

## Comunicação nos Sistemas Operacionais de Rede

- 1 - Comunicação nos Sist. Operacionais de Rede
  - ❖ TCP/IP e *Sockets*
  - ❖ IPX/SPX com TLI - *Transport Layer Interface*
  - ❖ APPC/SNA com CPI -C
  - ❖ RPC da Sun Microsystems
- 2 - TCP/IP e *Sockets*
- 3 - IPX/SPX com TLI - *Transport Layer Interface*
- 4 - APPC/SNA com CPI -C
- 5 - RPC da Sun Microsystems

Pag. 1

## 1 - Comunicação nos Sistemas Operacionais de Rede

- ◆ Dentre as principais opções de transporte disponíveis para implementação de serviços no âmbito dos sistemas operacionais de rede, podemos citar:
  - ❖ TCP/IP com *sockets*
  - ❖ IPX/SPX com TLI - *Transport Layer Interface*
  - ❖ APPC/SNA com CPI -C
  - ❖ RPC da Sun Microsystems
- ◆ As pilhas de transporte representam o sistema nervoso dos Sist. Operacionais de Rede e são elas que permitem agrupar os diferentes ambientes físicos.

Pag. 2

## ... 1 - Comunicação nos Sistemas Operacionais de Rede

Aplicação	Mensagem	Remote Procedure Call			Peer-to-Peer	
		Named Pipes	DCE	Sun		
Apresentação				NDR	XDR	
Sessão	NetBios	Sockets	TLI	CPI -C/APPC		
Transporte	NetBEUI	TCP/IP	SPX/IPX	LU6.2/APPN		
Rede		IEEE 802.2				
Enlace	Token Ring	Ethernet	SDLC	ISDN		
Física	Fibra Ótica		Cabo Coaxial		Par Trançado	

Pag. 3

## 2 - TCP/IP com sockets

- ◆ Trata-se de uma solução particularmente poderosa nos ambientes que requerem interconexão entre diferentes redes, que está disponível em praticamente todos S.O.
- ◆ Protocolo IP fornece o mecanismo básico para o roteamento de pacotes entre redes, mas por se tratar de um protocolo não confiável, depende da camada superior para o fornecimento de um serviço confiável como por exemplo, o Protocolo TCP.
- ◆ A Camada de Transporte do TCP/IP consiste de 2 protocolos que fornecem serviços de transporte fim-a-fim, são eles: TCP - *Transport Control Protocol* e UDP - *User Datagram Protocol*.

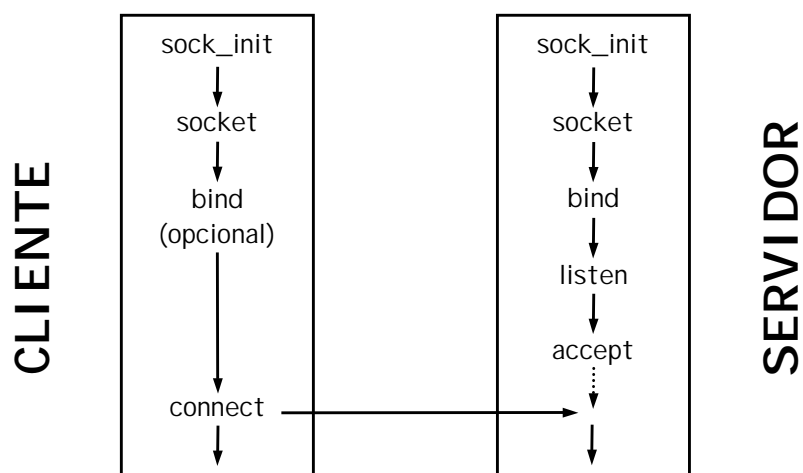
Pag. 4

## ... 2 - TCP/IP com sockets

- ◆ *Sockets* - foram introduzidos no Unix BSD em 1981 como uma interface genérica para suportar comunicação via rede entre sistemas UNIX.
- ◆ Em 1985 a Sun introduziu os protocolos NFS - *Network File System* e o RPC - *Remote Procedure Call* sobre a interface de *sockets*.
- ◆ Em 1986, a AT&T introduziu a interface TLI - *Transport Layer Interface*, interface esta que possui basicamente a mesma funcionalidade dos *sockets*, só que de uma forma mais independente da rede.

