

***Faculdade de Tecnologia Senac
Tecnologia em Sistemas de Informação
Sistemas Distribuídos***

Sistemas Distribuídos Visão Geral

Lincoln Moraes

Porto Alegre, Agosto/2006

- ➡ Revisão de conceitos básicos de Sistemas Operacionais;
- ➡ Evolução histórica;
- ➡ Definições sobre Sistemas Distribuídos (SD);
- ➡ Vantagens e desvantagens de SD;
- ➡ Desafios / oportunidades em SD.

➔ Revisão de conceitos básicos de Sistemas Operacionais;

➔ Evolução histórica:

➔ Definições sobre Sistemas Distribuídos (SD);

➔ Vantagens e desvantagens de SD;

➔ Desafios / oportunidades em SD.

Definição simplificada

Conjunto de programas que gerenciam os recursos (hardware e software) de um sistema de computação.

Aplicativos (solicitando recursos)



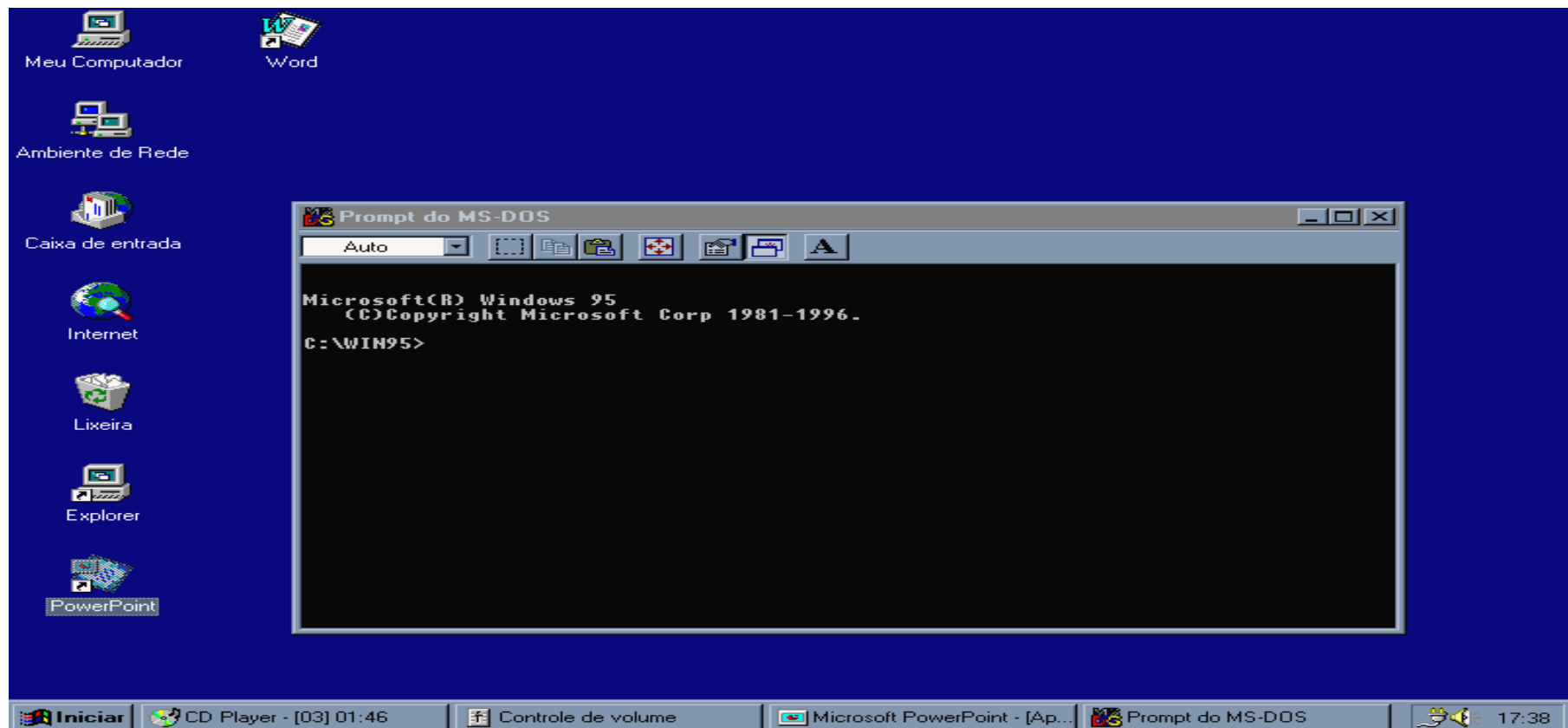
Hardware (recursos solicitados)

Principais finalidades

- ➡ Criar uma máquina virtual, através da facilidade de seu uso;
- ➡ Compartilhamento dos recursos de uma forma eficiente, organizada e segura;
- ➡ Evitar retrabalho e redundâncias de código.

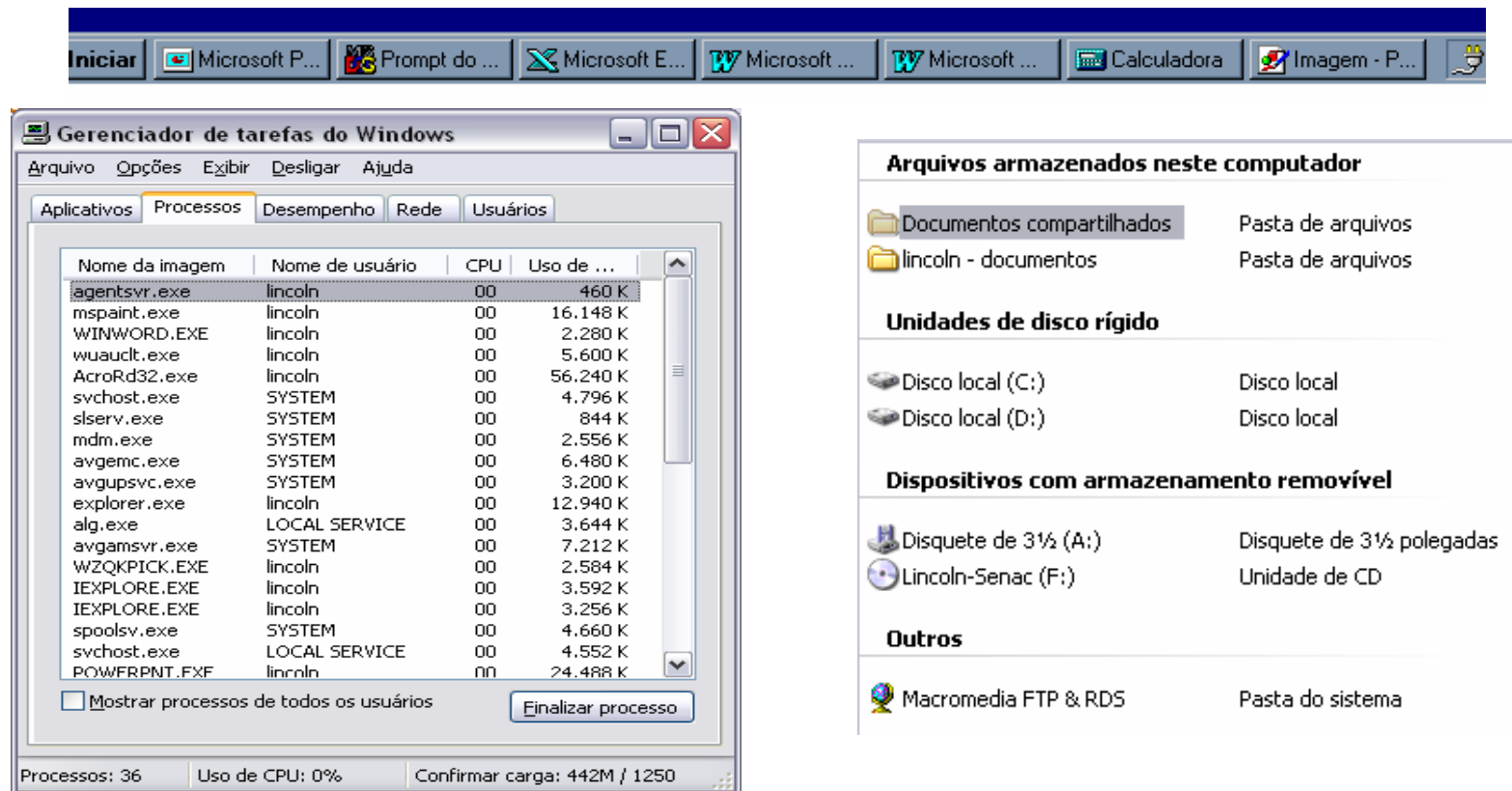
Principais finalidades

➡ Criar uma máquina virtual, através da facilidade de seu uso;



Principais finalidades

- ➔ Compartilhamento dos recursos de uma forma eficiente, organizada e segura;



The image shows two windows from a Windows XP desktop. The top window is the Task Manager (Gerenciador de tarefas do Windows) with the 'Processes' (Processos) tab selected. It displays a list of running processes with columns for Name, User, CPU usage, and Memory usage. The bottom window is 'My Computer' (Meu Computador), showing the 'Files stored on this computer' (Arquivos armazenados neste computador) section. It lists shared documents, hard disk drives (C: and D:), removable storage devices (A: and F:), and other system folders.

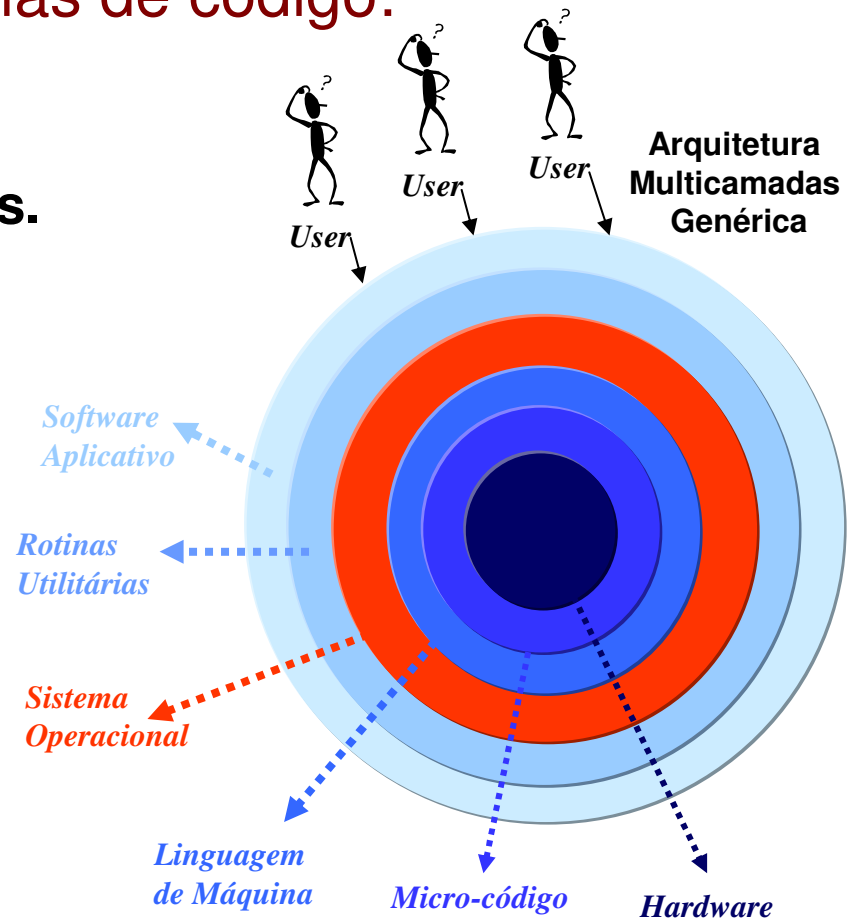
Nome da imagem	Nome de usuário	CPU	Uso de ...
agentsvr.exe	lincoln	00	460 K
mspaint.exe	lincoln	00	16.148 K
WINWORD.EXE	lincoln	00	2.280 K
wuauclt.exe	lincoln	00	5.600 K
AcroRd32.exe	lincoln	00	56.240 K
svchost.exe	SYSTEM	00	4.796 K
slserv.exe	SYSTEM	00	844 K
mdm.exe	SYSTEM	00	2.556 K
avgemc.exe	SYSTEM	00	6.480 K
avgupsvc.exe	SYSTEM	00	3.200 K
explorer.exe	lincoln	00	12.940 K
alg.exe	LOCAL SERVICE	00	3.644 K
avgamsvr.exe	SYSTEM	00	7.212 K
WZQKPICK.EXE	lincoln	00	2.584 K
IEXPLORE.EXE	lincoln	00	3.592 K
IEXPLORE.EXE	lincoln	00	3.256 K
spoolsv.exe	SYSTEM	00	4.660 K
svchost.exe	LOCAL SERVICE	00	4.552 K
POWERPNT.FXF	lincoln	00	24.488 K

Nome	Tipo
Documentos compartilhados	Pasta de arquivos
lincoln - documentos	Pasta de arquivos
Unidades de disco rígido	
Disco local (C:)	Disco local
Disco local (D:)	Disco local
Dispositivos com armazenamento removível	
Disquete de 3 1/2 (A:)	Disquete de 3 1/2 polegadas
Lincoln-Senac (F:)	Unidade de CD
Outros	
Macromedia FTP & RDS	Pasta do sistema

Principais finalidades

➡ Evitar retrabalho e redundâncias de código:

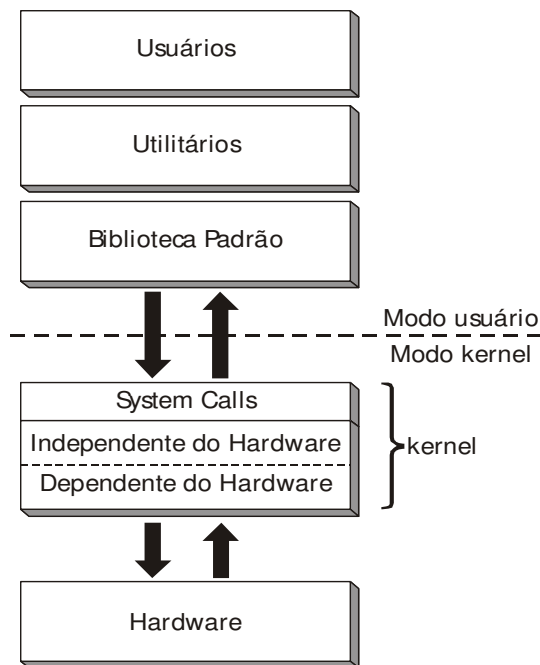
↓
Cada camada é responsável por implementar serviços específicos.



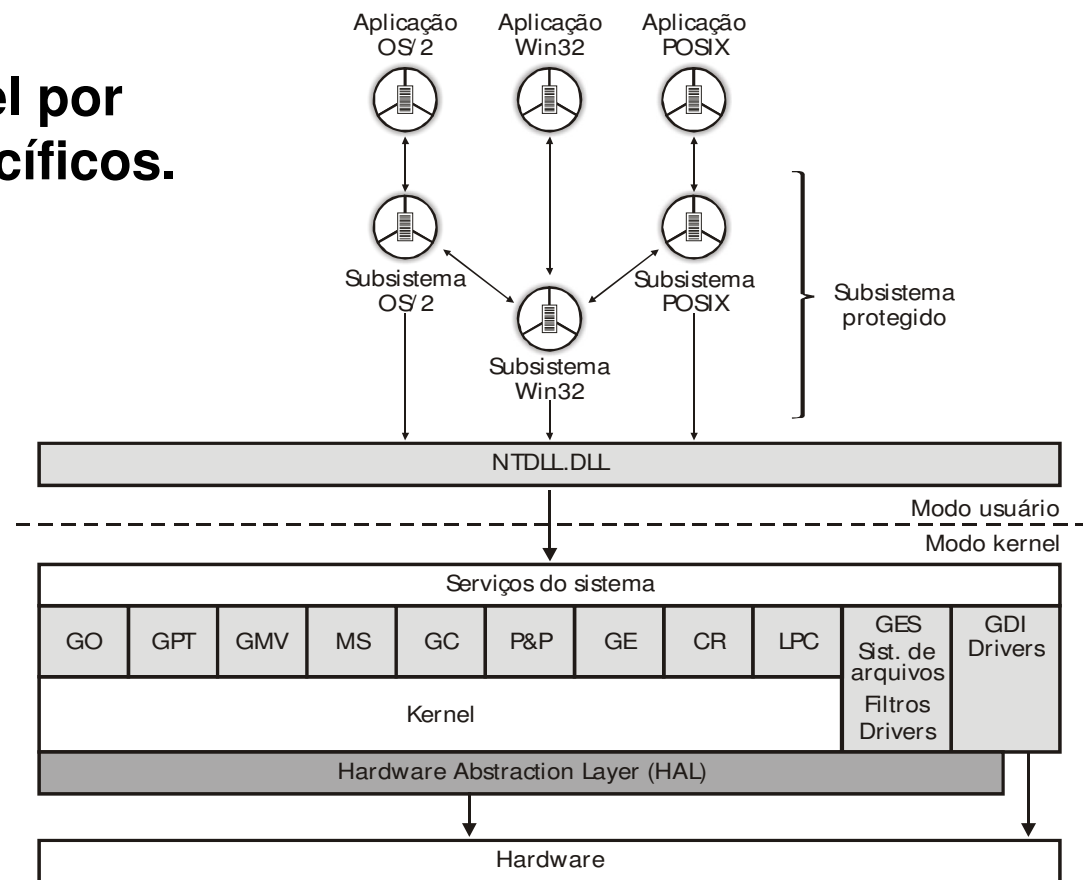
Principais finalidades

➡ Evitar retrabalho e redundâncias de código:

Cada camada é responsável por implementar serviços específicos.



