## Definições:

*“Várias definições de sistemas distribuídos já foram dadas na literatura, nenhuma delas satisfatória e de acordo com nenhuma das outras” [Tanenbaum]*

Michael Schröder : “Um sistema distribuído são vários computadores fazendo algo juntos”

Leslie Lamport: “Um sistema é distribuído quando a falha de um computador do qual você nunca ouviu falar o impede de fazer qualquer coisa”

Tanenbaum: “Um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente”

Colouris, Dolimore, Kindberg : “Coleção de computadores autônomos interconectados através de uma rede de comunicação, equipado com um software de sistema distribuído. Um software de sistema distribuído permite que computadores possam coordenar suas atividades e compartilhar os recursos do sistema: hardware, software e dados”

## Transparência:

Acesso – Oculta diferenças na representação de dados e no modo de acesso a um recurso

Localização – Oculta o lugar em que um recurso está localizado

Migração – Oculta que um recurso pode ser movido para outra localização

Relocação – Oculta que um recurso pode ser movido para outra localização enquanto em uso

Replicação – Oculta que um recurso é replicado

Concorrência – Oculta que um recurso pode ser compartilhado por diversos usuários concorrentes

Falha – Oculta a falha e a recuperação de um recurso

## Arquiteturas:

Em camadas

Em objetos

Em dados

Em eventos

Centralizadas

Descentralizadas

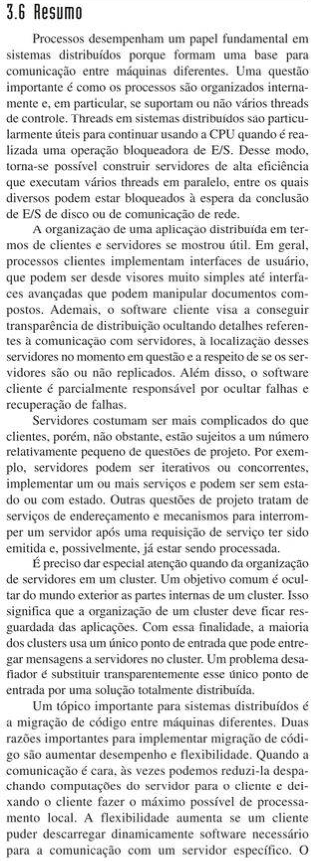
Cliente-servidor:

Peer-to-peer:

Middleware: camada de software

CORBA, RMI Java, web services, DCOM, RM-ODP

Processos:





Comunicação:

Protocolos em camadas

Protocolos TCP – FTP – HTTP

Assíncrona / Síncrona

RPC

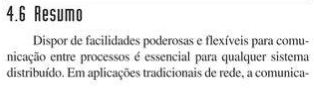
Passagem de parâmetros

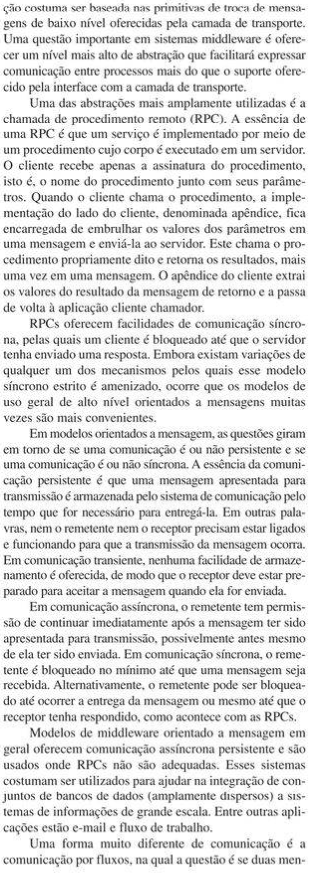
Socket

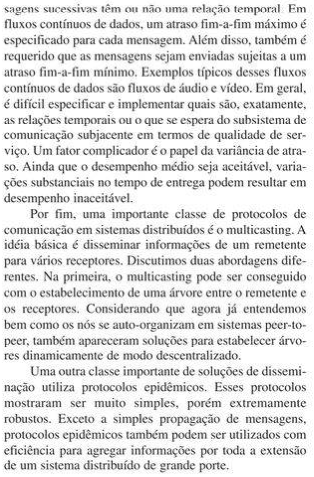
Mensagens

Qualidade de serviço

Multicast







Sincronização:

- precisão dos relógios

- algoritmos de sincronização

Segurança:

- criptografia

## Glossário:

OpenMP – OpenMultiprocessing - é uma interface de programação de aplicativo(API) para a programação multi-processo de memória compartilhada em múltiplas plataformas.

OpenCL – Open Computing Language – é um padrão para programação paralela em ambientes heterogêneos que possibilita o desenvolvimento de aplicações que executem sobre um conjunto de dispositivos de diferentes fabricantes, como CPUs e GPUs.

CUDA – Cumpute Unified Device Architeture – é uma plataforma para programação paralela de propósito geral em GPUs fabricadas pela NVIDIA

MPI – Message Passing Interface - é um padrão para comunicação de dados em computação paralela.

SIMD – Single Instruction, Multiple Data –

MIMD – Multiple Instruction, Multiple Data -

HPC – High Performance Computing -

DSM – Distributed Shared Memory – é uma abstração usada para o compartilhamento de dados entre computadores que não compartilham memória física. Os processos acessam a DSM por meio de leituras e atualizações no que parece ser uma memória normal, dentro de seu espaço de endereçamento.

UMA – Uniform Memory Access - é uma arquitetura de memória compartilhada, sendo o tempo de acesso ao pool de memórias uniforme, independente da localização do processador. Cada processador possui um cache privado que é sincronizado com a memória principal através de algoritmos de cache-coherence embarcados no hardware

NUMA – Non-Uniform Memory Access -