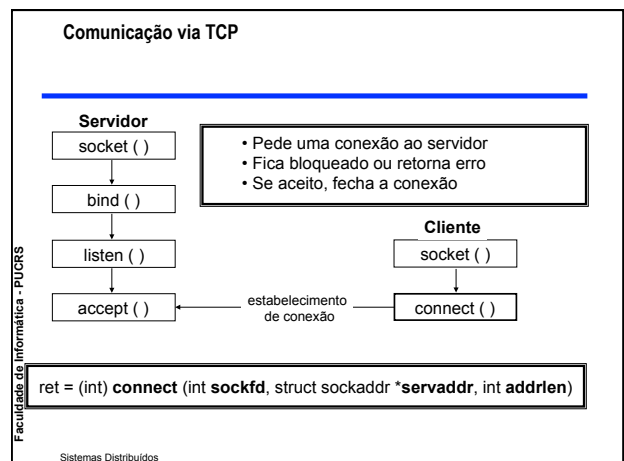
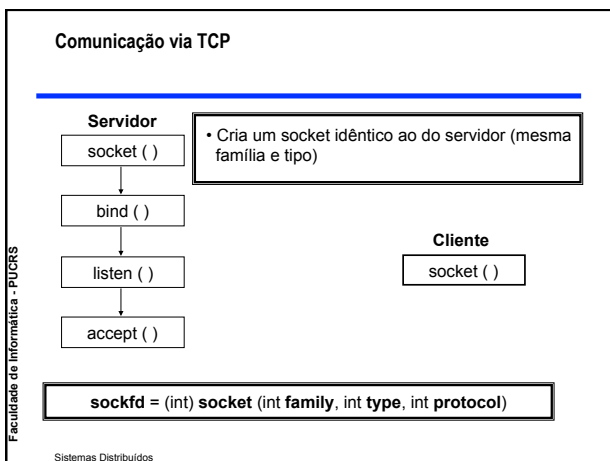
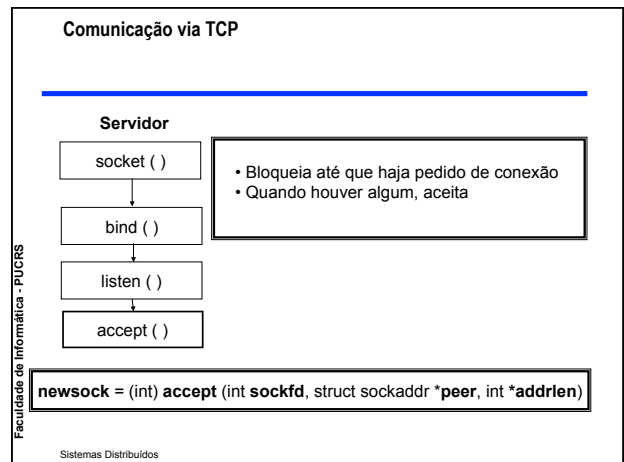
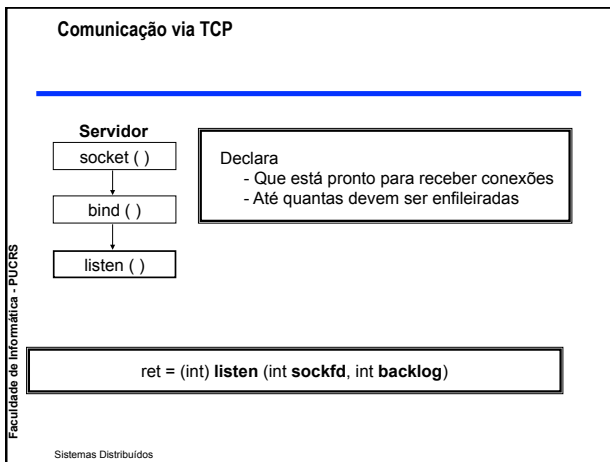
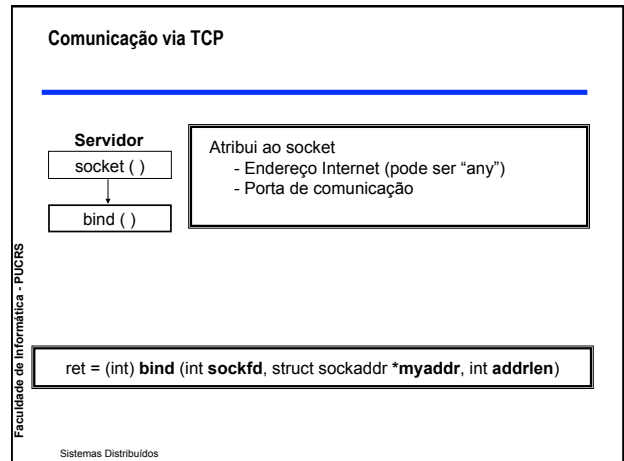
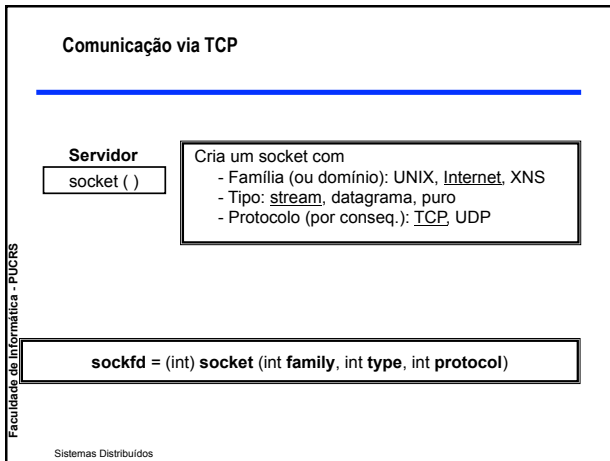