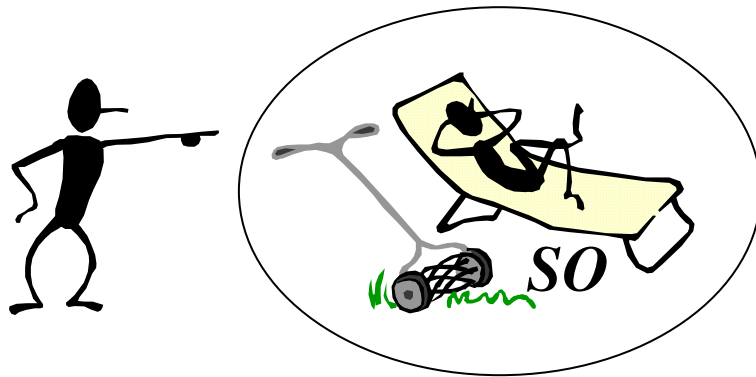
Família Unix



Família Windows 2K



- ➡ Sistemas Monotarefa / Monoprogramáveis;
- ➡ Sistemas Multitarefas / Multiprogramáveis;
- ➡ Sistemas Distribuídos (Multiprocessados).

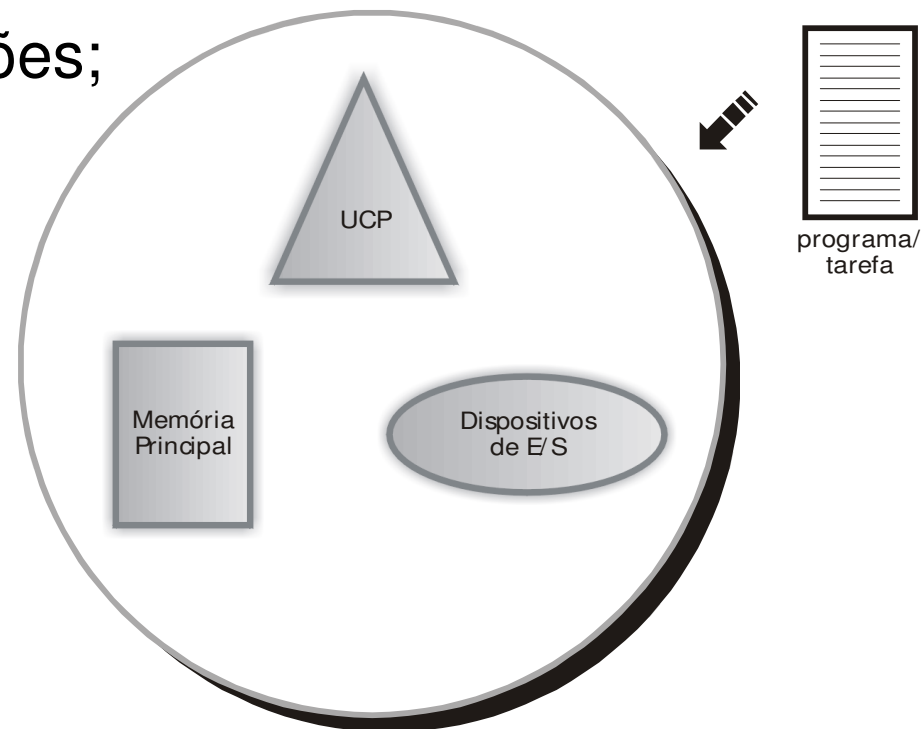


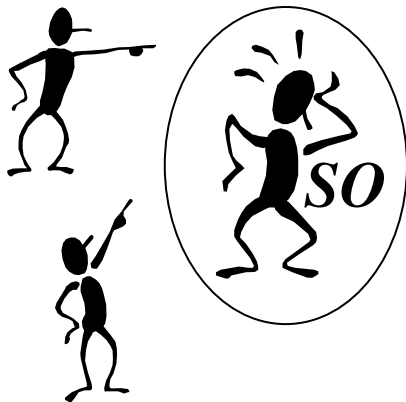
⇒ Único processo aloca todos os recursos disponíveis;

⇒ Execução serial das aplicações;

⇒ Sub-utilização de recursos (ex: processador);

⇒ ex: MS-DOS, CP/M.



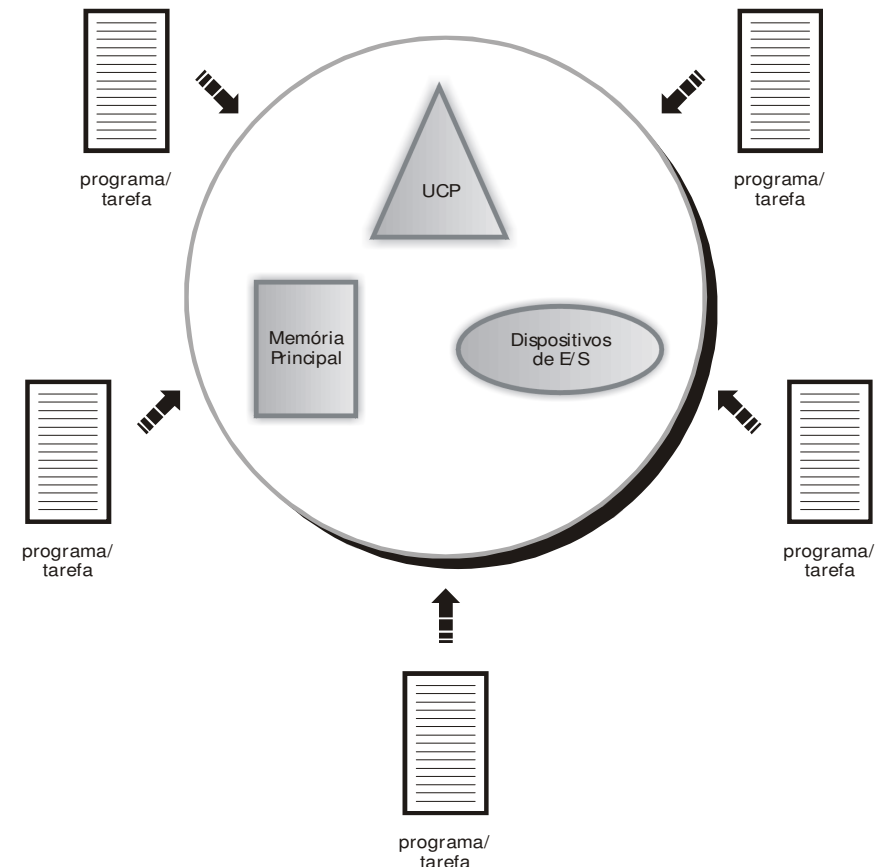


⇒ Vários processos compartilhando os recursos existentes;

⇒ Execução **CONCORRENTE** dos processos;

⇒ Preocupação com problemas de privacidade (segurança) e sincronismo na execução dos processos;

⇒ Sistemas mais complexos:
ex: W2K, Unix, Linux, Netware, OS/2, BeOS,
...





Classificação de acordo com:

⇒ Gerenciamento da execução de processos:

⇒ Lote / *Batch*;

⇒ Tempo-Compartilhado / *Time-Sharing*;

⇒ Tempo-Real / *Real-Time*.

⇒ Número de usuários por estação:

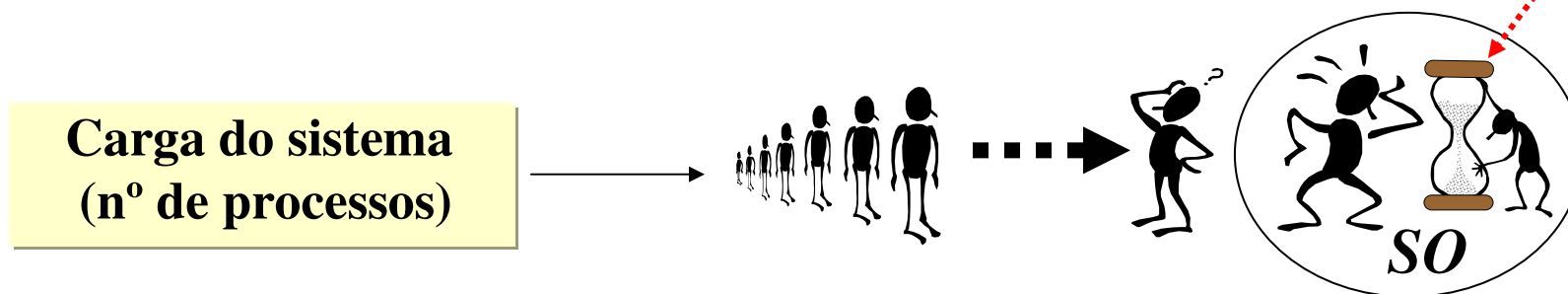
⇒ Monousuário;

⇒ Multiusuário.

Tempo-Compartilhado / Time-Sharing:

- ➔ Noção da fatia de tempo (*quantum* / *time-slice*);
- ➔ Sistemas conhecidos como *on-line*: baixo tempo de resposta;
- ➔ Processamento seqüencial dos processos;
- ➔ Processamento baseado em cooperação / **preempção**;
- ➔ Tempo de resposta depende da carga do sistema.

Time-slice

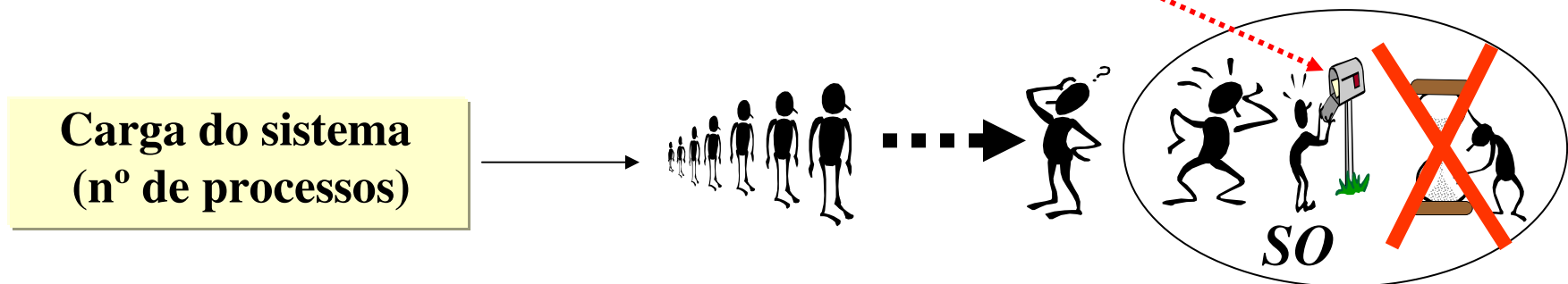


Tempo-Compartilhado / Time-Sharing:

- ➔ Noção da fatia de tempo (*quantum / time-slice*);
- ➔ Sistemas conhecidos como *on-line*: baixo tempo de resposta;
- ➔ Processamento seqüencial dos processos;
- ➔ Processamento baseado em **cooperação** / preempção;
- ➔ Tempo de resposta depende da carga do sistema.

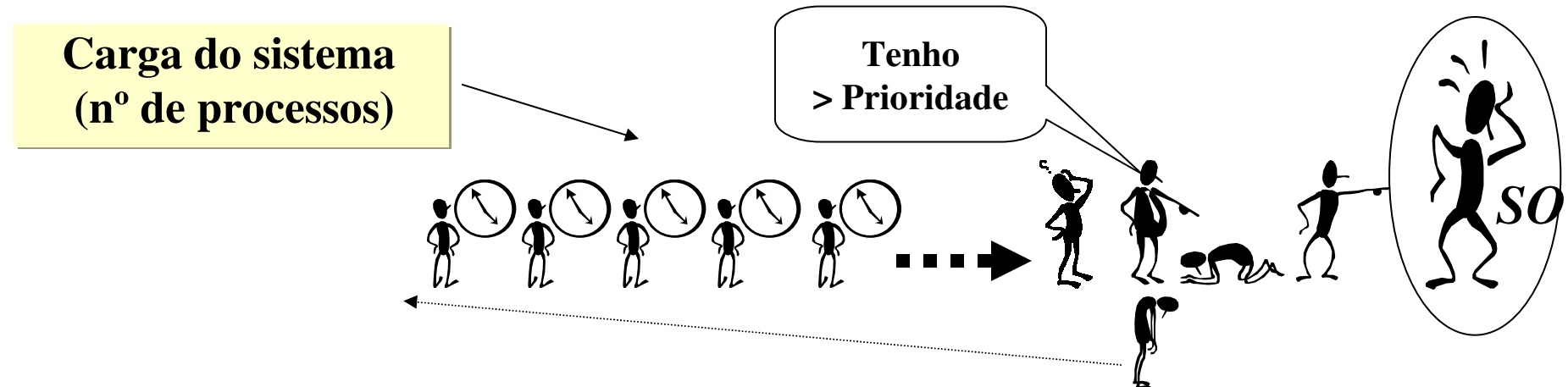
Troca de msg's

~~Time slice~~



Tempo-Real / Real-Time:

- ➡ Tempo de máximo para resposta (*deadline*) de cada processo é conhecido previamente;
- ➡ Sistemas customizados para aplicações específicas;
- ➡ Processamento baseado prioridades;
- ➡ Tempo de resposta independe da carga do sistema.



