Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

COMUNICAÇÃO EM UM AMBIENTE DE COMPUTAÇÃO DISTRIBUÍDA

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

COMUNICAÇÂO ENTRE PROCESSOS

- A comunicação entre processos está no coração de qualquer Sistema Distribuído
- Como processos em diferentes máquinas são capazes de trocar informações?
  - A comunicação em SD é sempre baseada em troca de mensagens
  - Troca de mensagens é mais difícil do que usar primitivas baseadas em memória compartilhada
- Desejável obter modelos onde a complexidade da comunicação seja transparente para o desenvolvedor

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

COMUNICAÇÂO ENTRE PROCESSOS

- Os processos que se comunicam tem que obedecer regras estabelecidas => **Protocolos**

- Como processos em diferentes máquinas trocam informações?
  - Processo A monta uma MSG em seu próprio espaço de endereçamento
  - Executa uma chamada de sistema, SO envia a MSG pela rede até B
  - A e B precisam concordar com significado de bits

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

COMUNICAÇÂO ENTRE PROCESSOS

Vários acordos diferentes são necessários:

- Quantos volts são necessários para sinalizar um bit 0 ou 1?
- Como o receptor sabe qual é o último bit da mensagem?
- Como o receptor pode detectar se uma mensagem foi danificada ou perdida?
- Qual o comprimento de cadeias e como elas são representadas?

Os acordos são estabelecidos em diferentes níveis

MODELO OSI

Para ficar mais fácil lidar com os vários níveis e questões envolvidas na comunicação:

- International Organization for Standardization (ISO) desenvolveu um modelo de referência que identifica os vários níveis envolvidos

Modelo OSI (Open Systems Interconnection Reference Model)

- Cada camada lida com um aspecto específico da comunicação
- Problema quebrado em pedaços gerenciados independentemente

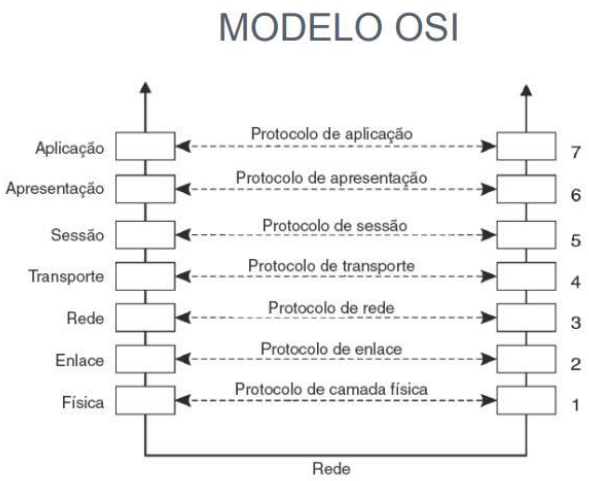
Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

MODELO OSI

- A comunicação é dividida em até **7 níveis/camadas**
- **Cada camada** oferece uma interface para a **camada que está acima** dela
  - A interface é um conjunto de operações que definem um serviço
- Modelo de referência ≠ Protocolos
  - Protocolos OSI nunca foram populares
  - Os protocolos desenvolvidos para internet são os mais usados

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

MODELO OSI



Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

PROTOCOLOS DE TRANSPORTE NA INTERNET

Transmission Control Protocol (TCP)
- Orientado a Conexão
- Confiável, porém "lento"

Universal Datagram Protocol (UDP)
- Sem conexão
- "Rápido", porém não confiável

Escolha esta, ligada as características da aplicação!

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

PROTOCOLOS DE TRANSPORTE NA INTERNET

CAMADA DE APLICAÇÃO

- A Internet agrupou camadas superiores em uma única camada: **Apresentação + Sessão + Aplicação**
- Na prática somente a camada de aplicação é usada
- A intenção original da camada de Aplicação OSI era conter um conjunto de aplicações padronizadas de rede como correio eletrônico e transferência de arquivos
- Atualmente é um repositório de todas as aplicações que não se ajustam as camadas subjacentes

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

PROTOCOLOS DE TRANSPORTE NA INTERNET

CAMADA DE APLICAÇÃO

- É necessário fazer uma distinção entre aplicação para redes e protocolos da camada de aplicação
  - **FTP** (File Transfer Protocol) define um protocolo para transferir arquivos entre um cliente e um servidor
  - Programa FTP => aplicação de usuário para transferir arquivos que implementa o FTP
  - **HTTP** (HyperText Transfer Protocol) é projetado para gerenciar e manipular remotamente a transferência de páginas Web
  - O protocolo é implementado por aplicações como *navegadores* Web

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

PROTOCOLOS DE TRANSPORTE NA INTERNET

CAMADA DE MIDDLEWARE

- O objetivo do Middleware é prover serviços e protocolos comuns que podem ser usados por diversas aplicações
- Consiste de um conjunto rico de protocolos
- (Un)marshaling de dados (padronização de formato), necessário para sistemas integrados
- Protocolos de nomeação, para permitir compartilhamento fácil de recursos
- Protocolos de segurança para comunicação segura
- Mecanismos de dimensionamento, para replicação e caching

