

INE 5410
PROGRAMAÇÃO
CONCORRENTE

Prof. Vitório Bruno
Mazzola
mazzola@inf.ufsc.br



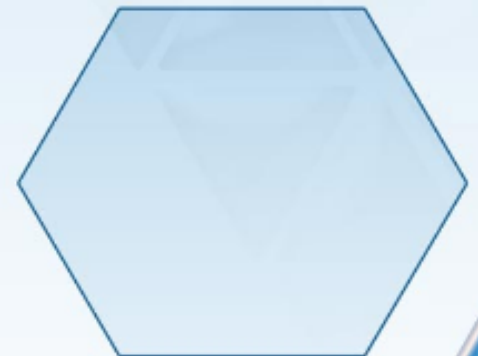
Capítulo 1

INTRODUÇÃO À PROGRAMAÇÃO CONCORRENTE



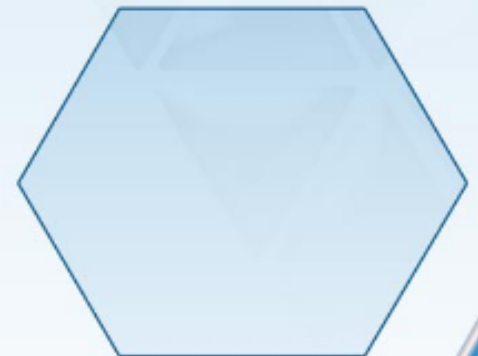
Motivação

- **Impacto do software sobre a sociedade**
 - Software é aplicado a um número cada vez maior de áreas da atividade humana
 - Exigência de qualidade e desempenho aumentam significativamente
 - Em muitas áreas, desempenho é requisito fundamental



Motivação

- **Impacto do software sobre a sociedade**
 - **Em muitas áreas de aplicação, as características intrínsecas ao problema sendo tratado requer a utilização de mecanismos de programação que contemple aspectos como:**
 - **Paralelismo**
 - **Não determinismo**
 - **Comunicação**
 - **Sincronização**



Motivação

- **Impacto do software sobre a sociedade**

- **Algumas áreas de aplicação:**


- **Controle de processos**
- **Controle de tráfego aéreo**
- **Monitoração ambiental**
- **Controle de veículos**
- **Robôs industriais**
- **Protocolos de comunicação**
- **Execução de algoritmos distribuídos**

Motivação

- **Sistemas embarcados**

- São sistemas onde é fundamental o gerenciamento do paralelismo natural dos diferentes componentes que o compõem
- O software associado é composto de diversos programas sequenciais, aliados ao suporte de um sistema operacional que seja capaz de gerenciar processos e a troca de dados entre estes

Plataformas

- **O suporte ao paralelismo é necessário para aumentar o desempenho de um sistema e ao mesmo tempo permitir a implementação de soluções que só seriam possíveis a partir do paralelismo**
 - **Níveis**
 - **Processador**
 - **Arquitetura do sistema computacional**
 - **Sistemas Operacionais**
 - **Infra-estrutura computacional**
 - **Linguagens de Programação**
- 

Plataformas

- **Processador**

- **Pipelining**

- Mecanismo que permite que o processador possa realizar a busca na memória de uma ou mais instruções, enquanto outras instruções estão sendo tratadas nos seus componentes;

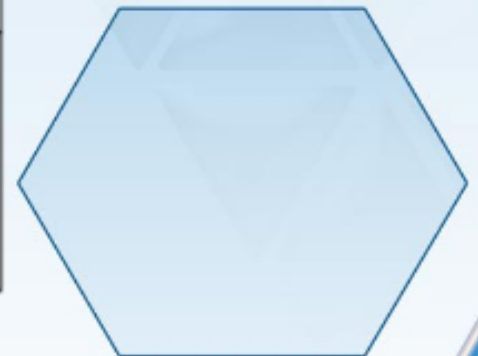
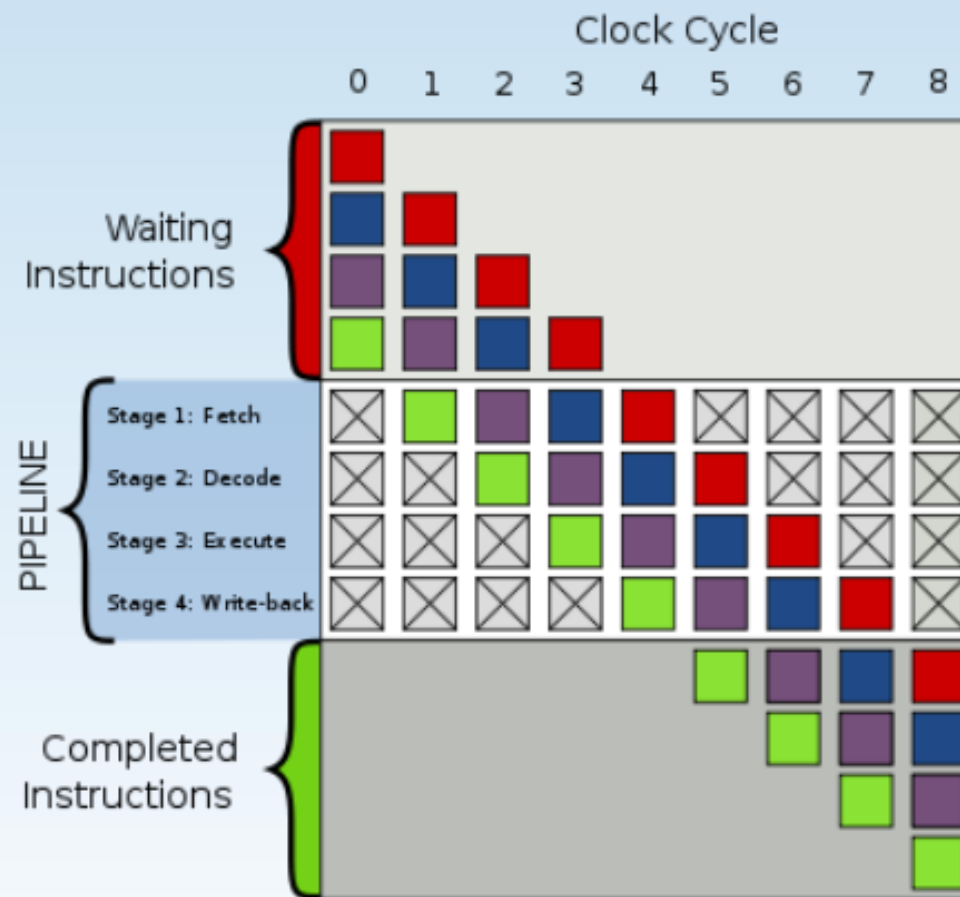
- **Memória cache**