➡ Revisão de conceitos básicos de Sistemas Operacionais; 😊

➡ Evolução histórica;

➡ Definições sobre Sistemas Distribuídos (SD);

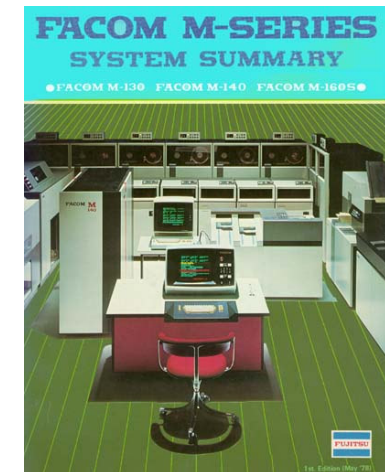
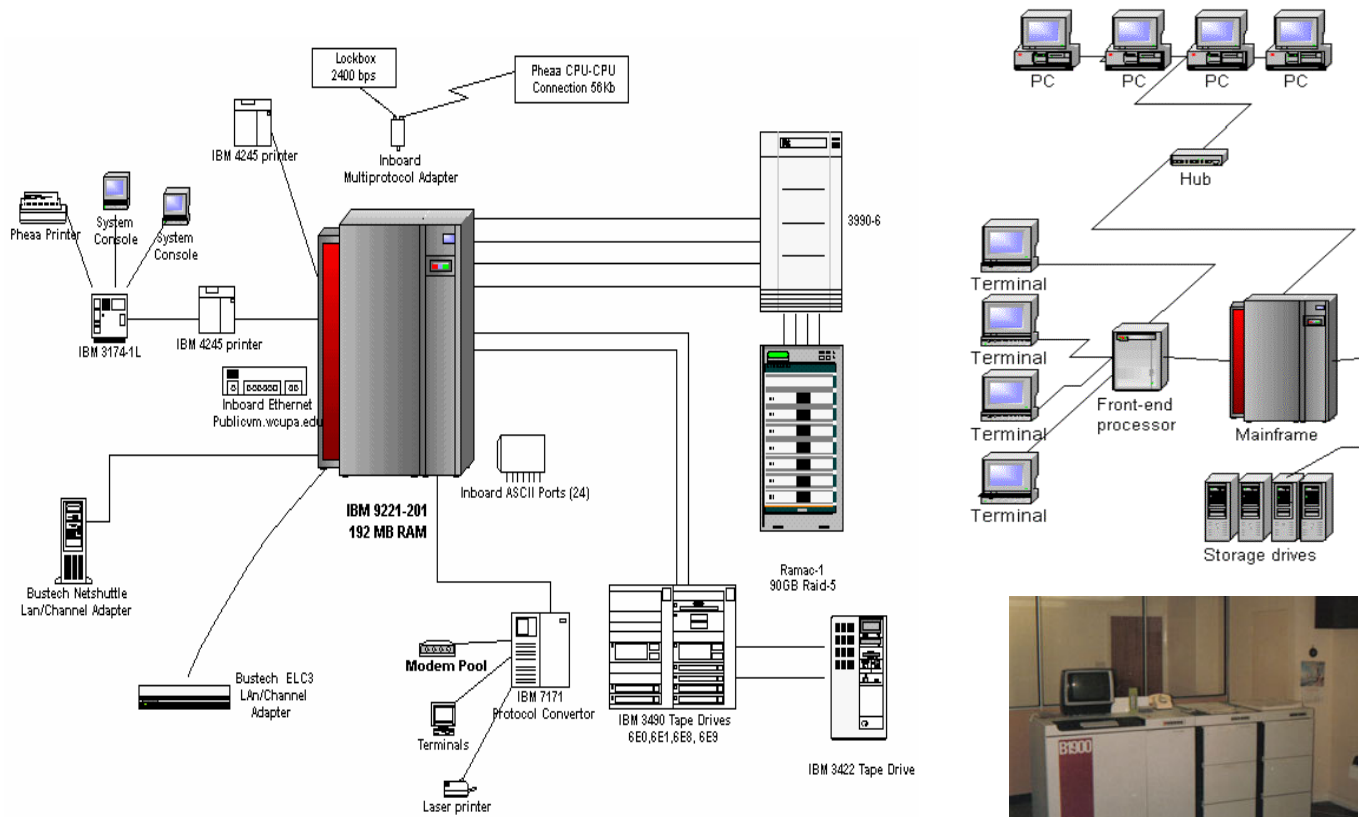
➡ Vantagens e desvantagens de SD;

➡ Desafios / oportunidades em SD.

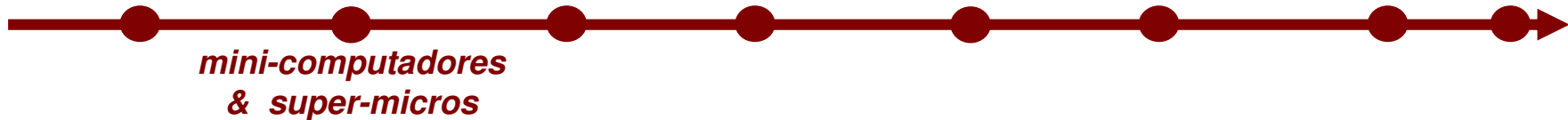


- Primeiros SO em computação científica e comercial;
- Atualmente, grande volume de dados;
- Arquitetura de hardware e infra-estrutura de instalação complexa;
- Processamento centralizado;
- Trabalham em time-sharing ou batch;
- Ex: IBM, Unisys-Burroughs, Fujitsu.

Mainframe

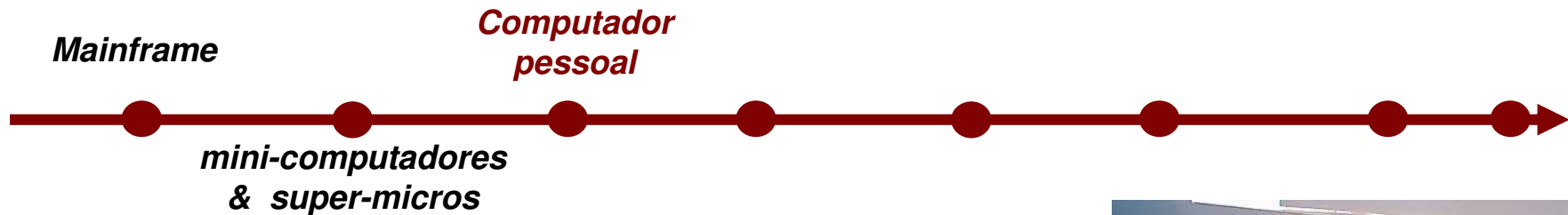


Mainframe



- Processamento centralizado;
- Alternativa de menor custo aos mainframes;
- Atualmente, comparado aos servidores;
- Ex: Cobra, Medidata, Labo, Sisco, Edisa.





- Mudanças de paradigmas;
- Usuário final com cultura tecnológica;
- Explosão de aplicativos e interface gráfica;
- Popularização da informática;
- Adotou tecnologias de SO desenvolvida para mainframes;
- Bom tempo de resposta a baixo custo;
- Arquitetura de hardware e infra-estrutura para instalação simplificada;
- Ex: HP, Compaq, Dell, Microtec, Itaotec, Monydata.

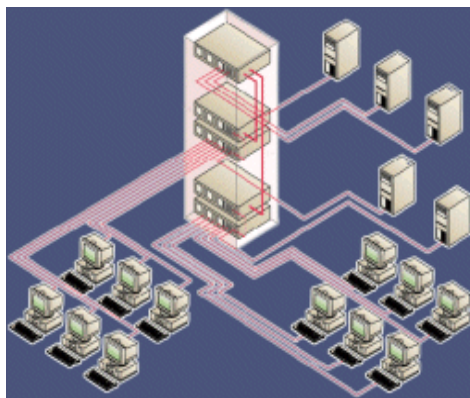
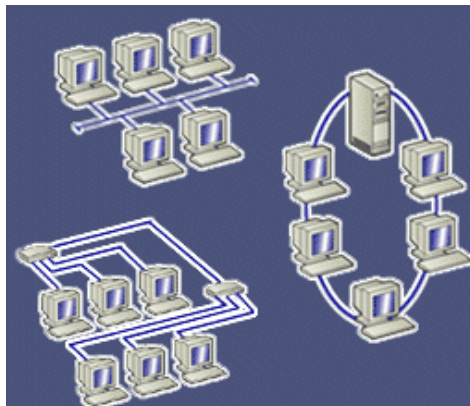


Mainframe

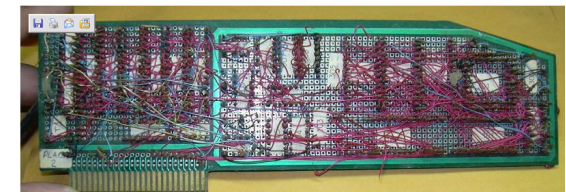
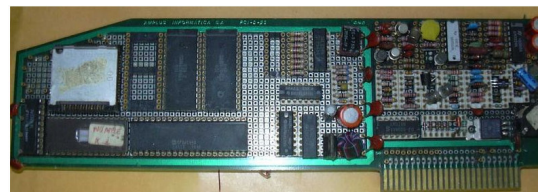
Computador
pessoal

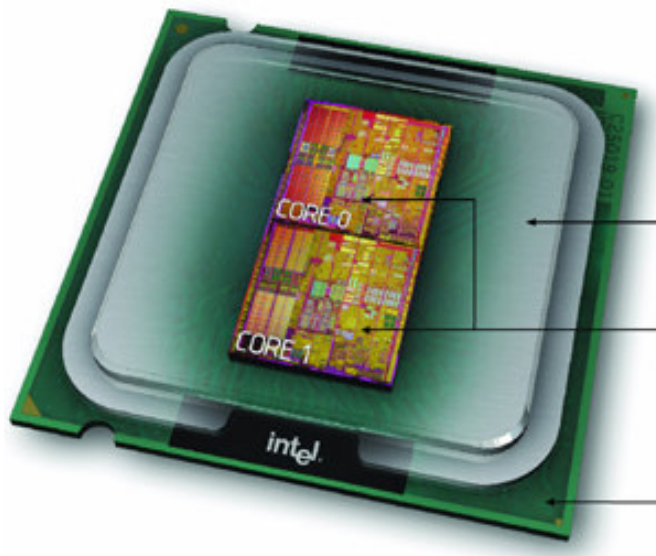
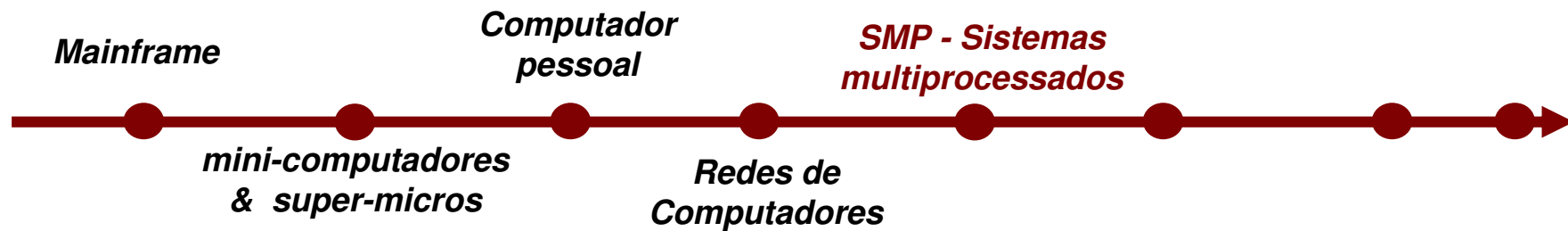
mini-computadores
& super-micros

Redes de
Computadores



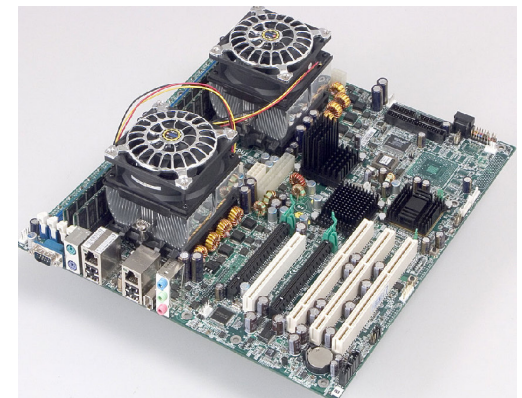
- Compartilhamento de recursos computacionais;
- Downsizing tecnológico;
- **Sistemas fracamente acoplado:**
 - Cada nodo da rede tem o seu próprio SO, memória e relógio;
 - “Relativa transparência” de hardware e software;
 - Infra-estrutura de comunicação e conversões → troca de mensagens;
- Ex: NetWare, LanManager, Família Unix, Banyan Vines, Amplus, W2K, ...



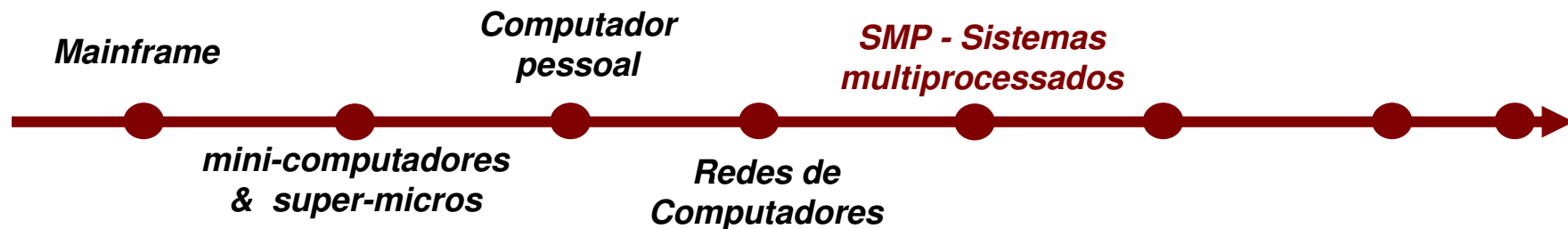


Processador com 2 núcleos (dual core).

- ⇒ Alto desempenho nos desktop (paralelismo);
- ⇒ Maior poder as sistemas servidores (paralelismo);
- ⇒ Cliente magro (*thin client*);
- ⇒ Tendência a computação centralizada;
- ⇒ SO's modernos suportam SMP;
- ⇒ Aumento da vazão (*throughput*);
- ⇒ Economia de escala;
- ⇒ Confiabilidade;
- ⇒ ex: W2k, Linux, Solaris, Unix, ...

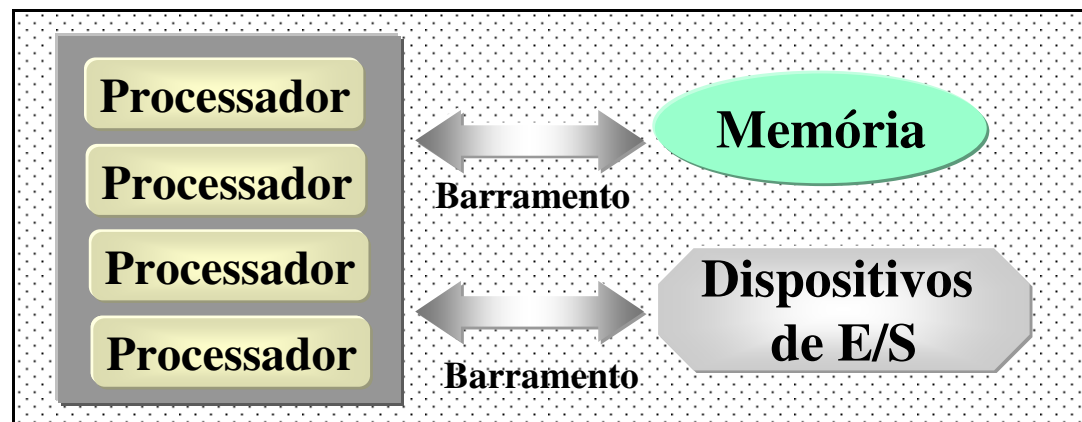


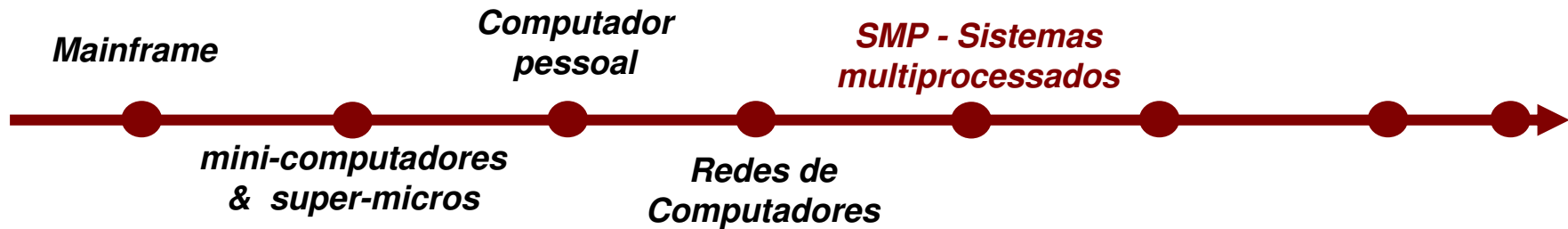
Sistema com 2 processadores.