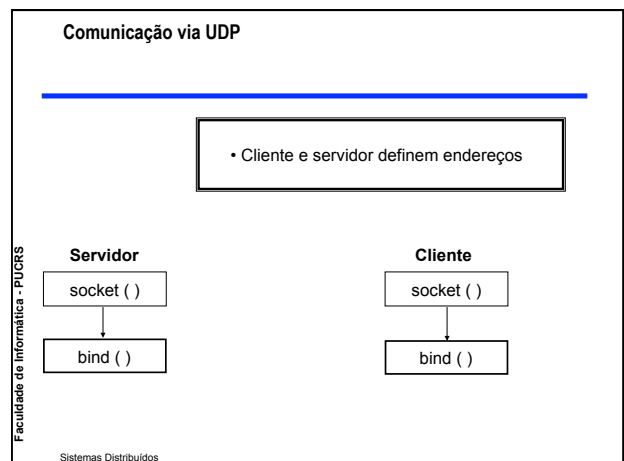
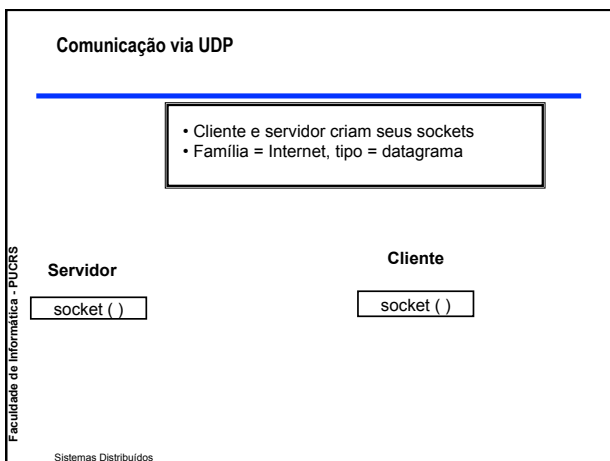
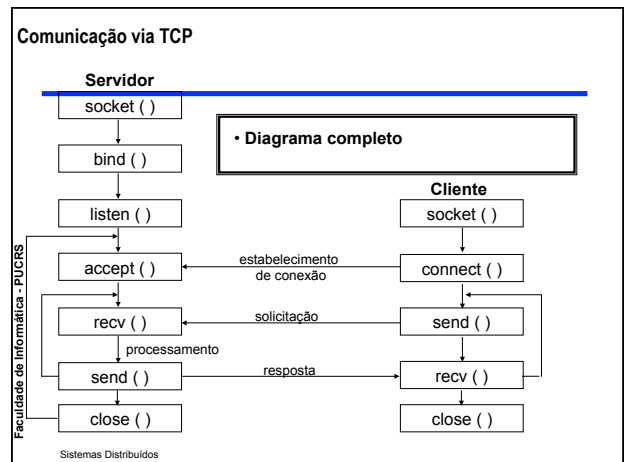
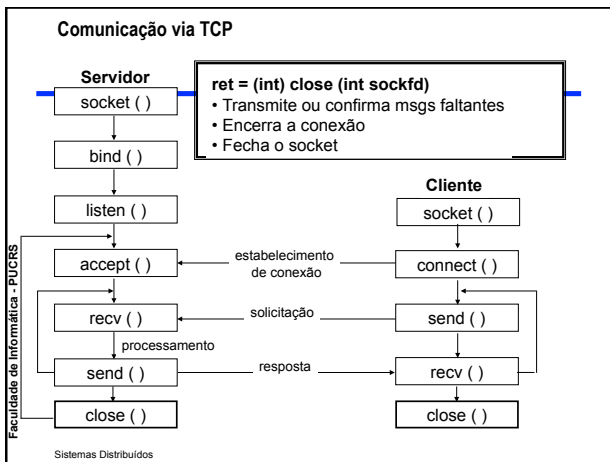
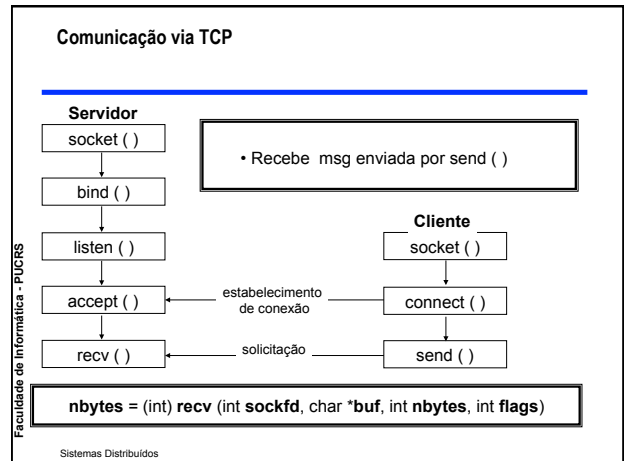
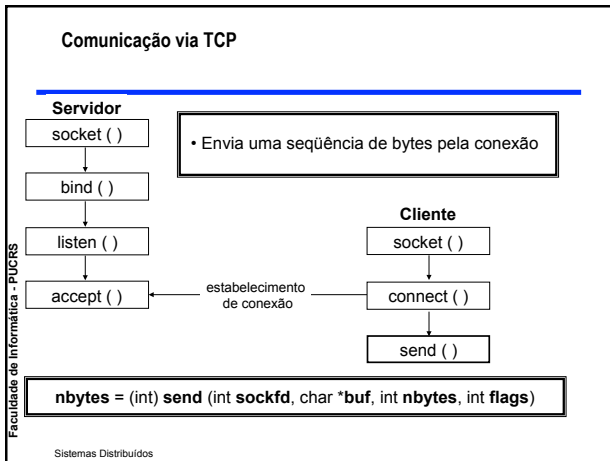
TCP