- Pequena quantidade de memória de acesso rápido para onde são copiados blocos de instruções de um programa, com o objetivo de acelerar o acesso às instruções, melhorando o desempenho da execução de um programa;

Plataformas

- **Processador**

- **Pipelining**

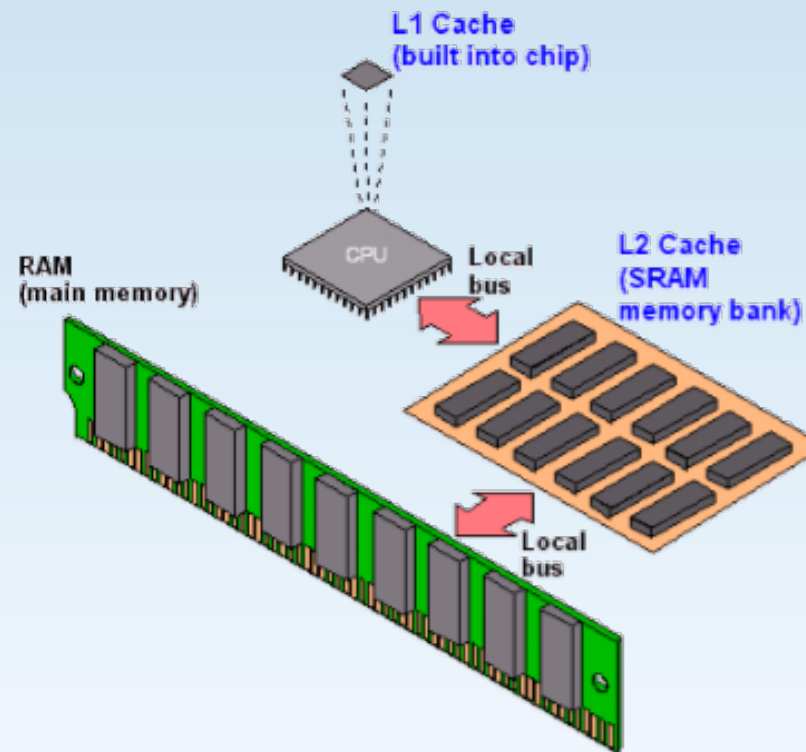


Plataformas

- **Processador**

- **Memória cache**

From Computer Desktop Encyclopedia
© 1999 The Computer Language Co., Inc.