Sistemas Fortemente Acoplados

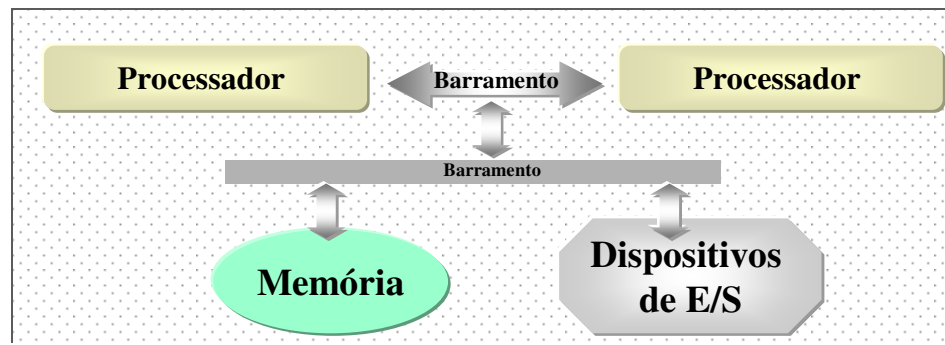
- Processadores compartilham memória (*Data Shared Memory – DSM*);
- Processadores conectados através de barramento comum;
- Relógio (*clock*) único;
- Controlados por um único SO;
- Processamento voltado para a resolução de um único problema.

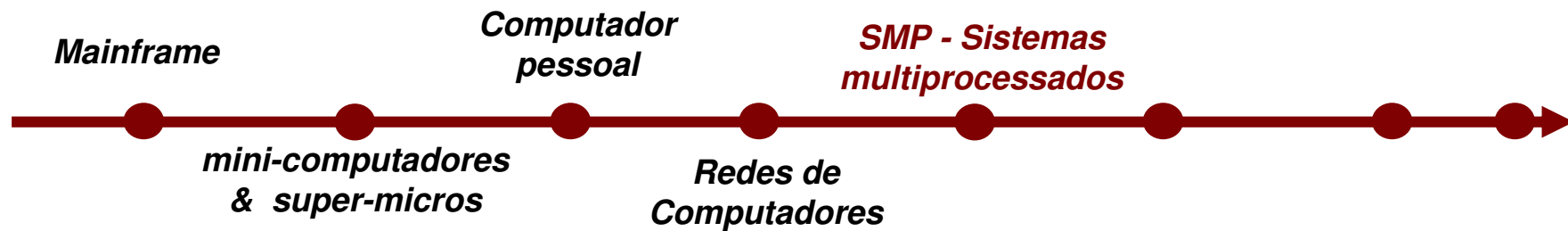




Sistemas Fortemente Acoplados

- Classificados de acordo com a **simetria** da execução dos processos pelos processadores:
- **Simétricos:**
 - Todos os processadores podem realizar todos os tipos de processos;
 - Acessos simultâneos à memória (*DSM*);
 - Solução mais complexa que o assimétrico: melhor reconfiguração e tolerância à falhas;



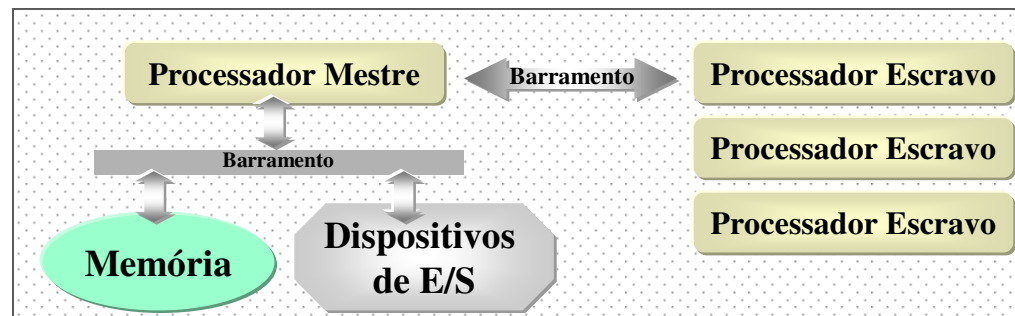


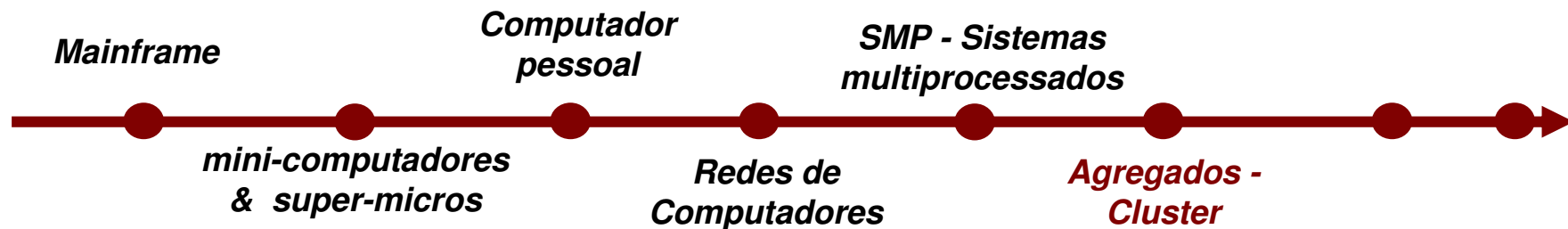
Sistemas Fortemente Acoplados:

⇒ Classificados de acordo com a **simetria** da execução dos processos pelos processadores:

⇒ Assimétricos:

- ⇒ Organização mestre/escravo;
- ⇒ Processador mestre executa as transações do SO;
- ⇒ Processadores escravos executam processos de usuários;
- ⇒ Problemas de sobrecarga ou falhas no processador mestre.

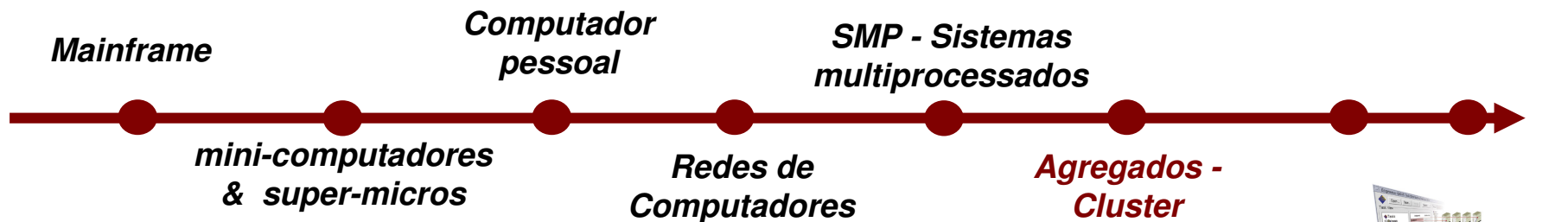




Sistema fracamente acoplado:

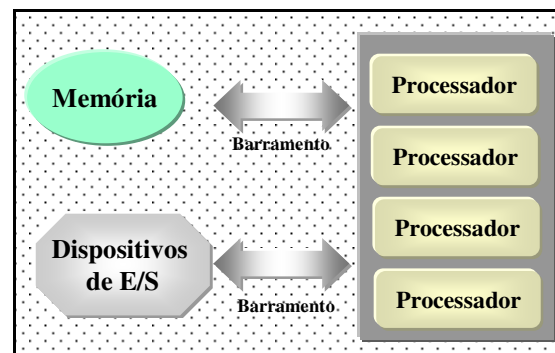
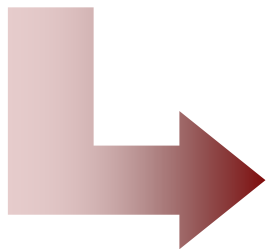
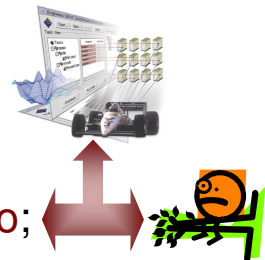
- Cada nodo de execução tem SO (cópia idêntica), memória e relógio;
- Conexões de alto desempenho (fast, gigabit, myrinet – TCP, Cliente/Servidor);
- Aplicações críticas (alta disponibilidade);
- Aplicações com demanda de processamento;



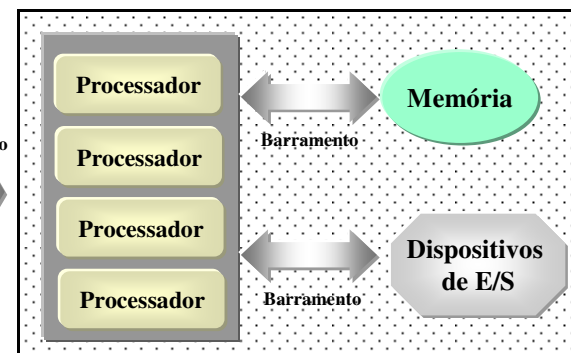


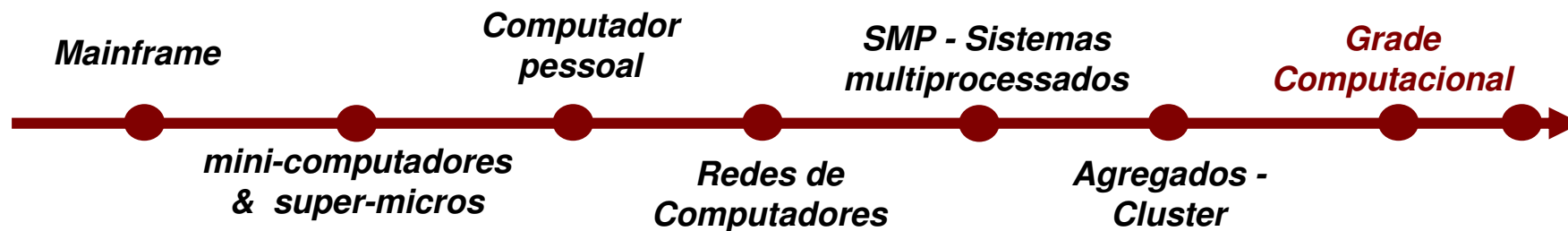
Diferente de Rede de Computadores:

- Processamento **paralelo** e **colaborativo/cooperativo**;
- Visão única do sistema:
 - Compartilhamento de recursos;
 - Balanceamento de carga;
 - Redundância → reconfiguração, tolerância à falhas.



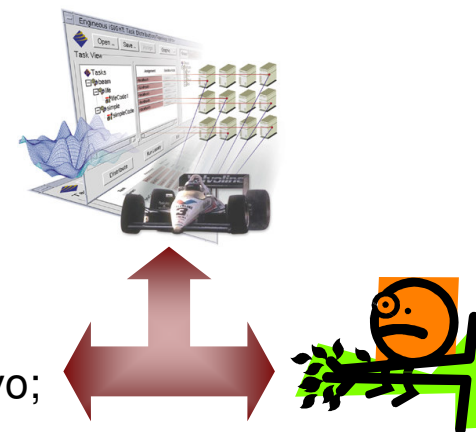
Sistema de Comunicação
Troca de mensagens

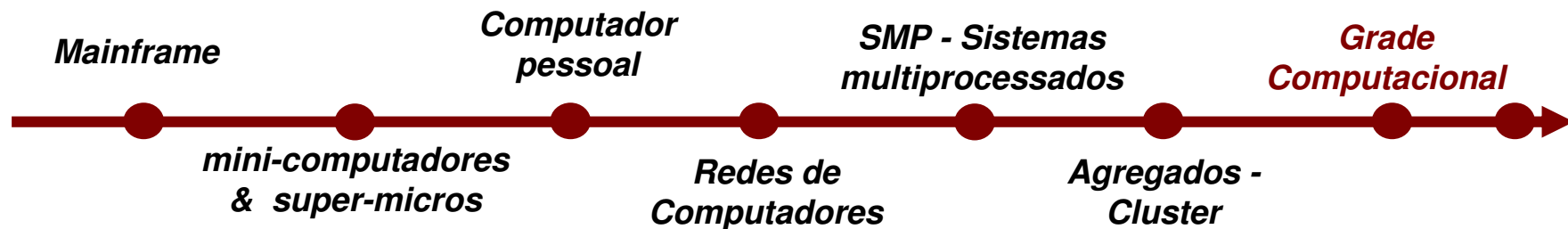




Sistema fracamente acoplado:

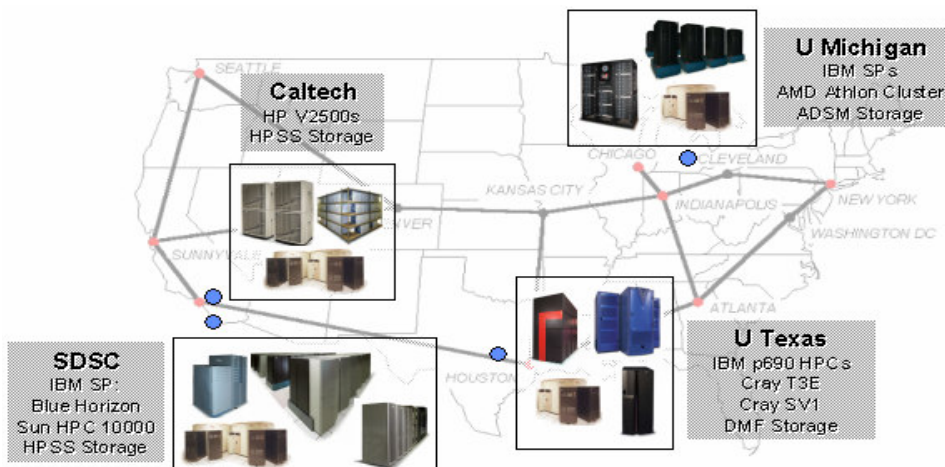
- Processamento paralelo e cooperativo;
- Diferentes proprietários dos recursos;
- Mecanismos de descoberta de recursos;
- Conceito de Organização Virtual (VO-Virtual Organization).





Sistema fracamente acoplado:

- Usuários “enxergam” os recursos distribuídos como se fossem centralizados (**Portal**);
- Provedores e consumidores de recursos;
- Grandes distâncias geográficas;
- Recursos computacionais heterogêneos.




**Centro Nacional de Supercomputação
CESUP - RS**

Bem-Vindo | Grid | Portlets GT4

Credenciais Recursos Arquivos Jobs

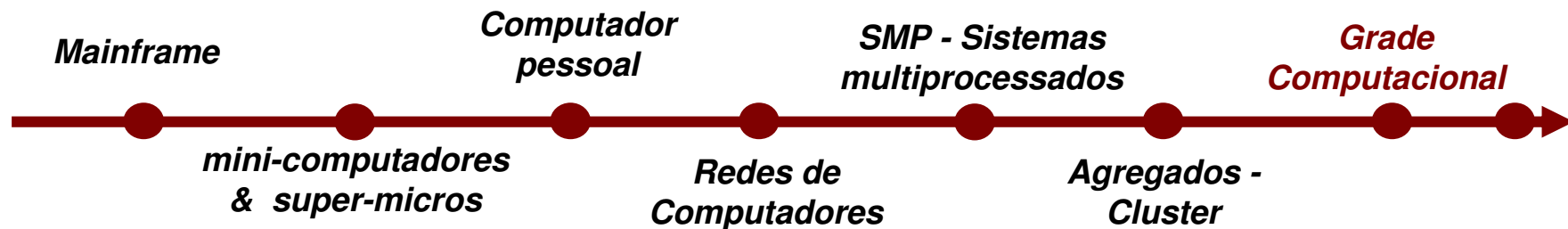
Portlet Browser de Recursos

Resources Services Job Queues Jobs Accounts

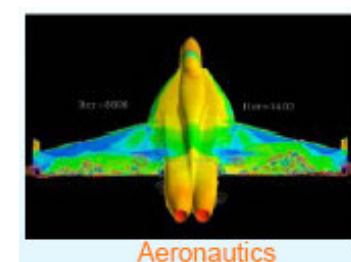
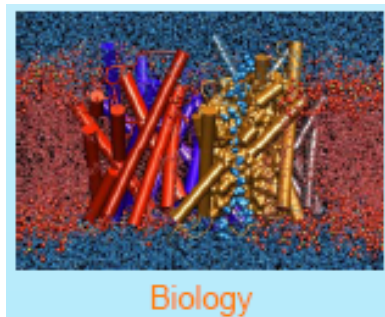
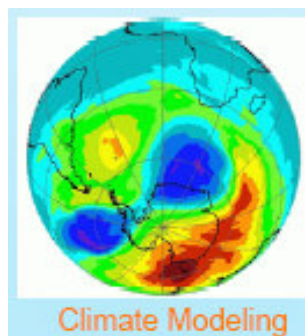
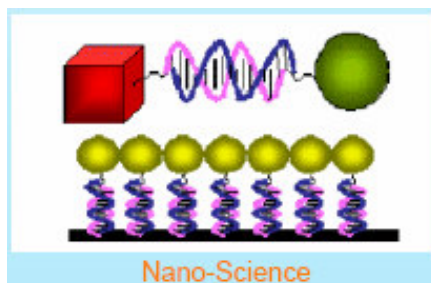
Refresh List Mds2 Change Profile