- *Transmission Control Protocol*
- Para comunicação longa (conexão)
- Confiável
- Transmissão de fluxo de dados: Não respeita limites de mensagens
- Baixo desempenho em comunicações curtas (?)
- Usos típicos:
 - Login remoto
 - Transferência de arquivo

UDP

- *User Datagram Protocol*
- Para comunicação curta (sem conexão)
- Não confiável
- Transmissão de pacotes: respeita limites de mensagens
- Pouco prático para comunicações longas (confiabilidade precisa ser programada)
- Usos típicos:
 - RPC
 - Broadcast





Comunicação via UDP

- Recebe pacote enviado do endereço informado
- Se não houver nada, bloqueia

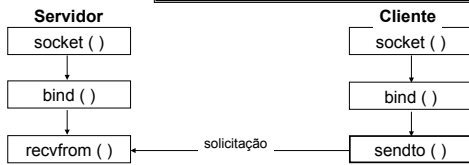


nbytes = (int) recvfrom (int sockfd, char *buf, int nbytes, int flags, struct sockaddr *from, int *addrlen)

Sistemas Distribuídos

Comunicação via UDP

- Envia pacote para o endereço informado

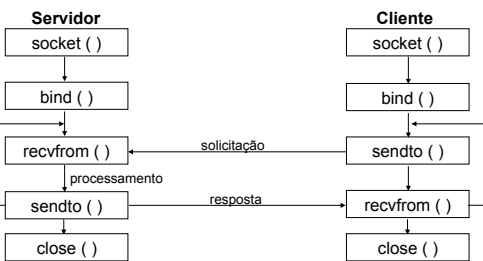


nbytes = (int) recvfrom (int sockfd, char *buf, int nbytes, int flags, struct sockaddr *from, int *addrlen)

Sistemas Distribuídos

Comunicação via UDP

- Diagrama completo



Sistemas Distribuídos

Comunicação entre processos

Exercícios:

- classifique cada uma das primitivas de pipes e sockets com relação
 - a modelo implícito e explícito de endereçamento
 - a comunicação bloqueante e não bloqueante
 - e a bufferização de mensagens
- Obs.: analise o maior número possível de opções oferecidas por estas interfaces
- usando PIPES, teste, com programas:
 - uso bidirecional de um pipe (existe?);
 - uso concorrente de um pipe;

Sistemas Distribuídos

Comunicação entre processos

Exercícios:

- usando sockets em plataforma linux ou solaris, e comunicação no modo datagrama,
 - implemente em linguagem C um mecanismo de comunicação com suporte a falhas, do tipo "two-message"
 - analise bem a interface de sockets para utilizar o máximo de sua funcionalidade
- implemente um servidor concorrente (trata vários clientes simultaneamente) de operações matemáticas utilizando pipes e depois sockets (tipo stream)
 - os clientes mandam os operandos e a operação, ficam a espera do resultado, e voltam a sortear operandos para o próximo pedido
 - o servidor calcula e manda o resultado devolta para os vários clientes
 - modifique a implementação para trabalhar com sockets no modo datagrama, adicionando o mecanismo de confirmação da questão anterior

Sistemas Distribuídos