Plataformas

- **Processador**

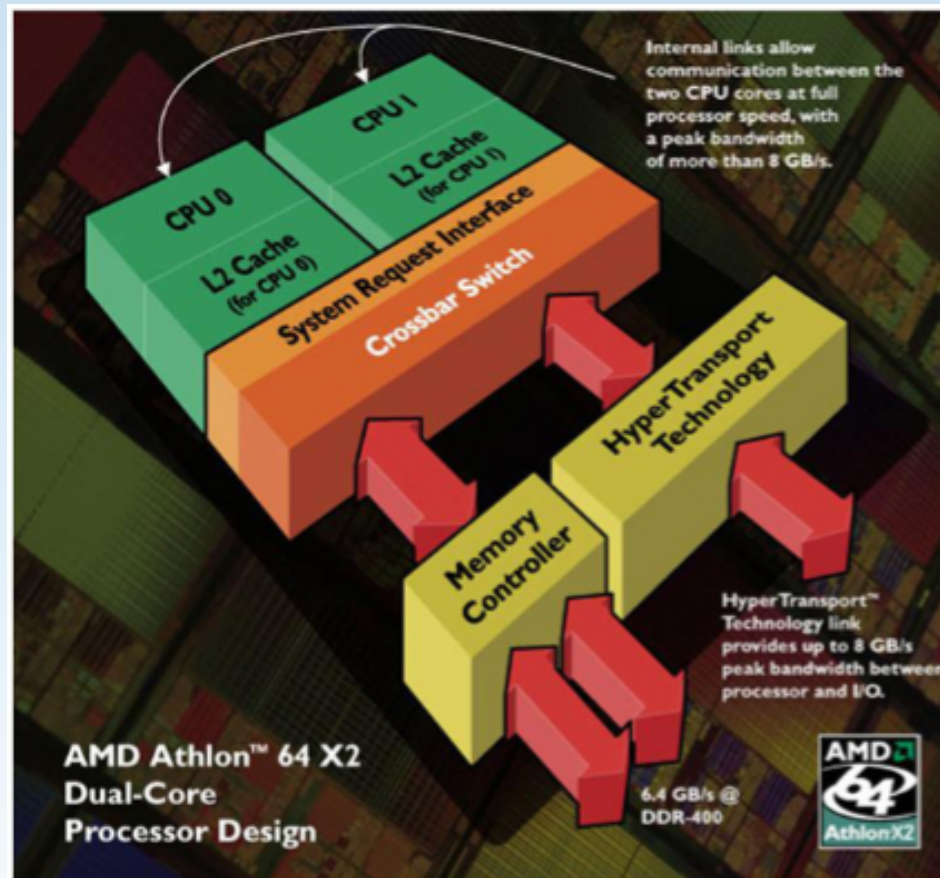
- **Multiplicidade de núcleos**

- Há alguns anos, o surgimento de processadores de vários núcleos (Dual Core, Core 2 Duo, Core 2 Quad, etc) teve como consequência o benefício no que diz respeito ao desempenho de um sistema computacional
 - Aplicações que são mapeadas em diversos fluxos de instrução (threads) e são as mais beneficiadas
 - A execução de diversas aplicações exigentes em processamento é outro aspecto que encontra benefícios através desta classe de processadores