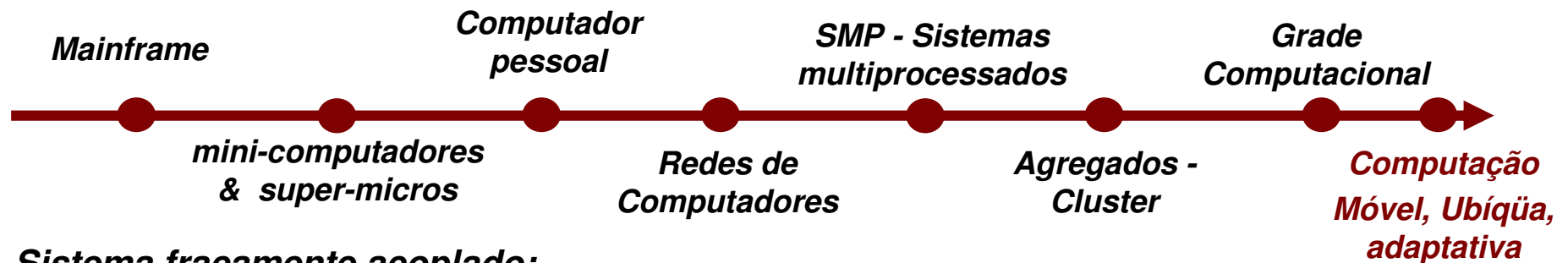
Host List

Resource	Hostname	Platform
Pascal	pascal.cesup.ufrgs.br	
Pople	pople.iq.ufrgs.br	i686 AMD Duron(tm) Processor
Pcpl8	pcpl8.if.ufrgs.br	i686 AMD Athlon(tm) Processor
Grade	grade.mat.ufrgs.br	i686 AMD Sempron(tm)



Aplicações características:



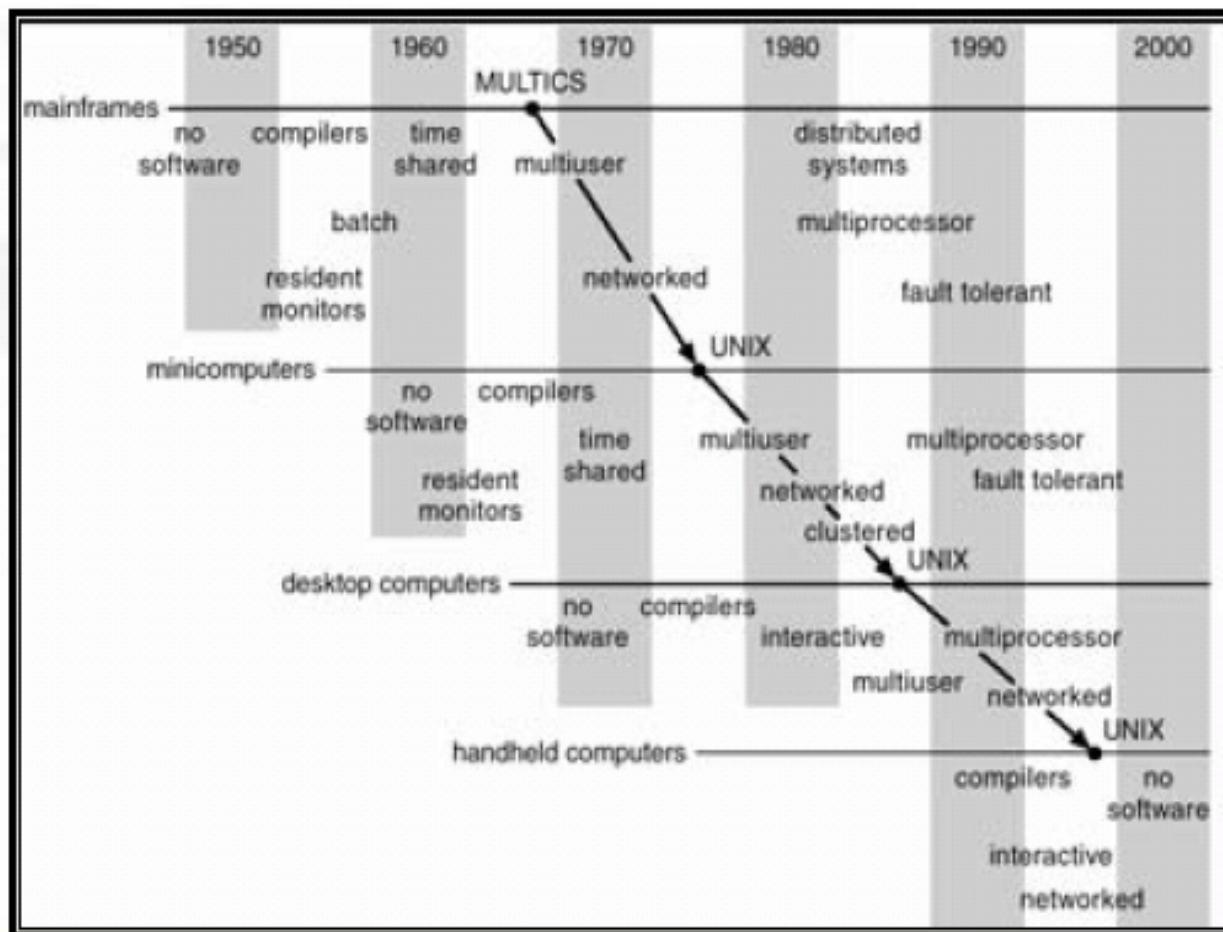
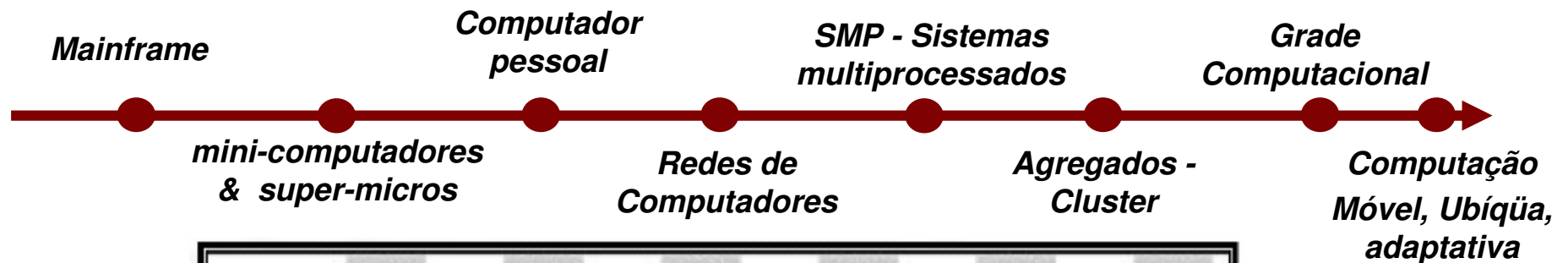


Sistema fracamente acoplado:

⇒ SO's adaptados para dispositivos móveis, como PDA (*Personal Digital Assistents*) e telefones celulares:

- ⇒ Memória limitada, processador mais lento, display de pequenas dimensões;
- ⇒ SO e aplicações projetadas para minimizar o uso de bateria;
- ⇒ Comunicação sem fio & alta conectividade (MP3, Cameras, cartões diversos,...);
- ⇒ Miniaturização (redes de sensores, computadores vestíveis, etc);
- ⇒ Aplicações nômades, adaptativas, embarcadas.





⇒ Revisão de conceitos básicos de Sistemas Operacionais; 😊

⇒ Evolução histórica; 😊

⇒ **Definições sobre Sistemas Distribuídos (SD);**

⇒ Vantagens e desvantagens de SD;

⇒ Desafios / oportunidades em SD.



Definições de SD

Faculdades SENAC-RS

Sistemas Distribuídos

Lincoln L. de Moraes

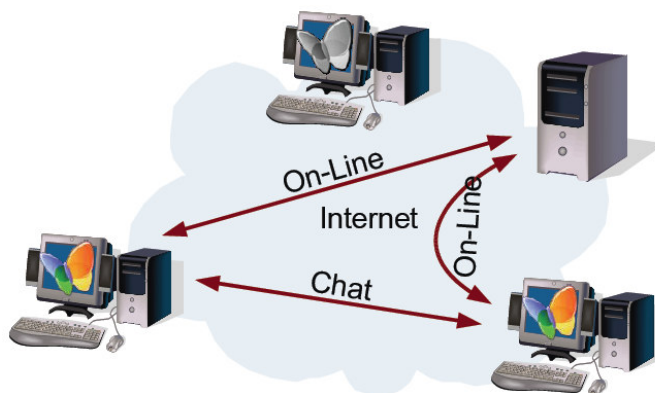
⇒ ***“Um sistema distribuído são vários computadores fazendo algo juntos” [Michael Schröder]***

- ⇒ ***“Um sistema distribuído são vários computadores fazendo algo juntos” [Michael Schröder]***
- ⇒ ***“Um sistema é distribuído quando a falha de um computador do qual você nunca ouviu falar o impede de fazer qualquer coisa” [Leslie Lamport]***

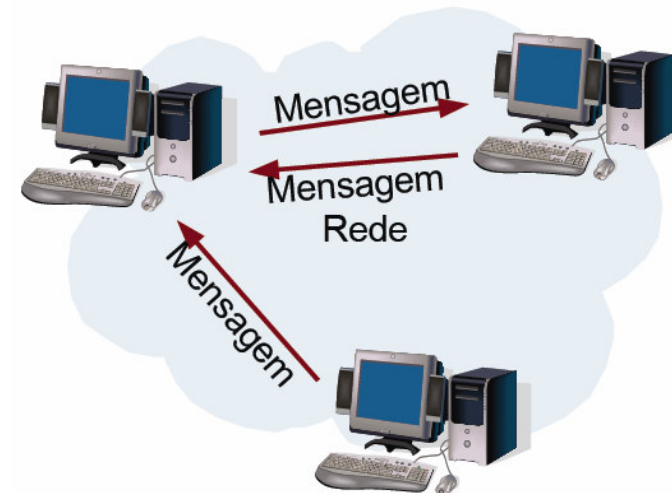
- ⇒ ***“Um sistema distribuído são vários computadores fazendo algo juntos” [Michael Schröder]***
- ⇒ ***“Um sistema é distribuído quando a falha de um computador do qual você nunca ouviu falar o impede de fazer qualquer coisa” [Leslie Lamport]***
- ⇒ ***“Conjunto de CPU interconectadas por meio de uma rede de comunicação” [Tanenbaum]***

- ***“Um sistema distribuído são vários computadores fazendo algo juntos” [Michael Schröder]***
- ***“Um sistema é distribuído quando a falha de um computador do qual você nunca ouviu falar o impede de fazer qualquer coisa” [Leslie Lamport]***
- ***“Conjunto de CPU interconectadas por meio de uma rede de comunicação” [Tanenbaum]***
- ***“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” [Colouris, Dolimore, Kindberg]***

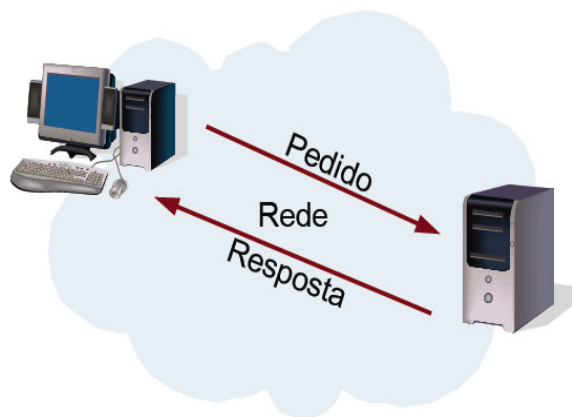
Internet e Intranets



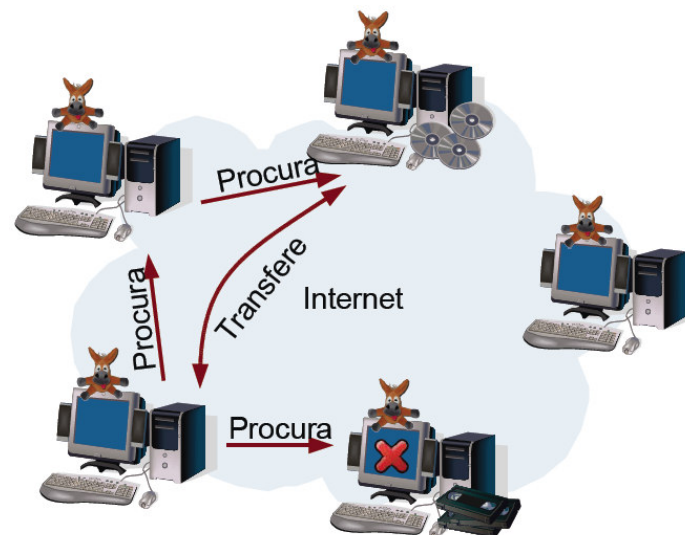
Mensagens instantâneas



Correio eletrônico



WWW (World-Wide Web)



Transferência de arquivos



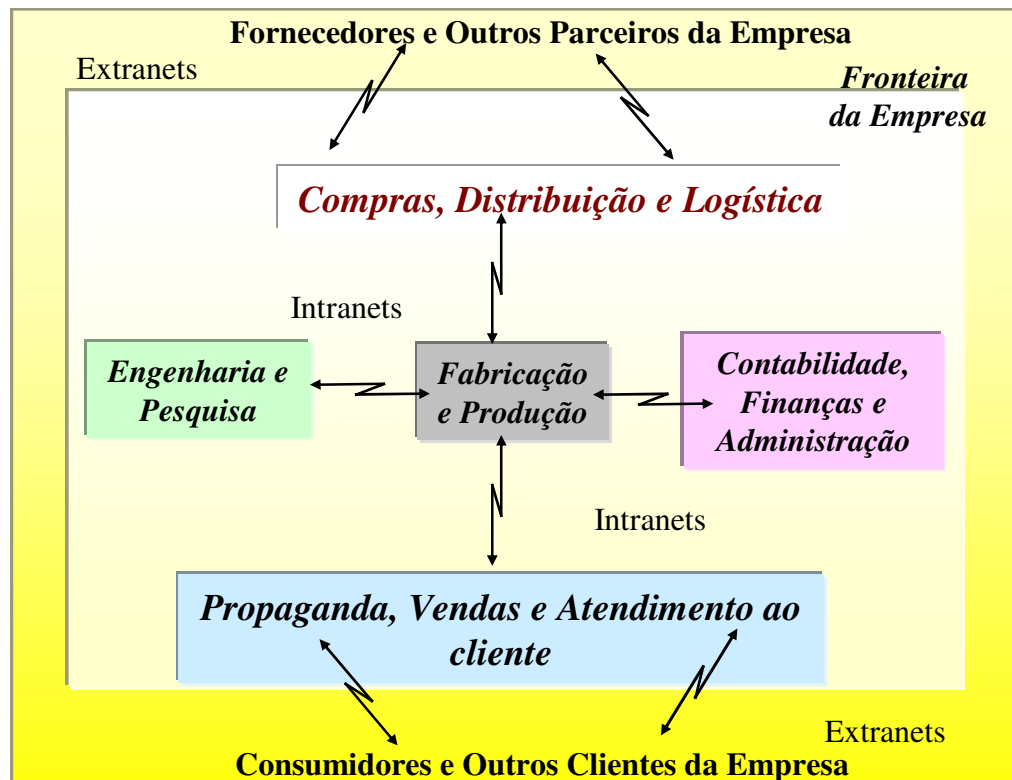
Internet e Intranets

- Correio eletrônico;
- WWW (World-Wide Web);
- Transferência de arquivos;
- Mensagens Instantâneas.

Requisitos

- Alta extensibilidade e escalabilidade;
- Mecanismos de resolução de nomes:
 - zebeleza[@yahoo.com.br](mailto:zebeleza@yahoo.com.br);
 - www.zebeleza.com.br.
- Esquemas de endereçamento e roteamento;
- Controle de acesso e segurança.