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

PROTOCOLOS DE TRANSPORTE NA INTERNET

CAMADA DE MIDDLEWARE

- Domain Name System (DNS): procura um endereço de rede associado a um nome
    - No modelo OSI estaria no nível de aplicação, mas este serviço independe da aplicação

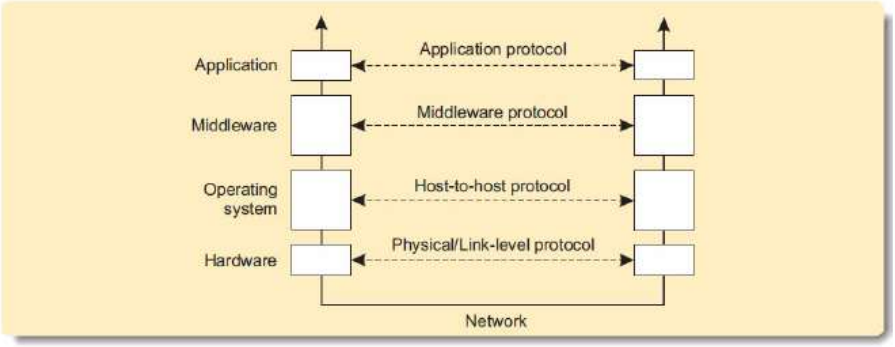
- Protocolos de autorização para garantir que usuários e processos só tenham acesso a recursos autorizados

- Protocolo para acessar acesso compartilhado

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

PROTOCOLOS DE TRANSPORTE NA INTERNET

MODELO DE REFERÊNCIA ADAPTADO



Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

PROTOCOLOS DE TRANSPORTE NA INTERNET

MODELO DE REFERÊNCIA ADAPTADO

- As camadas de apresentação e sessão foram substituídas pela camada de middleware que contém protocolos independentes da aplicação
- As camadas de transporte e rede foram agrupadas em serviços de comunicação oferecidos pelo sistema operacional
- Sistema operacional gerencia hardware

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

TIPOS DE COMUNICAÇÃO

PERSISTÊNCIA

Comunicação Persistente:

- Uma mensagem é armazenada pelo *middleware* de comunicação **durante o tempo que for necessário para entregá-la ao receptor**
- Não é necessário que a aplicação remetente continue em execução após submeter a mensagem
- A aplicação receptora não precisa estar em execução no momento da submissão da mensagem

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

TIPOS DE COMUNICAÇÃO

PERSISTÊNCIA

Comunicação Transiente:

- Uma mensagem é armazenada pelo *middleware* de comunicação **somente durante o tempo em que a aplicação remetente e a aplicação receptora estiverem executando**
- Caso o *middleware* não possa entregar a mensagem, ela será descartada
- Normalmente os serviços de comunicação do nível de transporte oferecem comunicação transiente

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

TIPOS DE COMUNICAÇÃO

SINCRONIZAÇÃO

Comunicação Assíncrona:

- Remetente continua sua execução imediatamente após ter submetido sua mensagem para transmissão
- Middleware armazena temporariamente a mensagem

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

TIPOS DE COMUNICAÇÃO

SINCRONIZAÇÃO

Comunicação Síncrona:

- Remetente é bloqueado até saber que sua requisição foi aceita:
  - Até que o *middleware* avise que se encarregará da transmissão
  - Até que sua requisição seja entregue ao receptor
  - Até que sua requisição tenha sido totalmente processada, isto é quando o receptor retornar uma resposta

TIPOS DE COMUNICAÇÃO

EXEMPLO 1

Computação **cliente/servidor** é geralmente baseada no modelo de comunicação **transiente síncrona.**

- Cliente e servidor devem estar ativos no momento da comunicação
- Cliente submete um pedido e bloqueia até que receba uma resposta
- Servidor essencialmente fica esperando requisições dos clientes, e as processa subsequentemente

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

TIPOS DE COMUNICAÇÃO

EXEMPLO 1

Desvantagens

- Cliente não pode fazer outra coisa enquanto espera resposta
- Falhas devem ser tratadas imediatamente porque o cliente está esperando
- Para algumas aplicações não é apropriado (correio, news)

TIPOS DE COMUNICAÇÃO

EXEMPLO 2

Middleware orientado a mensagem com comunicação **persistente assíncrona**

- Processos enviam mensagens uns aos outros que são enfileiradas
- Remetente envia uma mensagem e não precisa receber uma resposta imediatamente
- Middleware deve oferecer tolerância a falhas

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

CHAMADA DE PROCEDIMENTO REMOTO

"Permite a processos chamar procedimentos localizados em outras máquinas"

(Birrell e Nelson - 1984)

Surge da necessidade de obter transparência de acesso em sistemas distribuídos => o que não ocorre com o uso dos procedimentos *send e receive*

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

CHAMADA DE PROCEDIMENTO REMOTO

RPC (Chamada de Procedimento Remoto)

- Um processo da máquina A chama um procedimento da máquina B
- O processo A é suspenso e o procedimento é executado em B
- Informações podem ser transportadas nos parâmetros e no resultado do procedimento
- Nenhum aspecto da troca de mensagens é visível para o programador!