Plataformas

- **Processador**

- **Multiplicidade de núcleos**

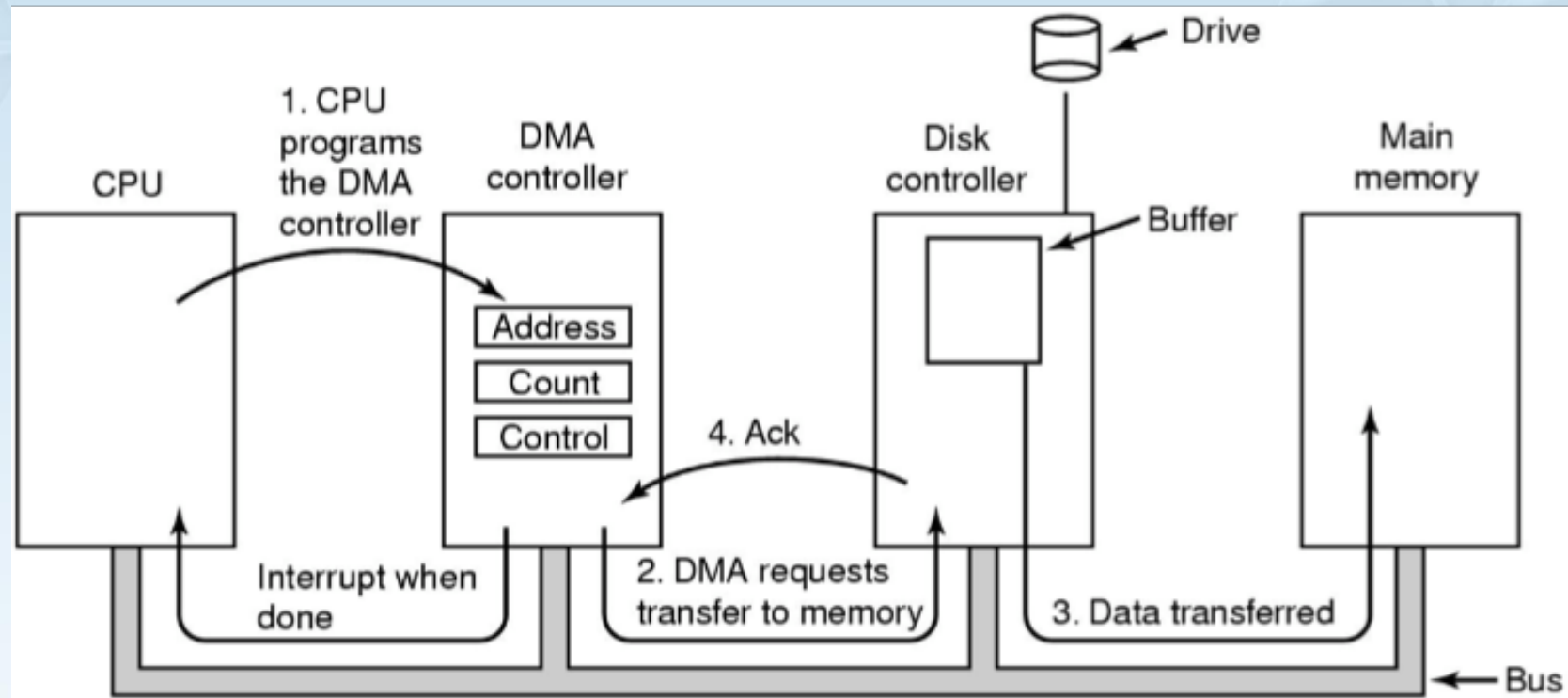


Plataformas

- **Arquitetura do sistema computacional**
- **Controle de Acesso à Memória (DMA)**
 - Controlador utilizado para o gerenciamento de transferência de dados entre memória e outros dispositivos
 - Útil para a transferência de grandes quantidades de dados entre blocos de memória e periféricos ou componentes internos (ex.: disco rígido)
 - Libera a CPU da realização desta tarefa, permitindo que esta continue a realizar outras instruções

Plataformas

- Arquitetura do sistema computacional
- Controle de Acesso à Memória (DMA)

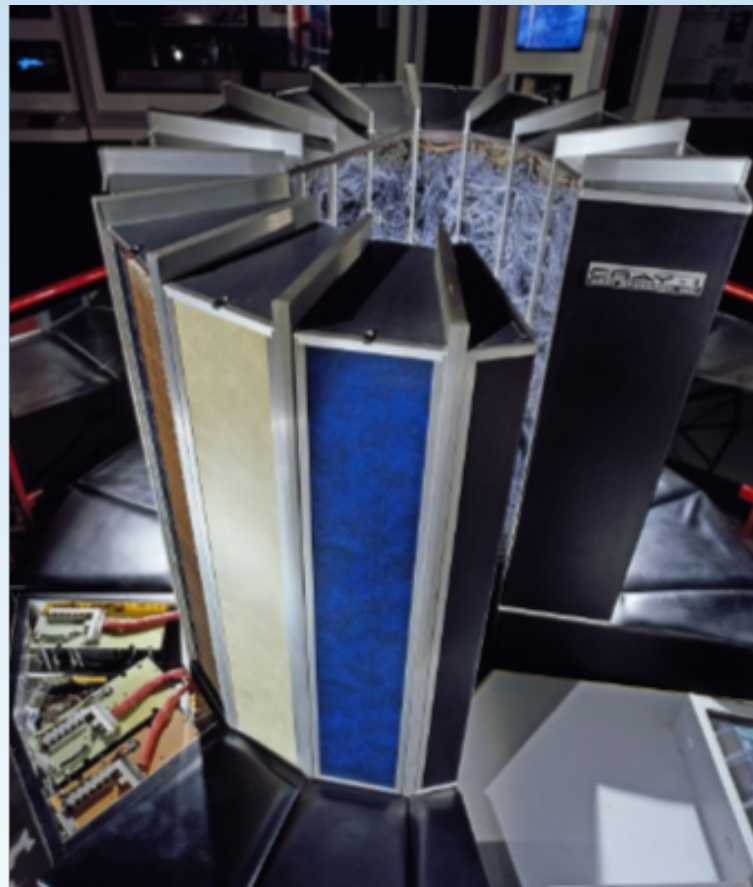


Plataformas

- **Arquitetura do sistema computacional**
- **Multiplicidade de CPUs**
 - Computadores vetoriais, caracterizados por diversos processadores residindo numa mesma arquitetura e comunicando-se através de barramentos específicos de alta velocidade
 - Clusters, compostos de diversos computadores que assumirão partes da execução de tarefas complexas
 - Grids, que fazem uso de uma infra-estrutura de rede de computadores para a realização de tarefas num ambiente distribuído