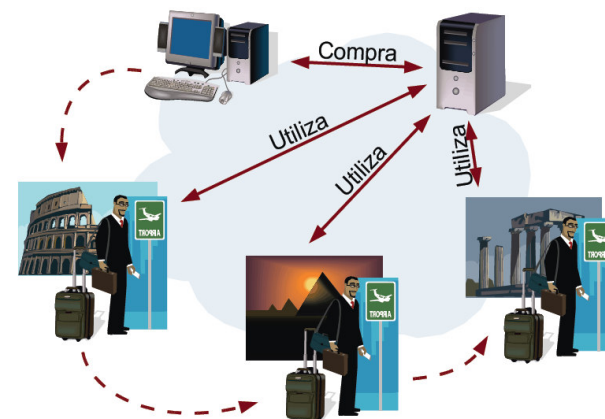
Aplicações Comerciais



**Controle de estoque, entregas e vendas
(Supply Chain Management).**



Sistema bancário



Sistema de reservas de passagens



Aplicações Comerciais

1. Sistema de reservas de passagens;
2. Sistema bancário (interconexão de agências, terminais bancários);
3. Controle de estoque, entregas e vendas (*Supply Chain Management*).

Requisitos

- ➡ Alta grau de confiabilidade;
- ➡ Alto grau de segurança e privacidade de informações;
- ➡ Suporte à concorrência de usuários;
- ➡ Garantia de tempos de respostas satisfatórios;
- ➡ Suportar massiva distribuição e extensibilidade;
- ➡ Suporte a sistemas operados por diferentes organizações.



Exemplo de SD e seus requisitos

Faculdades SENAC-RS

Sistemas Distribuídos

Lincoln L. de Moraes

Computação Móvel - integração de dispositivos miniaturizados e portáteis ao SD. Classificada em:

⇒ ***Computação Nômade – Nomadic Computing:***

⇒ Execução de tarefas enquanto o usuário se movimento – movimentação do hardware;



Exemplo de SD e seus requisitos

Faculdades SENAC-RS

Sistemas Distribuídos

Lincoln L. de Moraes

Computação Móvel - integração de dispositivos miniaturizados e portáteis ao SD. Classificada em:

⇒ **Computação Nômade – Nomadic Computing:**

⇒ Execução de tarefas enquanto o usuário se movimento – movimentação do hardware;

⇒ **Computação Sem Fio – Wireless Computing:**

⇒ Usuário se movimenta conectado ao um conjunto fixo de estações conectadas à rede;



Exemplo de SD e seus requisitos

Faculdades SENAC-RS

Sistemas Distribuídos

Lincoln L. de Moraes

Computação Móvel - integração de dispositivos miniaturizados e portáteis ao SD. Classificada em:

➤ **Computação Nômade – Nomadic Computing:**

➤ Execução de tarefas enquanto o usuário se movimento – movimentação do hardware;

➤ **Computação Sem Fio – Wireless Computing:**

➤ Usuário se movimenta conectado ao um conjunto fixo de estações conectadas à rede;

➤ **Computação Adaptativa – Adaptive Computing:**

➤ A aplicação se movimenta;

➤ Utiliza informações do ambiente, construindo modelos computacionais dinamicamente;



Exemplo de SD e seus requisitos

Faculdades SENAC-RS

Sistemas Distribuídos

Lincoln L. de Moraes

Computação Móvel - integração de dispositivos miniaturizados e portáteis ao SD. Classificada em:

⇒ **Computação Nômade – Nomadic Computing:**

- ⇒ Execução de tarefas enquanto o usuário se movimento – movimentação do hardware;

⇒ **Computação Sem Fio – Wireless Computing:**

- ⇒ Usuário se movimenta conectado ao um conjunto fixo de estações conectadas à rede;

⇒ **Computação Adaptativa – Adaptive Computing:**

- ⇒ A aplicação se movimenta;
- ⇒ Utiliza informações do ambiente, construindo modelos computacionais dinamicamente;

⇒ **Computação ubíqua – Ubiquitous Computing:**

- ⇒ Computação em larga escala;
- ⇒ Computador inserido no ambiente de forma invisível;
- ⇒ Todos os elementos do sistema de computação tem a propriedade de mobilidade.
- ⇒ Também chamada de computação pervasiva ou *pervasive computing*.



Exemplo de SD e seus requisitos

Faculdades SENAC-RS

Sistemas Distribuídos

Lincoln L. de Moraes

Computação Móvel - integração de dispositivos miniaturizados e portáteis ao SD. Classificada em:

Requisitos

- ➡ Suporte transparente à mobilidade;
- ➡ Tratamento de contexto;
- ➡ Otimização de espaço de armazenamento, largura de banda, consumo de energia;
- ➡ Tratamento de conteúdo adequado aos recursos momentâneos;
- ➡ Formatação, apresentação, compressão, entrega e armazenamento adaptável à largura de banda e recursos do dispositivo.

Aplicações Multimídias e Teleconferências

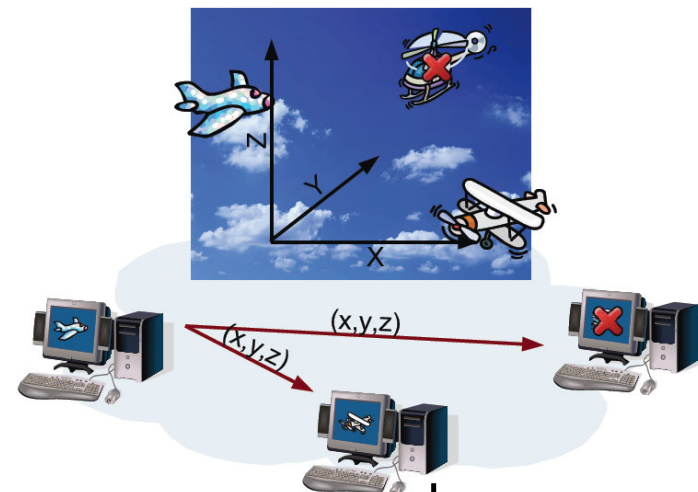
SETI@HOME



Sistema de suporte ao trabalho cooperativo



Sistema de suporte a EaD



Jogos em rede



Aplicações Multimídias e Teleconferências

- ➡ Sistema de suporte a Educação à Distância;
- ➡ Sistema de suporte ao trabalho cooperativo;
- ➡ Jogos em rede.

Requisitos

- ➡ Suporte ao trabalho cooperativo (comunicação confiável, suporte a grupos, autenticação, controle de versões, etc);
- ➡ Suporte para sincronização de canais de transmissão (ex: imagem e som);
- ➡ Garantias de qualidade de serviços:
 - ➡ Atrasos máximos;
 - ➡ Taxas de transmissão;
 - ➡ Restrições de tempo.

Resumindo SD...

- ⇒ Conjunto de máquinas autônomas;
- ⇒ Interconectadas por canais de comunicação;
- ⇒ Comunicando-se através de mensagens;
- ⇒ Ausência de um estado global;
- ⇒ Independência de falhas;
- ⇒ Ausência de sincronização de relógios (geral);
- ⇒ Compartilhamento de recursos (software e/ou hardware).

Implicações de SD no desenvolvimento de aplicações?



Mudança de paradigmas

<i>Antes</i>	<i>Depois</i>
Controle Central	Controle Distribuído
Nomeação Global	Nomeação Federada
Consistência Global	Consistência Fraca
Execução seqüencial	Execução Paralela
Vulnerabilidade a falhas	Tolerância a falhas
Homogeneidade	Heterogeneidade
Expansibilidade cara e limitada	Expansibilidade ilimitada (teoricamente)
Informação local	Informação remota
Localização fixa	Migração

SISTEMAS FRACAMENTE ACOPLADOS.

➡ Revisão de conceitos básicos de Sistemas Operacionais; 😊

➡ Evolução histórica; 😊

➡ Definições sobre Sistemas Distribuídos (SD) 😊

➡ ***Vantagens e desvantagens de SD;***

➡ Desafios / oportunidades em SD.

CONTEXTO ATUAL

- ➡ É relativamente fácil agrupar um grande número de CPUs, conectando-as por uma rede de alta velocidade.
- ➡ O software para sistemas distribuídos é completamente diferente do software para sistemas centralizados e está apenas começando a se desenvolver.







- 👍 ***Economia***: melhor relação custo x benefício quanto ao desempenho (supercomputador virtual);
- 👍 ***Eficiência***: maior poder total de computação;
- 👍 ***Distribuição de recursos***: máquinas geograficamente separadas;
- 👍 ***Confiabilidade e Disponibilidade***: se uma máquina falha, o sistema permanece funcional;
- 👍 ***Normalização***: necessária à integração;



- 👍 **Escalabilidade**: poder computacional adicionado em incrementos;
- 👍 **Aplicações modulares**: domínios maiores, baixo acoplamento e reaproveitamento;
- 👍 **Flexibilidade**: diferentes plataformas podem ser integradas;
- 👍 **Compartilhamento de recursos**: computação como commodities (discos, processadores, memória, software);
- 👍 **NOVOS MERCADOS.**

Aplicação:

-  Pouca disponibilidade de ferramentas de suporte (ex:compiladores, middleware);
-  Deve ser bem planejada para obter os benefícios propostos (concorrência, consistência, multithread, sincronismo, comunicação);
-  Conceitos e sua utilização em fase de amadurecimento;
-  Soluções mais complexas.

 **Rede:** seu estado causa alto impacto na aplicação, **podendo eliminar suas vantagens;**

 **Segurança:** várias portas de entrada (queijo suíço);

 **Falhas:** instabilidade das aplicações, modelagem e testes falhos.

⇒ **MERCADO**: especialização e segmentação de área de atuação → convergência.



⇒ Revisão de conceitos básicos de Sistemas Operacionais; 😊

⇒ Evolução histórica; 😊

⇒ Definições sobre Sistemas Distribuídos (SD) 😊

⇒ Vantagens e desvantagens de SD; 😊

⇒ ***Desafios / oportunidades em SD.***



- ⇒ ***Heterogeneidade;***
- ⇒ ***Abertura (openness);***
- ⇒ ***Segurança;***
- ⇒ ***Concorrência / Paralelismo;***
- ⇒ ***Escalabilidade;***
- ⇒ ***Tratamento de falhas;***
- ⇒ ***Transparência.***

Um SD tende a evoluir para a heterogeneidade:

- ➡ Equipamentos de diferentes vendedores;
- ➡ Diferentes aplicações para o mesmo fim;
- ➡ Diferentes versões da mesma aplicação;
- ➡ Diferentes administradores;
- ➡ Possibilidade de evolução sem troca completa do sistema.

Um SD tende a evoluir para a heterogeneidade:

- ➡ Equipamentos de diferentes vendedores;
- ➡ Diferentes aplicações para o mesmo fim;
- ➡ Diferentes versões da mesma aplicação;
- ➡ Diferentes administradores;
- ➡ Possibilidade de evolução sem troca completa do sistema.



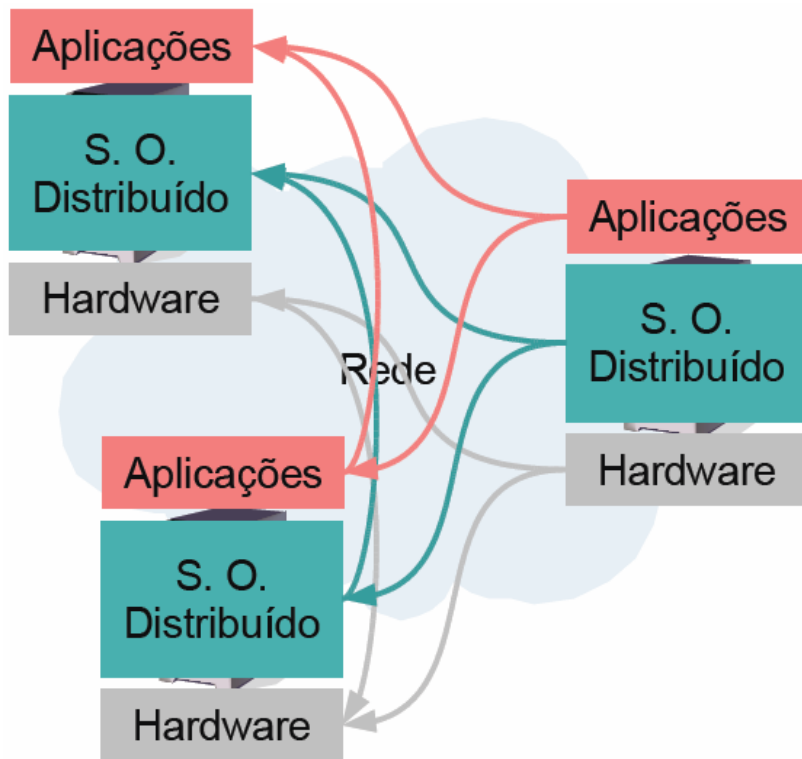
Uso de protocolos e padrões comuns (utilizados na Internet)

Ex: Myrinet (protocolo TCP/IP), GridFTP (FTP).

Middleware.

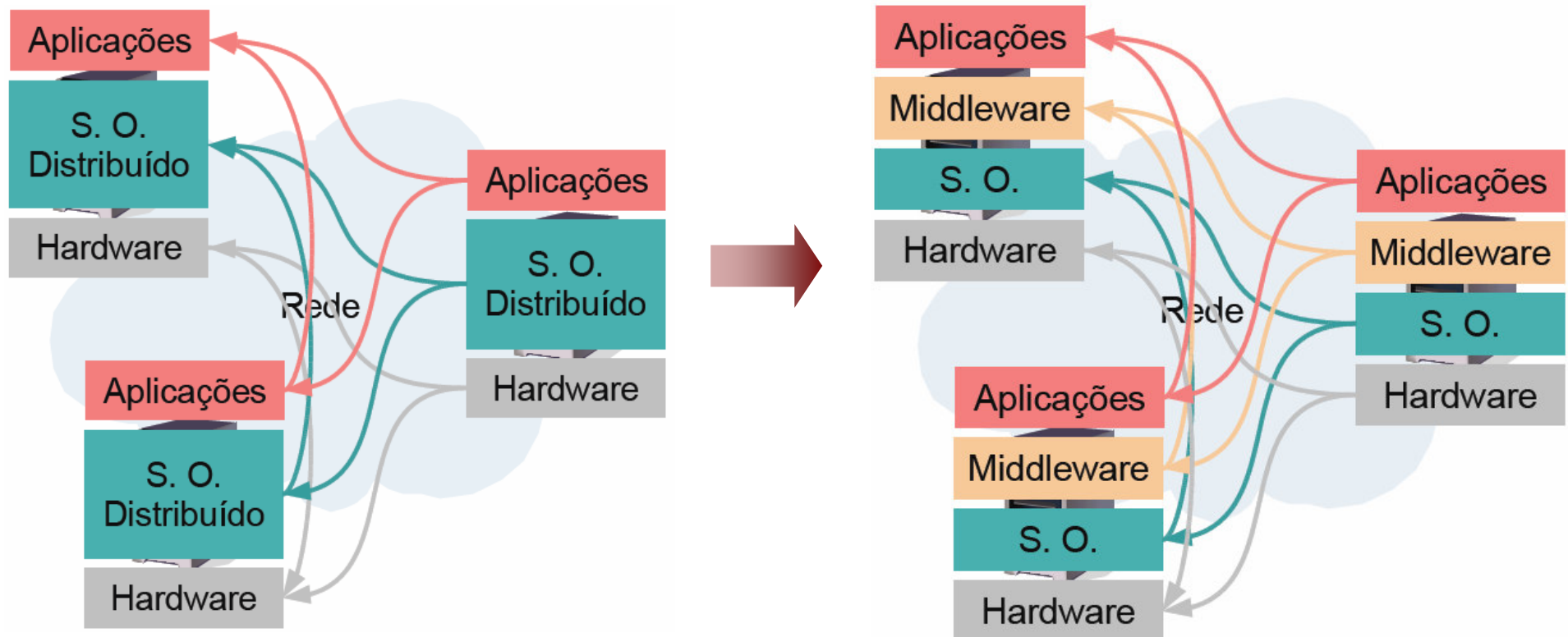
Middleware:

- ➡ Camada de software;
- ➡ Mascara a heterogeneidade do ambiente.