Pag. 5

## ... 2 - TCP/IP com sockets



Pag. 6

## ... 2 - TCP/IP com sockets

- ◆ **criação dos *sockets*** - os processos que utilizam *sockets* devem iniciar o ambiente de tempo de execução vinculado aos *sockets* através da chamada *socket\_init*, e depois deve ser criado o *socket* através da chamada *socket* ;
- ◆ **estabelecimento de uma porta de serviço** - os processos servidores devem associar *bind* os seus *sockets* a uma porta única de modo a se tornar conhecido na rede;
- ◆ **aguardar pedidos de conexão** - os processos servidores que utilizam o *socket* do tipo *stream* com conexão devem evocar a função *listen* de modo a indicar que o processo encontra-se pronto para aceitar conexões solicitadas pelos clientes;

Pag. 7

## ... 2 - TCP/IP com sockets

- ◆ **conexão ao servidor** - o cliente emite uma chamada de *connect* no *socket* do tipo *stream* com a finalidade de iniciar uma conexão em uma porta associada a um determinado serviço de um processo servidor.
- ◆ ... o processo cliente deve localizar o endereço internet do servidor a partir de um servidor de nomes, e após o retorno com sucesso, o *socket* do cliente estará vinculado à conexão estabelecida;

Pag. 8

## ... 2 - TCP/IP com sockets

- ◆ **aceitação da conexão** - o servidor aceita uma requisição de conexão através da chamada *accept*. O servidor pode ficar ou não bloqueado caso não existam requisições de conexão pendentes. Caso várias requisições estejam pendentes, normalmente trabalha-se com uma FIFO, ou seja, a requisição mais antiga será atendida primeiro.
- ◆ ... no retorno da chamada, e em existindo solicitação de requisição, é criado um novo *socket* o qual será utilizado daquele momento em diante para receber as solicitações emitidas por aquele cliente, assim como, quando do retorno dos resultados após a realização do serviço.

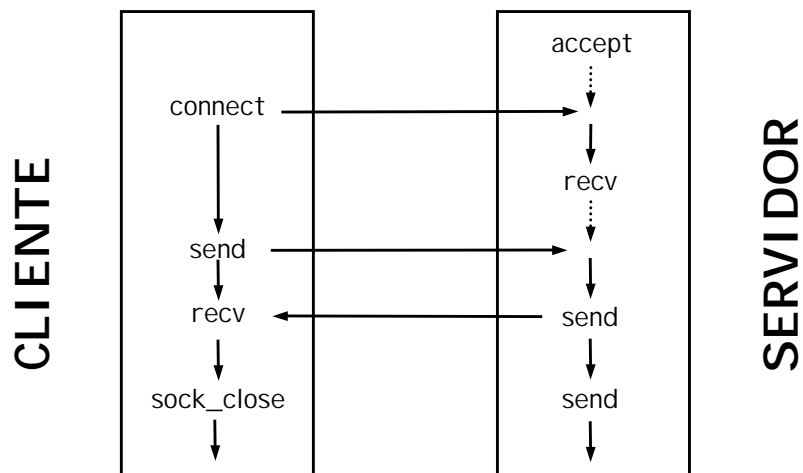
Pag. 9

## ... 2 - TCP/IP com sockets

- ◆ ... o *socket* original permanece livre para aceitar evocações de conexão emitidas por outros clientes;
- ◆ ... no caso da conexão estabelecida com um determinado cliente, o servidor pode decidir criar uma nova *thread* para gerenciar a comunicação com o cliente conectado e neste caso a thread original é utilizada para ficar aguardando novos clientes;
- ◆ ... uma outra possibilidade é utilizar a *thread* original para realizar o serviço solicitado.

Pag. 10

## ... 2 - TCP/IP com sockets



Pag. 11

## ... 2 - TCP/IP com sockets

- ◆ **interação cliente/servidor** - várias chamadas podem ser utilizadas pelos clientes e servidores para interação. As chamadas *readv*, *writv*, *send* e *recv* podem ser empregadas somente no caso de sockets que se encontram no estado conectado;
- ◆ ... as chamadas *sendto* e *recvfrom* podem ser utilizadas em qualquer momento mas necessitam de informação adicional relativamente aos endereços;

Pag. 12

## ... 2 - TCP/IP com sockets

- ◆ **eliminação do socket** - o cliente e o servidor utilizam a chamada ***sock\_close*** para encerramento da sessão TCP. Deve ser observado que o *socket* original continua apto para iniciar outras sessões. Caso o servidor tenha criado uma *thread* específica para atender o serviço, esta *thread* é encerrada;
- ◆ **aceitar novas interações ou encerrar as operações** - o processo servidor pode evocar a chamada ***sock\_close*** relativamente ao *socket* original conhecido pelos clientes encerrando as suas atividades ou então aceitar novos pedidos de conexão por parte dos clientes;

Pag. 13

## 3 - IPX/SPX da Novell

- ◆ **IPX/SPX**, que é a pilha de comunicação nativa do ***NetWare*** da ***Novell***, é uma implementação do protocolo de transporte e de rede **XNS - Xerox Network Services**.
- ◆ A camada de rede do **IPX/SPX** é suportada pelo protocolo **IPX - Internal Packet Exchange** - um protocolo do tipo *datagrama* sem garantias, enquanto o **SPX** fornece um protocolo confiável sobre o **IPX**.
- ◆ ... o **IPX** possui 12 chamadas de API que podem ser utilizadas para realização de serviços de *datagrama*, enquanto o **SPX** oferece 16 chamadas na sua API.