Plataformas

- **Arquitetura do sistema computacional**
- **Multiplicidade de CPUs**



Plataformas

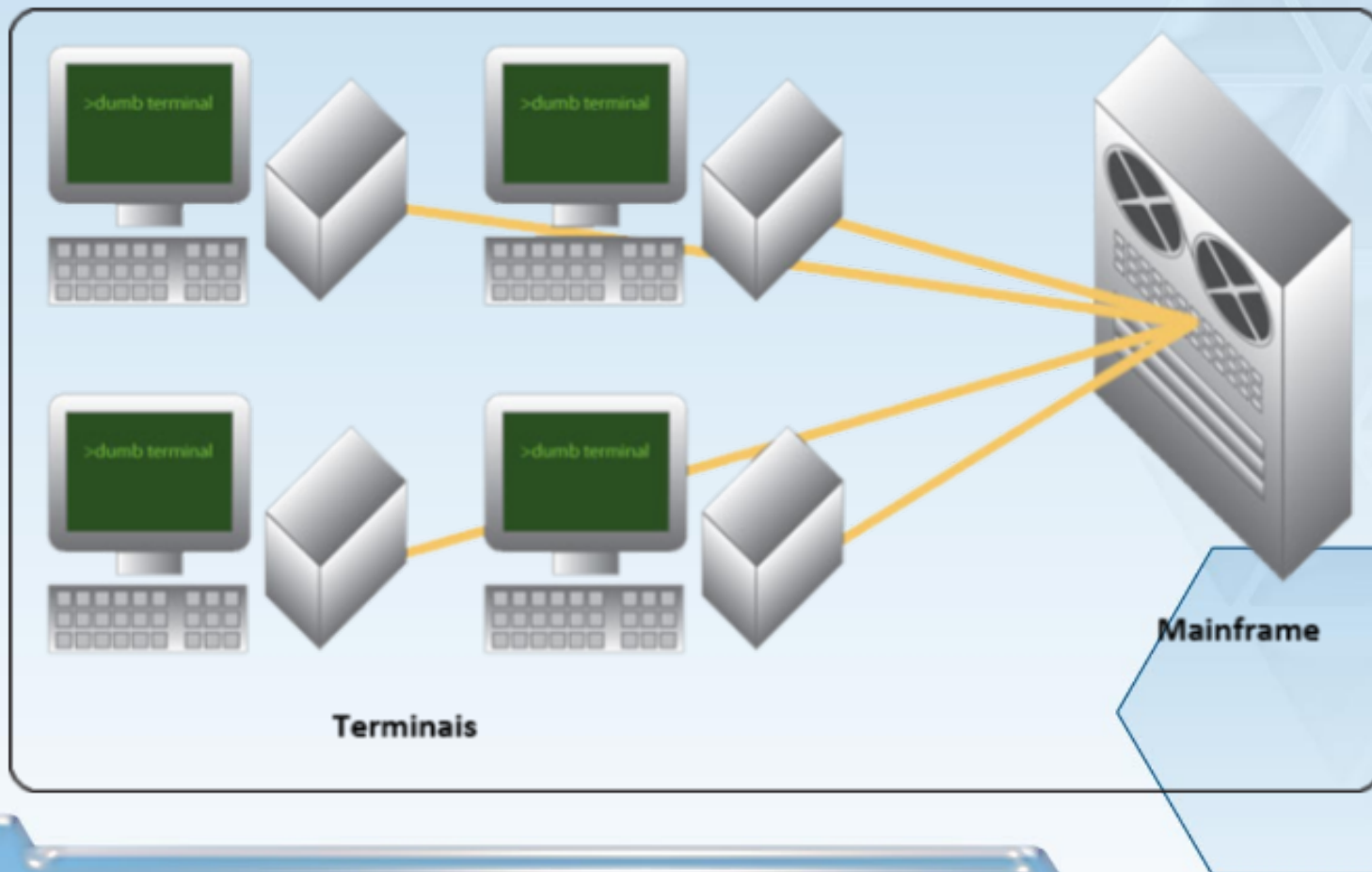
- **Sistemas Operacionais**

- **Time-sharing**

- A técnica de time-sharing foi a primeira tentativa de trazer o conceito de “paralelismo” aos sistemas computacionais
- Através desta técnica, a posse da cpu e seus recursos para os diversos programas permitia um virtual paralelismo entre os programas de usuários a serem executador num único computador
- Infra-estrutura dependia de um único computador (mainframe) conectado a diversos terminais

Plataformas

- **Sistemas Operacionais**
- **Time-sharing**

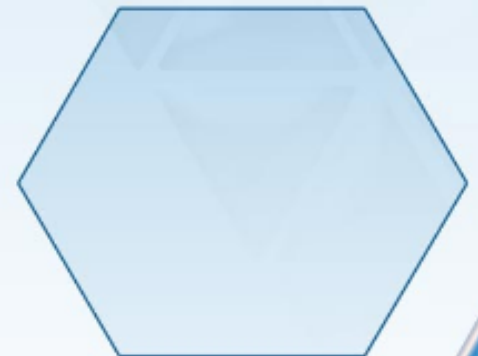


Plataformas

- **Sistemas Operacionais**

- **Multitarefaamento**

- O surgimento dos sistemas baseados em multitarefaamento revoluciona a forma de execução de programas num sistema computacional
- Surgimento de esquema de prioridade entre tarefas
- Escalonador gerencia a alocação dos recursos para necessários à execução de tarefas, em especial o processador