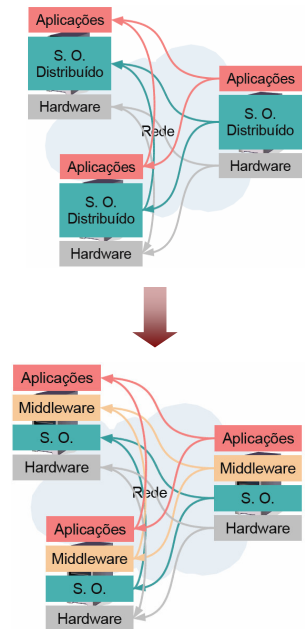
Middleware:

- ➡ Camada de software;
- ➡ Mascara a heterogeneidade do ambiente.



Middleware:

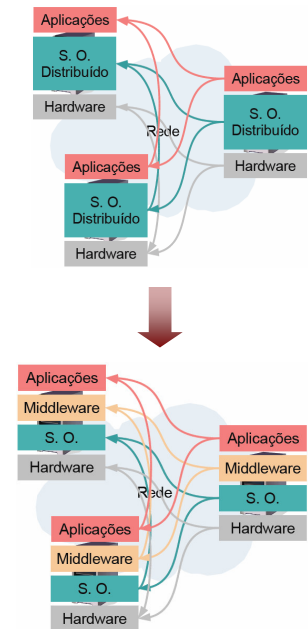
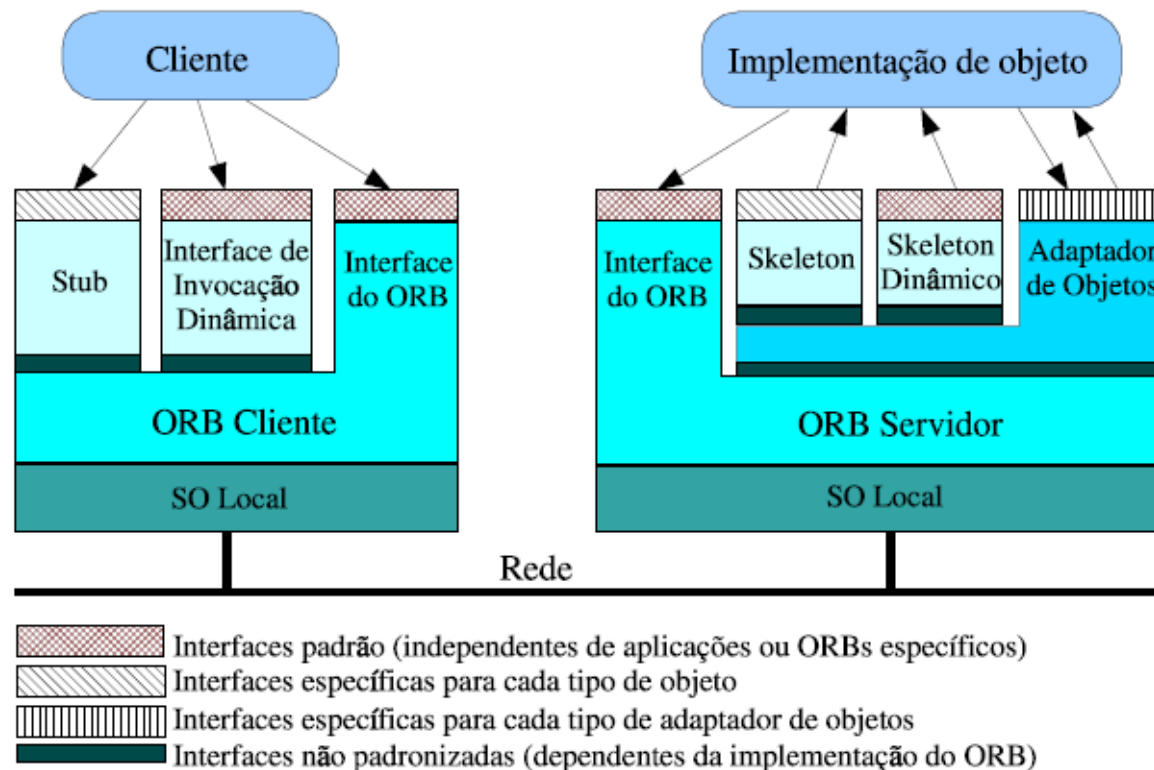
- ➡ Camada de software;
- ➡ Mascara a heterogeneidade do ambiente.
Ex: CORBA, JAVA-RMI, DCOM.



Middleware:

- ➔ Camada de software;
- ➔ Mascara a heterogeneidade do ambiente.

Ex: CORBA, JAVA-RMI, DCOM.



Capacidade de um sistema se *estender* e *interoperar*:

- ⇒ aspectos de hardware;
- ⇒ aspectos de software.



- ⇒ Especificações de interfaces → públicas e padronizadas
- ⇒ Padrões sugeridos e padrões de fato.
 - ⇒ ex: serviço WWW com protocolo HTTP, Linux, *Web Services, ambientes de cluster e grade.*

Confidencialidade:

⇒ proteção contra acesso indevido;

Integridade:

⇒ proteção contra alteração ou corrupção indevida;

Disponibilidade:

Armazenamento / transmissão segura de dados:

Confidencialidade:

⇒ proteção contra acesso indevido;

Integridade:

⇒ proteção contra alteração ou corrupção indevida;

Disponibilidade:

⇒ proteção contra impedimento de acesso;

Armazenamento / transmissão segura de dados:

⇒ criptografia;

Paralelismo

⇒ processos que executam simultaneamente;

Concorrência

⇒ processos que executam simultaneamente e concorrem por recursos.

Paralelismo

- ⇒ processos que executam simultaneamente;

Concorrência

- ⇒ processos que executam simultaneamente e concorrem por recursos.



- ⇒ Requisições concorrentes a um mesmo serviço ou recurso;
- ⇒ Vários servidores oferecendo um mesmo serviço;
- ⇒ Mesma aplicação executando em diferentes computadores.



***Interações concorrentes devem ser sincronizadas:
garantia de consistência do estado da aplicação***

Capacidade de um sistema suportar aumento de escala sem sofrer alterações estruturais.

- ➡ Filosofia do projeto: prever aumento de demanda;
- ➡ Operar efetivamente em escalas diferentes;
- ➡ Processamento independente do tamanho da rede.

Capacidade de um sistema suportar aumento de escala sem sofrer alterações estruturais.

Desafios:

- ➡ Controle de custos;
- ➡ Controle de perda de desempenho;
- ➡ Evitar gargalos;
- ➡ Prevenir escassez de elementos de software:
Ex: endereços IP na Internet.

Capacidade de um sistema suportar aumento de escala sem sofrer alterações estruturais.

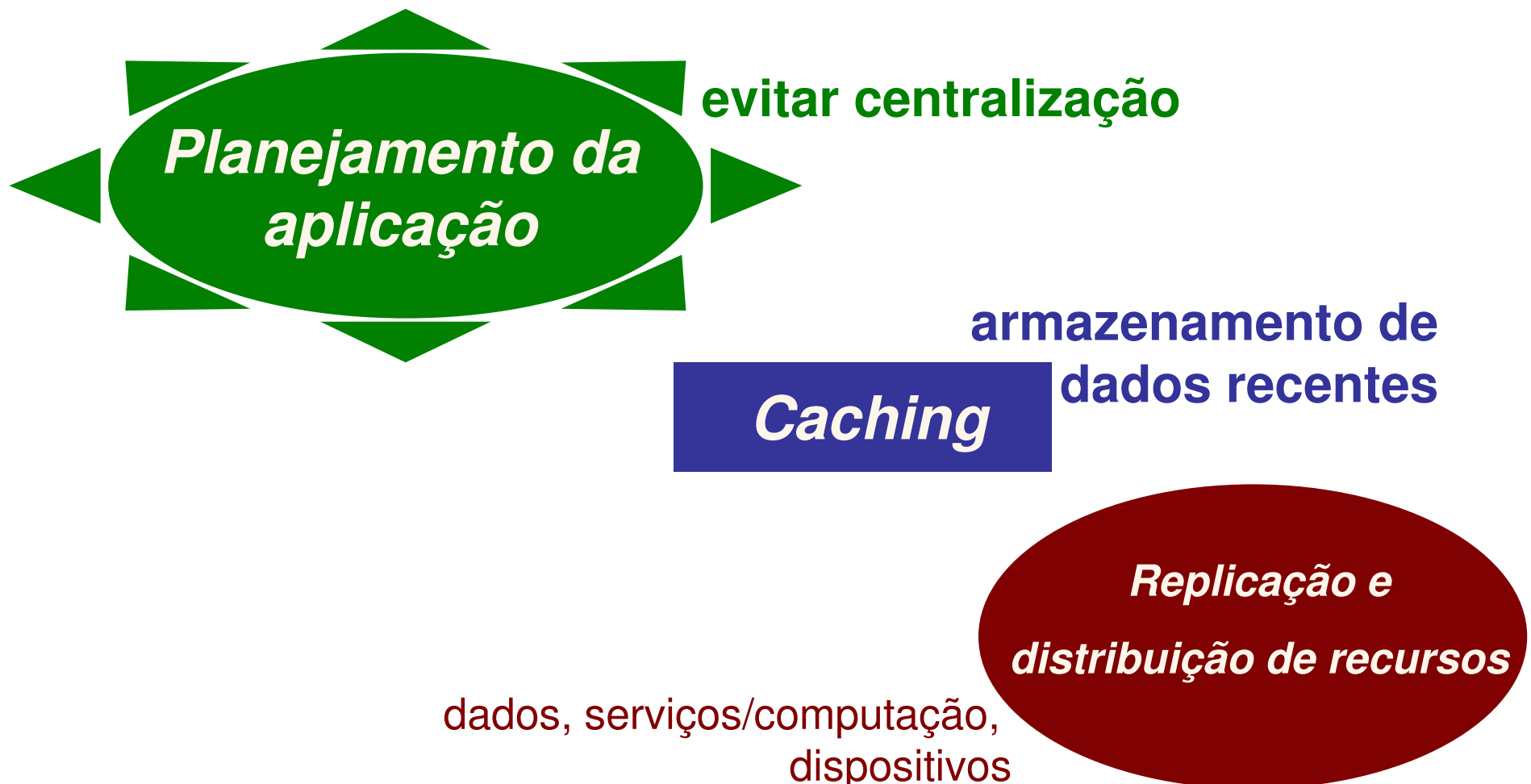
Desafios:

- ➡ Controle de custos;
- ➡ Controle de perda de desempenho;
- ➡ Evitar gargalos;
- ➡ Prevenir escassez de elementos de software:
Ex: endereços IP na Internet.

Problemas:

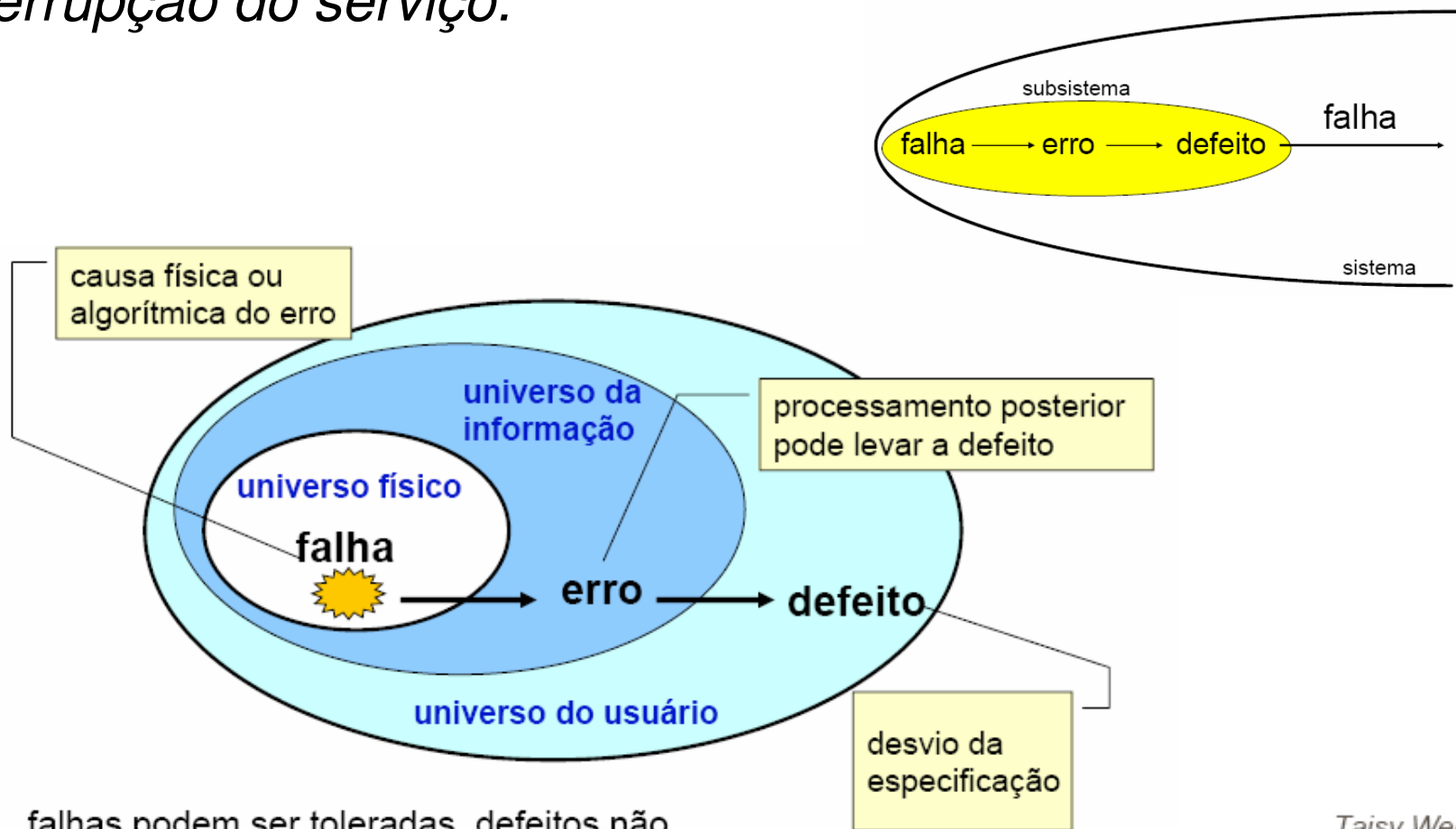
Serviços, dados e algoritmos centralizados.

Capacidade de um sistema suportar aumento de escala sem sofrer alterações estruturais.



Na ocorrência de falhas:

- ➔ *Resultados errôneos;*
- ➔ *Interrupção do serviço.*



falhas podem ser toleradas, defeitos não

Taisy Weber

Na ocorrência
de falhas

Mecanismos para garantia de funcionamento (com falhas):

⇒ **Detecção:**

- ⇒ Códigos de verificação e correções de falha (ex: checksums);
- ⇒ Impossibilidades;



A página não pode ser exibida

A página que você procura não está disponível no momento.
Talvez o site esteja passando por dificuldades técnicas ou você
precise ajustar as configurações do navegador.

Na ocorrência
de falhas

Mecanismos para garantia de funcionamento (com falhas):

⇒ **Detecção:**

- ⇒ Códigos de verificação e correções de falha (ex: checksums);
- ⇒ Impossibilidades;



A página não pode ser exibida

A página que você procura não está disponível no momento.
Talvez o site esteja passando por dificuldades técnicas ou você
precise ajustar as configurações do navegador.

⇒ **Mascaramento:**

- ⇒ Confinamento e tratamento da falha;
- ⇒ ex: retransmissão de mensagens, sistema baseados em RAID;

Na ocorrência
de falhas

Mecanismos para garantia de funcionamento (com falhas):

⇒ **Detecção;**

⇒ **Mascaramento;**

⇒ **Recuperação:**

⇒ garantia de consistência (ex: *roll back*, BCP, *checkpoints*);

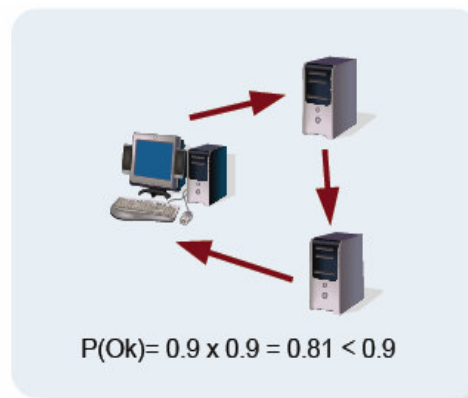
⇒ **Tolerância:**

⇒ garantia de correção e disponibilidade → através de redundância.