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

CHAMADA DE PROCEDIMENTO REMOTO

A ideia fundamental é fazer com que uma chamada de procedimento remoto pareça com uma chamada local => **Transparência**

A Transparência é alcançada com o uso de *stubs* => apêndices

- *Stub do cliente*
- *Stub do servidor*

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

CHAMADA DE PROCEDIMENTO REMOTO

Stub do cliente

- Responsável por empacotar os parâmetros em uma mensagem e enviá-la para a máquina do servidor
- Quando a resposta chega, o resultado é copiado para o cliente, e o controle volta a ele

Stub do servidor

- Responsável por desempacotar os parâmetros
- Chama o procedimento do servidor e retorna a resposta para máquina do cliente

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

CHAMADA DE PROCEDIMENTO REMOTO

STUB DO CLIENTE

- Em um sistema tradicional, *read* é um procedimento que faz uma chamada de sistema ao SO
- Com RPC uma versão diferente do *read* é colocada na biblioteca => stub/apêndice do cliente
- Ela empacota os parâmetros em uma mensagem e pede que esta mensagem seja enviada para o servidor
- O apêndice de cliente efetua o *send* e o *receive*, bloqueando a si mesmo até que a resposta volte

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

CHAMADA DE PROCEDIMENTO REMOTO

STUB DO CLIENTE

CHAMADA DE PROCEDIMENTO REMOTO

Princípio de RPC entre um programa cliente e um programa servidor.

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

CHAMADA DE PROCEDIMENTO REMOTO

STUB DO SERVIDOR

- Quando a mensagem chega ao servidor, o SO do servidor a repassa para um stub de servidor
- Equivalente ao stub de cliente => transforma requisições que vêm pela rede em chamadas de procedimento locais
- Desempacota os parâmetros da mensagem vinda do cliente e executa o procedimento do servidor
- O resultado é armazenado em um buffer e devolvido ao stub do servidor, que empacota os dados e o envia ao stub do cliente

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

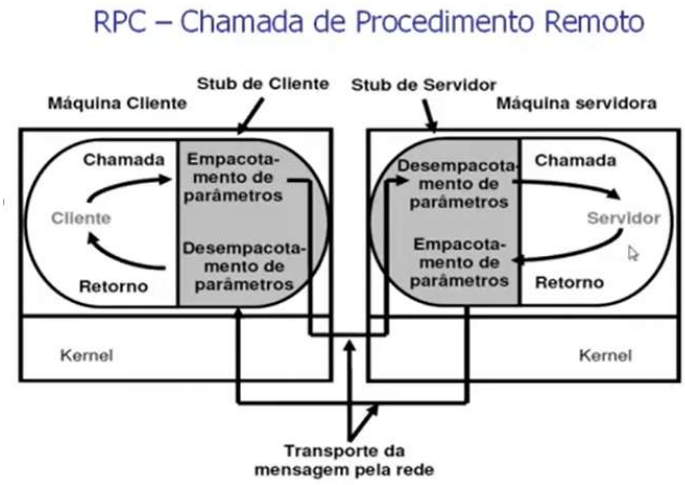
CHAMADA DE PROCEDIMENTO REMOTO

ETAPAS DE PROCEDIMENTO REMOTO

1. Procedimento de cliente chama o stub de cliente do modo normal
2. O stub de cliente constrói uma mensagem e chama o SO local
3. O SO do cliente envia a mensagem para o SO remoto
4. O SO remoto dá a mensagem ao stub de servidor
5. O stub de servidor desempacota os parâmetros e chama o serviço
6. O servidor faz o serviço e retorna o resultado para o stub
7. O stub de servidor empacota o resultado em uma mensagem e chama o SO Local
8. O SO do servidor envia a mensagem ao SO do cliente
9. O SO do cliente dá a mensagem ao stub de cliente
10. O stub desempacota o resultado e retorna ao cliente

Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

CHAMADA DE PROCEDIMENTO REMOTO



Curso: **ENGENHARIA DE COMPUTAÇÃO**
Disciplina: **SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS**
Professor: **KLAUSNER VIEIRA GONÇALVES**

CHAMADA DE PROCEDIMENTO REMOTO

Resumindo

- Permite a um cliente o acesso a um serviço remoto por meio de uma simples chamada a um procedimento local
- Possibilita que programas clientes sejam escritos de modo simples
- Pode localizar automaticamente o servidor correto
- Estabelece a comunicação entre software cliente e software servidor