Plataformas

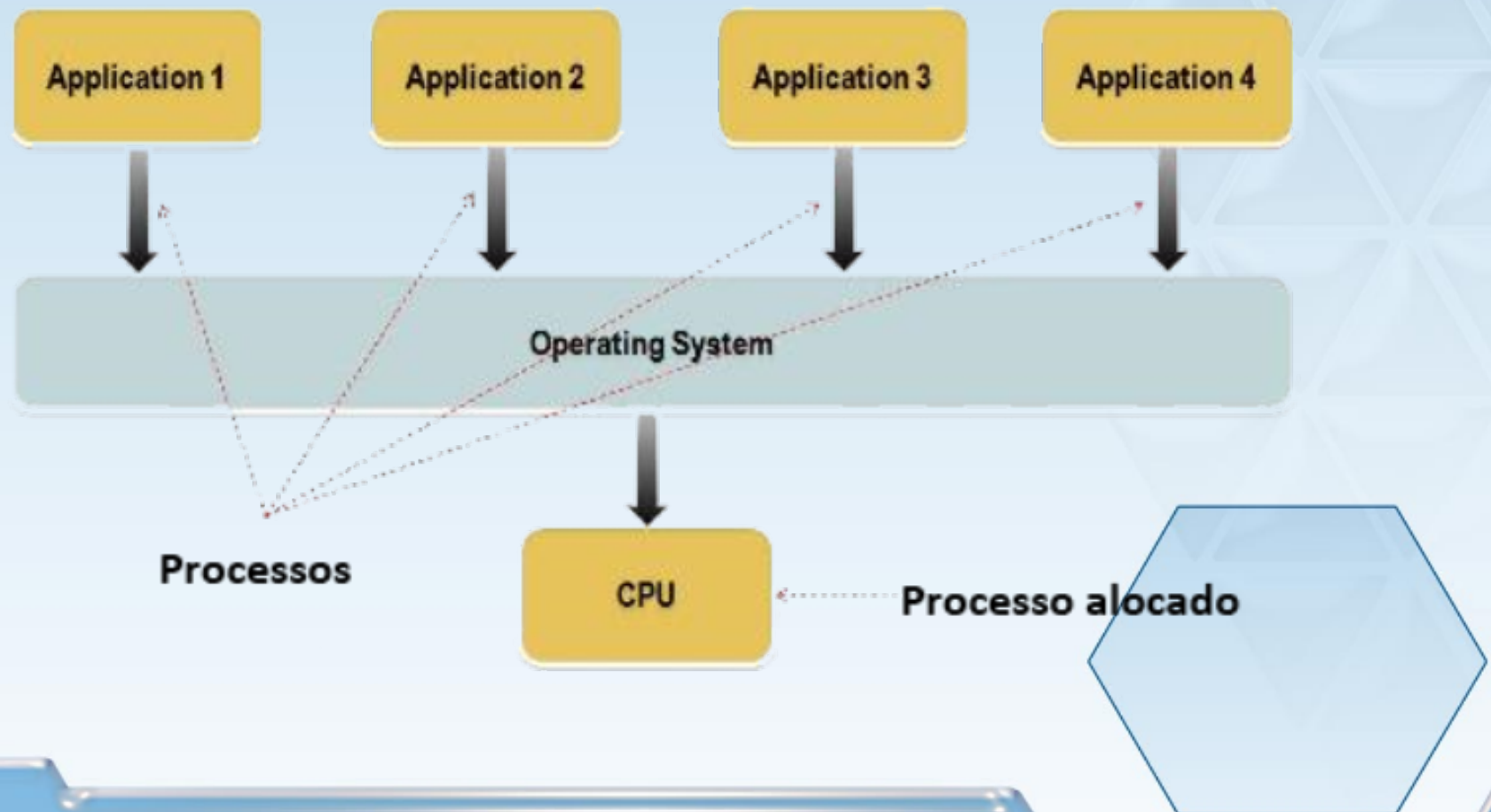
- **Sistemas Operacionais**

- **Multitarefaamento**

- Cada programa é mapeado numa tarefa ou processo
- Conceito de preemptividade passa a ser adotado em todos os sistemas como forma de melhor atender às necessidades de execução das tarefas
- Evolução importante que é incorporada aos sistemas operacionais mais usados

Plataformas

- **Sistemas Operacionais**
- **Multitasking**



Plataformas

- **Infra-estrutura computacional**
- **Redes de Computadores**
 - Locais ou em longa distância, cabeadas ou wireless, as redes de computadores representam um avanço inestimável no desenvolvimento de aplicações paralelas;
 - A repartição das capacidades de processamento entre diversos computadores, de certo modo, autosuficientes, permite uma eficiente adaptação dos recursos computacionais às aplicações distribuídas, cada vez mais exigentes;
 - Protocolos de comunicação já são exemplo disso;