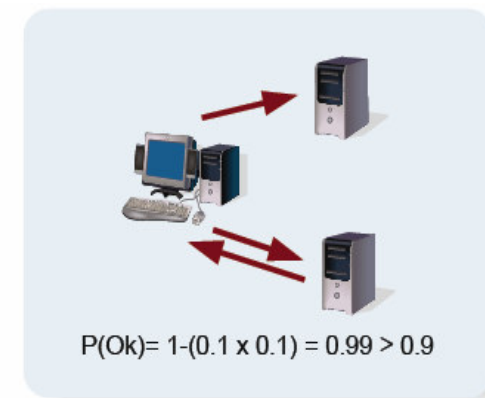
Abordagens:

- *Redundância de hardware*: servidores múltiplos;
- *Redundância de software*: manter dados permanentes sempre consistentes.

Considere um sistema com a probabilidade
 $P(\text{Falha})=0.1$, ou seja, $P(\text{OK})=0.9$.



Dependência de
componente único.



Dependência de múltiplos
componentes.

Abordagens:

- Diferentes componentes podem falhar independentemente;
- Sistema centralizado: falha de um componente afeta todo o sistema.

Componente em falha:

apenas a parte que o usa é afetada.

Componente (sw) pode ser redirecionado e reinicializado em outro computador.



No projeto prever:
recuperação do
estado anterior à
falha.

Componente em falha:

apenas a parte que o usa é afetada.

Componente (sw) pode ser redirecionado e reinicializado em outro computador.

Componente de hardware deve ser replicado:



Atividades não-críticas em ausência de falhas.

No projeto prever:
recuperação do estado anterior à falha.

Tipos de falha:

- ⇒ **Transiente:** acontece uma vez;
- ⇒ **Intermitente:** acontece periodicamente de maneira assíncrona;
- ⇒ **Permanente:** acontece sempre.

Tipos de falha:

- ⇒ **Transiente:** acontece uma vez;
- ⇒ **Intermitente:** acontece periodicamente de maneira assíncrona;
- ⇒ **Permanente:** acontece sempre.

Classificação:

⇒ **Falhas físicas:**

- ⇒ Permanentes;
- ⇒ Temporárias: intermitentes ou transitórias.

confiabilidade sempre foi um problema de engenharia, assim **falhas físicas**, que afetam diretamente o **hardware**, vem tradicionalmente recebendo atenção especial

Falhas são inevitáveis

Tipos de falha:

- ⇒ **Transiente:** acontece uma vez;
- ⇒ **Intermitente:** acontece periodicamente de maneira assíncrona;
- ⇒ **Permanente:** acontece sempre.

Classificação:

⇒ **Falhas físicas:**

- ⇒ Permanentes;
- ⇒ Temporárias: intermitentes ou transitórias.



descrição de falhas

natureza: falha de hardware, falha de software, etc..

duração: permanente ou temporária

extensão: local a um módulo, global

valor: determinado ou indeterminado no tempo

Tipos de falha:

- **Transiente:** acontece uma vez;
- **Intermitente:** acontece periodicamente de maneira assíncrona;
- **Permanente:** acontece sempre.

Classificação:

➤ **Falhas físicas:**

- Permanentes;
- Temporárias: intermitentes ou transitórias.



➤ **Falhas humanas:**

- Projeto;
- Interação: intencionais ou não-intencionais.

Security

Taisy Weber

➡ Problemas de especificação;

➡ Problemas de implementação;

➡ Componentes defeituosos:

➡ Imperfeições de manufatura;

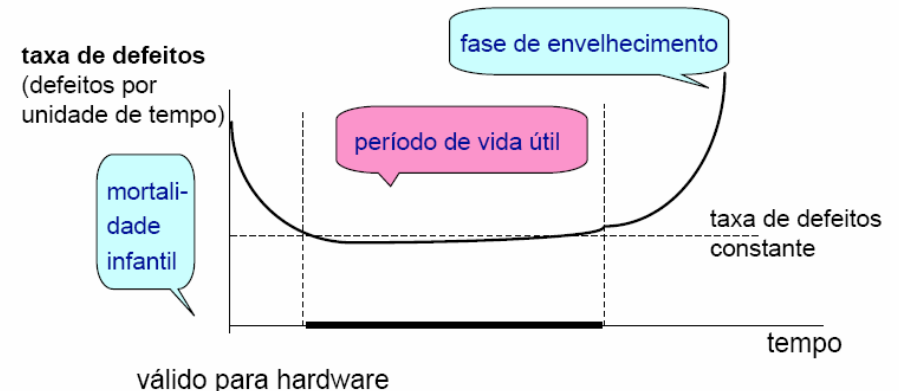
➡ Fadiga;

➡ Distúrbios externos:

➡ Radiação, interferência eletromagnética, variações ambientais (temperatura, pressão, umidade), problemas de operação.

Curva da banheira

fases de mortalidade infantil e envelhecimento muito pequenas comparadas ao período de vida útil



Taisy Weber

Taisy Weber

Causas de defeitos

Dependability of Computer Systems:
from Concepts to Limits
Jean-Claude Laprie - DCIA 98

Sistemas tradicionais		Redes cliente-servidor
Não tolerante a falhas	Tolerante a falhas	(não tolerantes a falhas)
MTBF: 6 a 12 semanas Indisponibilidade após defeito: 1 a 4 h Defeitos: hardware 50% software 25% operações 10% comunicações 15% / ambiente	MTBF: 21 anos (Tandem) Defeitos: software 65% operações 10% hardware 8% ambiente 7%	Disponibilidade média: 98% Defeitos: projeto 60% operações 24% físicos 16%

Causas usuais de defeitos em sistemas de computação

<http://www.cs.wits.ac.za/research/workshop/ifip98.html>

Taisy Weber

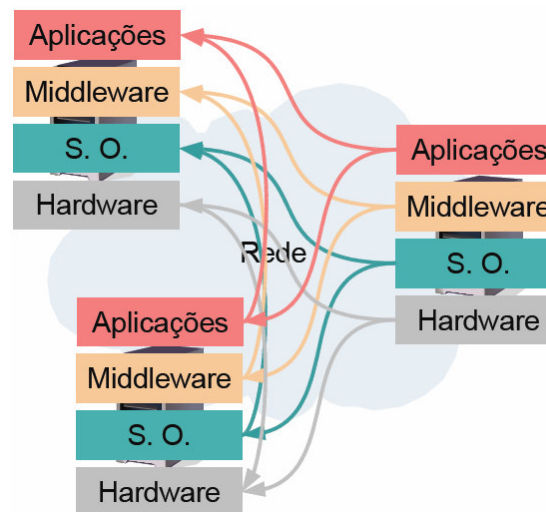
➔ Visão única e não coleção de componentes:

ex: portal de uma grade computacional.



➔ Abstração das camadas subjacentes:

ex: middleware.



Sucesso de um SD



grau de transparência

Em que medida é indistinguível de um sistema centralizado com a mesma funcionalidade?

Mantendo compromissos com
custos, desempenho, segurança,...

Transparência	Descrição
Acesso	Esconde diferenças na representação de dados e como um recurso é acessado
Localização	Esconde onde um recurso está localizado
Migração	Esconde que um recurso pode mover-se para outra localização
Relocação	Esconde que um recurso pode ser movido para outra localização enquanto esta sendo usado
Replicação	Esconde que um recurso pode ser compartilhado por vários usuários concorrentes
Concorrência	Esconde que um recurso pode ser compartilhado por vários usuários concorrentes
Falha	Esconde a falha e recuperação de um recurso
Persistência	Esconde quando um recurso (software) esta em memória ou em disco

Segundo Tanenbaum:

Esconde dos usuários ou desenvolvedores a existência de réplicas de recursos.

Transparência	Descrição
Acesso	Esconde diferenças na representação de dados e como um recurso é acessado
Localização	Esconde onde um recurso está localizado
Migração	Esconde que um recurso pode mover-se para outra localização
Relocação	Esconde que um recurso pode ser movido para outra localização enquanto esta sendo usado
Replicação	Esconde que um recurso pode ser compartilhado por vários usuários concorrentes
Concorrência	Esconde que um recurso pode ser compartilhado por vários usuários concorrentes
Falha	Esconde a falha e recuperação de um recurso
Persistência	Esconde quando um recurso (software) esta em memória ou em disco

Segundo Tanenbaum:

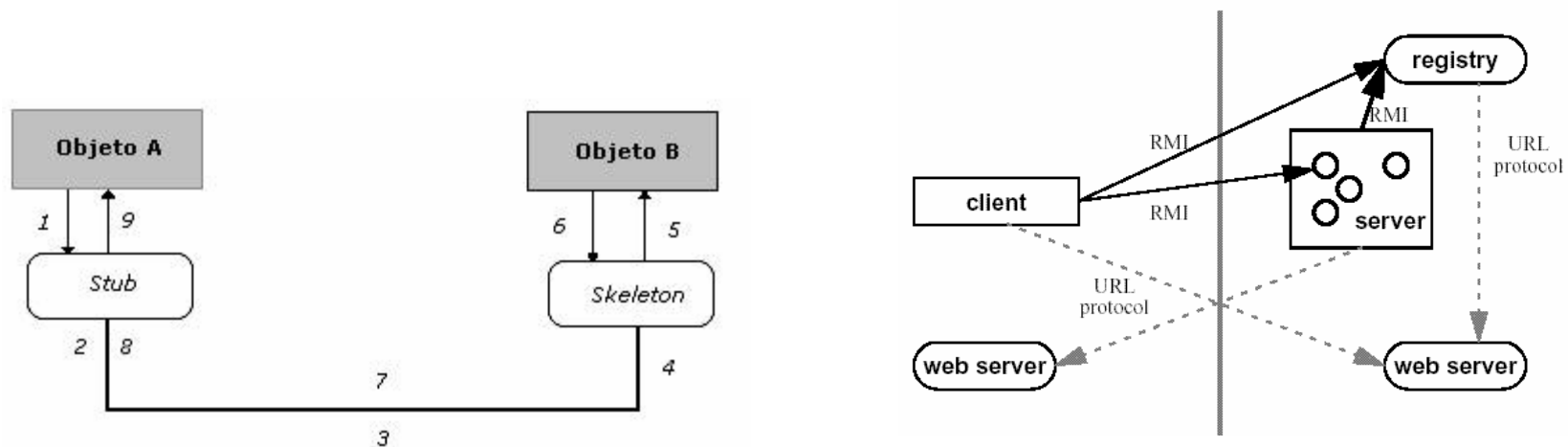
Esconde dos usuários ou desenvolvedores a existência de réplicas de recursos.

Complementando...

- ➔ **Desempenho**: reconfiguração para melhorar desempenho(balancamento de carga);
- ➔ **Escala**: expandir sem modificar: arquitetura do sistema ou algoritmos de aplicação.

- ➔ Recursos locais e remotos são acessados com as mesmas operações:

Exemplo: Java-RMI, serviço WWW.



1. O objeto A faz a chamada do método desejado no Stub
2. O Stub converte a chamada em protocolo RMI e envia pela Rede
3. A requisição é enviada pela rede
4. O Skeleton recebe a requisição em RMI e reconhece qual método deve ser chamado.
5. Chama o método desejado no Objeto B.
6. Recebe o retorno do método
7. Envia a o objeto de Retorno para o Cliente
8. O Stub recebe o retorno
9. O Stub repassa o retorno para o Objeto A.

Transparência de Localização

- ➡ Recurso é acessado independentemente da sua localização;
- ➡ Implica em mecanismos de resolução de nomes;

Exemplo: redes P2P, Mensagens Instantâneas.

Transparência de Localização

- ➡ Recurso é acessado independentemente da sua localização;
- ➡ Implica em mecanismos de resolução de nomes;

Exemplo: redes P2P, Mensagens Instantâneas.

Transparência de Concorrência

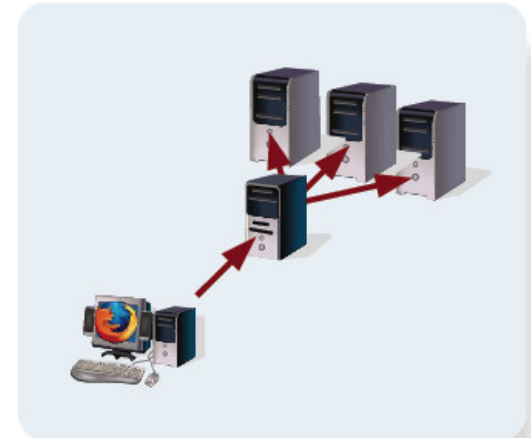
- ➡ O recurso é acessado independentemente da quantidade de usuários:

Exemplo: impressora em rede.

Transparência de Replicação

⇒ Várias cópias sem que o usuário perceba;

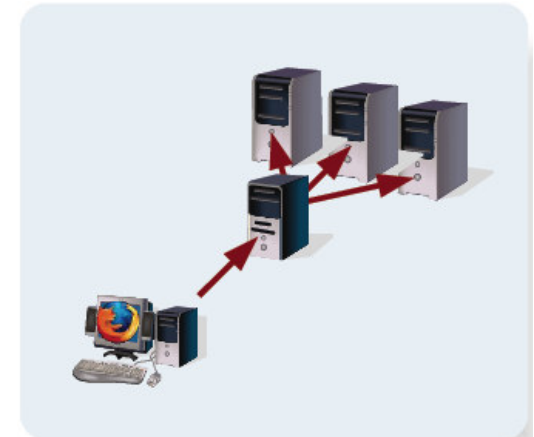
Exemplo: servidor WWW de provedores de acesso
(Terra, UOL,...)



Transparência de Replicação

➡ Várias cópias sem que o usuário perceba;

Exemplo: servidor WWW de provedores de acesso (Terra, UOL,...)



Transparência de Falha

➡ Na falha de um recurso, o usuário continua sendo atendido sem notar a ocorrência da falha;

➡ Implica na inexistência de um único ponto de falha (????):

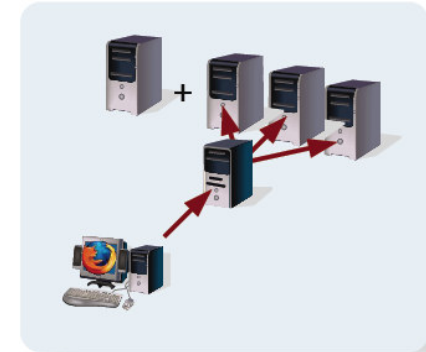
Exemplo: sistema bancário.



Transparência de Desempenho

- ➡ O sistema tira partido de recursos adicionais, sem que o usuário perceba.

Exemplo: computação colaborativa (SETI@home)



Transparência de Desempenho

- ➡ O sistema tira partido de recursos adicionais, sem que o usuário perceba.

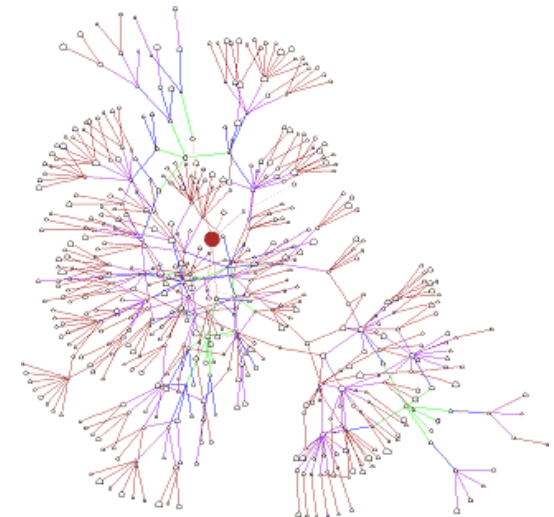
Exemplo: computação colaborativa (SETI@home)



Transparência de Escala

- ➡ Sistema funciona em escalas diferentes, sem alteração da aplicação.
- ➡ Implica na ausência de pontos de estrangulamento ou gargalos:

Exemplo: compartilhamentos de arquivos P2P.





Exercícios

Faculdades SENAC-RS

Sistemas Distribuídos

Lincoln L. de Moraes



Bibliografia

Faculdades SENAC-RS

Sistemas Distribuídos

Lincoln L. de Moraes