Pag. 14

## ... 3 - IPX/SPX da Novell

- ◆ A pilha **NetWare** da **Novell** suporta quatro protocolos no topo da pilha SPX/IPX, são eles:
  - ❖ TLI - *Transport Layer Interface*;
  - ❖ NetBios;
  - ❖ Named Pipes;
  - ❖ APIs IPX/SPX.
- ◆ O Protocolo **TLI** é utilizado pela **Novell** como padrão.

Pag. 15

### 3.1 - IPX/SPX da Novell - TLI

- ◆ Dentre esses protocolos, o **TLI** é utilizado pela **Novell** como padrão para as aplicações que necessitam da camada de transporte. Uma aplicação desenvolvida em TLI é, teoricamente, independente da pilha de protocolos.
- ◆ Com pouquíssimas modificações, é possível que uma mesma aplicação execute em uma pilha IPX/SPX ou TCP/IP. TLI é uma implementação atualizada dos *sockets* de Berkeley e a sua API consiste de 25 chamadas.

Pag. 16



## 3.2 - IPX/SPX da Novell - NetBIOS

- ◆ **NetBIOS** foi o primeiro protocolo de redes voltado para comunicação programa. É uma interface utilizada sobre várias pilhas de protocolo como TCP/IP, XNS, Vines, OSI, SPX/IPX e redes IBM/Microsoft.
- ◆ **NetBEUI** é a pilha de protocolo que acompanha os produtos IBM e Microsoft. Originalmente foi introduzido como o transporte utilizado pelos comandos NetBIOS.
- ◆ Em geral, a literatura referencia NetBIOS englobando a interface NetBIOS e a pilha NetBEUI. Algumas vezes isto provoca confusão pois NetWare não tem nada a haver com NetBEUI mas, por outro lado, utiliza NetBIOS como interface tanto para o SPX/IPX como para o TCP/IP.

Pag. 17

## ... 3.2 - IPX/SPX da Novell - NetBIOS

- ◆ NetBEUI suporta serviços de datagrama e orientado a conexão. Também suporta um serviço dinâmico de nomes.
- ◆ Os comandos do NetBIOS são especificados em uma estrutura denominada ***Network Control Block*** - NCB.
- ◆ ... essa estrutura contém também os parâmetros associados com o comando e os campos nos quais o NetBIOS retornará informação para o programa.

Pag. 18

### **3.3 - IPX/SPX da Novell - Named Pipes**

- ◆ São mecanismos IPC usados no OS/2 e UNI X (onde são também denominados FI FOs).
- ◆ O objetivo é tratar o canal de comunicação envolvendo dois processos como se fosse um arquivo.
- ◆ Desta forma, através de canais nomeados, os processos podem trocar dados como se eles estivessem escrevendo ou lendo em/de um arquivo seqüencial.
- ◆ ... no lugar de colocar no disco os dados gravados no canal, os dados são convertidos para uma mensagem e enviados para outro processo.

Pag. 19

### **3.3 - IPX/SPX da Novell ... Named Pipes**

- ◆ Semelhantemente, uma operação de leitura retira os dados da mensagem recebida e não do disco.
- ◆ No UNI X, os canais nomeados somente são utilizados por dois processos executando na mesma máquina.
- ◆ No OS/2, os canais nomeados podem ser utilizados para comunicação entre processos via rede, especificando-os como canais remotos. O OS/2 redirecionam as operações relativas ao canal nomeado remoto usando o NetBI OS para alcançar o servidor especificado.

Pag. 20

## 4 - APPC/SNA com CPI -C

- ◆ APPC - *Application Program to Program Communication* é o protocolo da IBM para comunicação entre processos em um ambiente distribuído suportado por rede SNA.
- ◆ A IBM está evoluindo a SNA na direção de um sistema operacional distribuído verdadeiro suportando serviços de diretório de rede, acesso transparente a recursos da rede tais com em servidores, aplicações, *displays*, impressoras e dados, e fluxos de dado e gerenciamento integrado de rede.