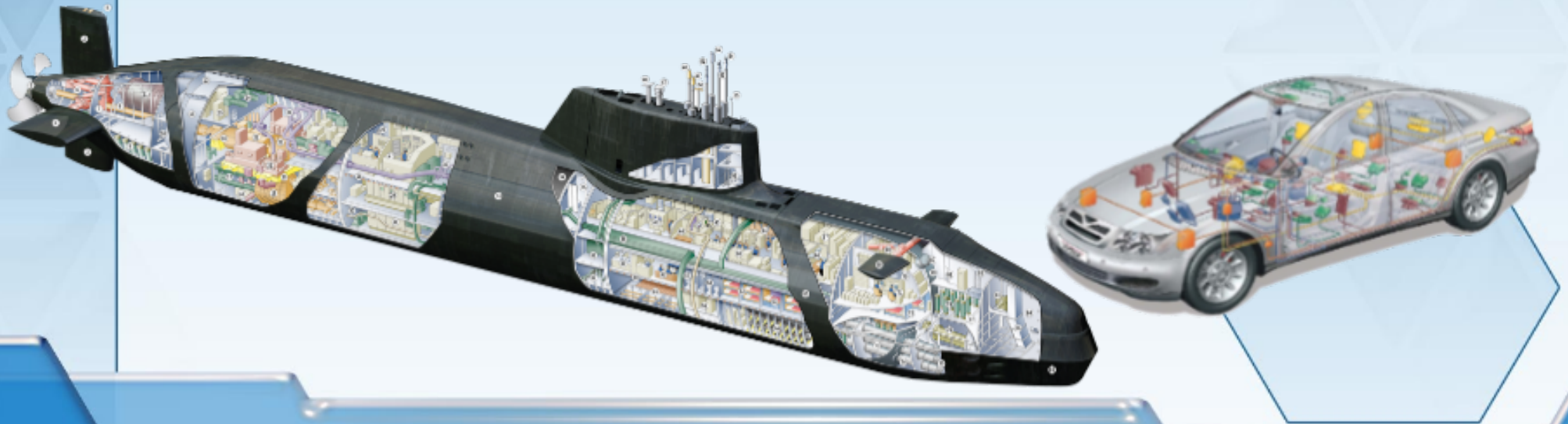
Plataformas

- **Infra-estrutura computacional**
- **Redes de Computadores**
 - Na área do controle de processos em automação industrial, o uso de arquiteturas adaptadas aos requisitos de tempo real inerentes das tarefas de controle já é uma realidade há décadas;



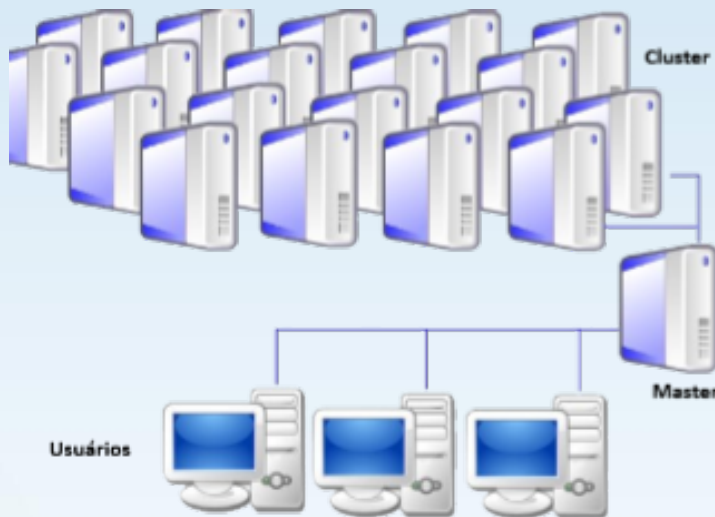
Plataformas

- **Infra-estrutura computacional**
- **Redes de Computadores**
 - Os modernos veículos, de finalidade militar ou de transporte, utilizam em sua eletrônica embarcada, redes de sensores para viabilizar a comunicação entre sensores e atuadores dos diversos componentes de controle do veículo;



Plataformas

- **Infra-estrutura computacional**
- **Clusters**
 - Clusters, compostos de diversos computadores que assumirão partes da execução de tarefas complexas



Plataformas

- **Infra-estrutura computacional**
- **Clusters**
 - As imagens em computação gráfica do naufrágio do navio Titanic no filme de James Cameron em 1996 só tiveram o grau de realismo graças à utilização de um clustes composto de 160 nós executando sistema operacional Linux



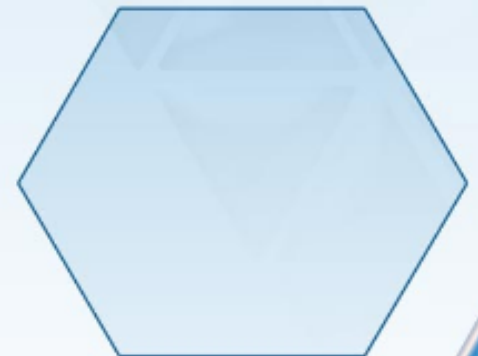
Plataformas

- **Infra-estrutura computacional**
- **Grids**
 - Grids, que fazem uso de uma infra-estrutura de rede de computadores para a realização de tarefas num ambiente distribuído



Plataformas

- **Linguagens de programação**
- **Suporte à concorrência**
 - Linguagens orientadas à programação concorrente passaram a fazer parte do universo dos programadores;
 - Oferecem mecanismos adequados para diversos aspectos da concorrência, entre eles:
 - Definição de unidades executáveis (módulos, tarefas, processos, etc.)
 - Comunicação entre tarefas
 - Mecanismos de sincronização