Pag. 21

## ... 4 - APPC/SNA com CPI -C

- ◆ A infra-estrutura responsável por este processamento distribuído real denomina-se APPN - *Advanced Peer-to-Peer Network*.
- ◆ O objetivo da APPN é criar uma internet SNA sem o papel central do *mainframe* na hierarquia tradicional das configurações SNA. O *mainframe* é um nó como outro qualquer na rede.
- ◆ APPN permite que aplicações LU6.2 (denomina-se da sessão na qual programas comunicam-se na forma de pares, isto é, programa a programa) através das APIs APPC ou CPI -C - *Common Programming Interface for Communications*.

Pag. 22

## ... 4 - APPC/SNA com CPI -C

- ◆ Através do uso da APPC ou CPI -C, um programa pode conversar através das 50.000 redes SNA instaladas no mundo, com qualquer outro programa.
- ◆ A arquitetura APPC é uma solução proposta pela IBM para comunicação programa-programa, processamento de transações distribuídas e acesso a base de dados remota. A interface de aplicação com o APPC denomina-se CPI -C.
- ◆ Trata-se de um grupo de chamadas do sistema correspondentes aos comandos, também chamados de verbos, da APPC.

Pag. 23

## ... 4 - APPC/SNA com CPI -C

- ◆ Ativar uma chamada do sistema CPI -C resulta na formatação automática de um bloco de controle APPC usando os parâmetros da chamada e o bloco de controle sendo passado à LU6.2.
- ◆ A IBM está expandindo o papel da interface CPI -C para ambientes não-SNA como, por exemplo, TCP/IP e OSI .
- ◆ A API CPI -C possui em torno de 40 chamadas e a APPC consiste de aproximadamente 60 chamadas. Muitas destas chamadas estão relacionada com funções de configuração.

Pag. 24

## ... 4 - APPC/SNA com CPI -C

- ◆ A arquitetura APPC é otimizada para suportar conversações de longa duração.
- ◆ No caso, o custo para criação de uma sessão e alocação de conversações é elevado se apenas algumas mensagens forem enviadas. No caso de uma conversação demorada, este *overhead* é amortizado.

Pag. 25

## 5 - RPC da Sun Microsystems

- ◆ A Sun Microsystems desenvolveu o RPC na década de 80 com o objetivo de permitir o desenvolvimento de aplicações cliente/servidor sem necessidade de programar com interfaces sob a camada de transporte como, por exemplo, *sockets*.
- ◆ Este mecanismo de interação cliente/servidor serve de base para o desenvolvimento de serviços estendidos em sistemas operacionais de rede e encontra-se disponível em praticamente todos os sistemas operacionais derivados do UNIX.

Pag. 26

## ... 5 - RPC da Sun Microsystems

- ◆ Exemplo de servidores acessados via RPC:
  - ❖ servidores NFS (*Network File System*): permitem acesso transparente à sistemas de arquivos remotos;
  - ❖ portmapper: diretório de servidores RPC;
  - ❖ servidor de NIS (*Network Information System*):
  - ❖ serviço de nomes e *passwords*;
  - ❖ servidores de estatísticas.
- ◆ RPC permite ao projetista de uma aplicação distribuída focalizar na aplicação e não na comunicação entre os vários componentes que compõem a aplicação.

Pag. 27

## ... 5 - RPC da Sun Microsystems

- ◆ RPC utiliza os conceitos sedimentados de programação modular e os conceitos mais recentes de orientação a objetos. Assim a aplicação é dividida em módulos:

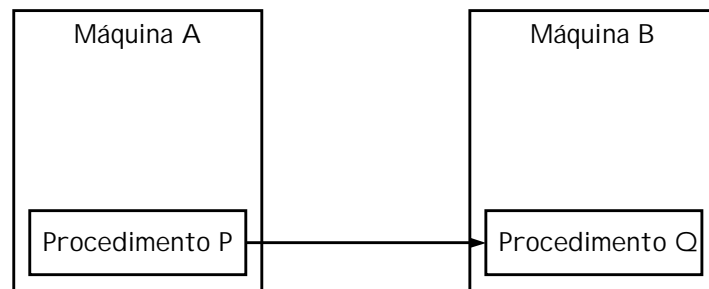
**módulo = { proc. relacionados + dados associados }**

- ❖ procedimentos de um módulo podem chamar procedimentos de outro módulo de forma arbitrariamente aninhada
- ❖ módulos podem ser localizados em máquinas distintas para aumentar o desempenho da aplicação

Pag. 28

## ... 5 - RPC da Sun Microsystems

- ◆ **chamada de procedimento remoto** - um procedimento P de um módulo localizado numa máquina A chama um procedimento Q de outro módulo localizado numa máquina distinta B.



Pag. 29