Plataformas

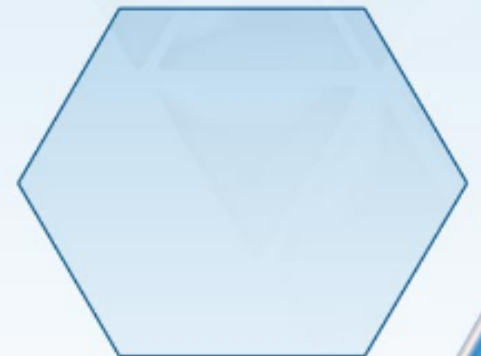
- **Linguagens de programação**
- **Suporte à concorrência**
 - Um exemplo clássico de linguagem de programação com este perfil é a linguagem Ada;
 - Ada foi criada em meados dos anos 1970 no Departamento de Defesa Americano para ser utilizada no desenvolvimento de programas para aplicações de controle de aviões
 - Um dos aspectos presentes em Ada é um mecanismo de comunicação síncrona denominado “rendez-vous” baseado na troca de mensagens entre tarefas definidas no programa

Plataformas

- Linguagens de programação
- Suporte à concorrência
 - Outro mecanismo existente em linguagens de programação mais recentes é o conceito de *thread*;
 - Uma thread corresponde a um fluxo de execução que é criado no contexto de um programa;
 - Um programa pode conter diversas threads que executam concorrentemente num sistema computacional, independente da quantidade de recursos disponíveis para sua execução;

Plataformas

- **Linguagens de programação**
- **Suporte à concorrência**
 - **Linguagens orientadas à programação concorrente incorporam ainda alguns mecanismos clássicos de controle da concorrência**
 - **Semáforos e monitores são exemplos de tais mecanismos**



Um exemplo

- Redes em barramento



Conceitos

- **Sincronização**

- **Mecanismo relacionado à execução de eventos de modo simultâneo**
- **O conceito de sincronização permite que se possa controlar a ocorrência de diversos eventos, de modo a estabelecer um relacionamento entre eles:**
 - **Um evento A deve ocorrer antes de um evento B**
 - **Dois eventos não podem ocorrer ao mesmo tempo**

Conceitos

- **Exclusão Mútua**

- **Técnica utilizada para controle do acesso a recursos por programas concorrentes**
- **Objetivo é evitar que programas tenham possam dispor simultaneamente de um único recurso do sistema**
- **Exemplo: dois ou mais programas que necessitem realizar operações sobre registros em tabelas de uma base de dados.**

Conceitos

- **Seção Crítica**

- **Corresponde a um trecho de um dado programa que é caracterizado pelo acesso a um ou mais recursos compartilhados num ambiente computacional**
- **O controle ao ingresso em seção crítica de um sistema é uma das importantes questões para evitar problemas de execução ou má utilização de recursos compartilhados pelos diversos programas concorrentes**

Conceitos

- **Seção Crítica**

- Uma vez que um programa entra em seção crítica, a execução das instruções relacionadas a este trecho deve ocorrer de modo atômico, sendo que o acesso ao(s) recurso(s) deve ser assegurado pelo tempo em que o referido programa necessite utilizá-lo
- Exemplo: programa vai realizar instruções que envolvam o uso de um processador gráfico para realizar uma dada função.

Conceitos

- **Deadlock**

- **Caracteriza uma situação em que dois ou mais processos tornam-se impedidos de continuar suas execuções**
- **Isto ocorre, em geral, quando processos aguardam a liberação de recursos alocados por outros processos, os quais encontram-se impedidos de continuar suas execuções, por estarem aguardando a liberação de outros recursos alocados pelos processos de um primeiro grupo.**

Conceitos

- **Deadlock (exemplo na vida prática)**



Conceitos

- **Paralelismo**

- **Dois ou mais eventos ocorrem paralelamente quando possuem independência entre eles sob os seguintes pontos de vista:**
 - **Restrições de serialização... (não existe uma ordenação entre eles na operação global no sistema a que pertencem)**
 - **Com relação ao agente (a entidade responsável pela ocorrência dos eventos)**
 - **Com relação aos recursos necessários para a ocorrência dos eventos**

Conceitos

- **Paralelismo**

- Quando não existe esta independência, os eventos estão ocorrendo de modo concorrente
- Isto significa que estarão, eventualmente, competindo por recursos do ambiente em que estão ocorrendo e, para que não haja problemas nestas ocorrências, há de se implementar mecanismos de controle da concorrência

Conceitos

- **Não determinismo**

- **A existência de processos, tarefas, threads, ou qualquer nome que se dê a diferentes eventos ocorrendo de modo simultâneo, evita que se possa prever, em muitos casos a ordem em que estes eventos venham a ocorrer**
- **A estas incertezas, pode-se associar grande parte em se desenvolver programas concorrentes...**

Conceitos

- **Não determinismo**

- **É justamente o reconhecimento do não determinismo inerente a um dado programa concorrente que nos permite manipular esta incerteza**
- **Técnicas de modelagem e linguagens de programação concorrente permitem o uso deste conceito em favor da realização dos sistemas concorrente de modo a evitar os danos advindos do não determinismo**

Conceitos

- **Programação Concorrente**

- **Paradigma de programação para o desenvolvimento de software baseado na execução simultânea de tarefas interativas**
- **Tarefas são implementadas por diversos programas ou threads definidas por um mesmo programa**
- **Executados por um único processador, diversos processadores numa única máquina ou processadores distribuídos numa rede**