

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

INTRODUÇÃO AO PARALELISMO: PROCESSADORES SUPERESCALARES

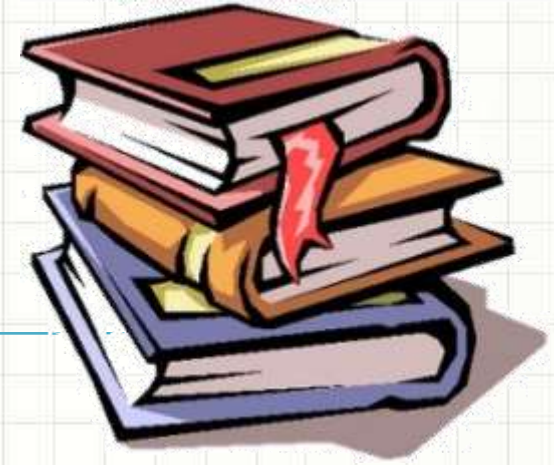
Prof. Dr. Daniel Caetano

2011 - 2

Visão Geral

- 1 • Introdução
- 2 • Arquitetura Superpipeline
- 3 • Arquitetura Superescalar
- 4 • Comparativo
- 5 • Multiprocessamento
- 6 • Computador Completo

Material de Estudo



Material

Acesso ao Material

Notas de Aula

<http://www.caetano.eng.br/aulas/aoc/>
(Aula 12)

Apresentação

<http://www.caetano.eng.br/aulas/aoc/>
(Aula 12)

Material Didático

-

Biblioteca Virtual

Arquitetura e Organização de Computadores, páginas
647 a 652

Lembretes

- **Lista 2 Online!**





INTRODUÇÃO

Introdução

- Aula passada: vantagem do Pipeline

Sequência no Tempo	SEM pipeline		COM pipeline	
	Busca	Execução	Busca	Execução
0	I1	-	I1	-
1	-	I1	I2	I1
2	I2	-	I3	I2
3	-	I2	I4	I3
4	I3	-	I5	I4

- Ocupar partes ociosas do processador



ARQUITETURA SUPERPIPELINE

Pipeline de Múltiplos Estágios

- Como visto na aula passada...

Tempo	BI	DI	CO	BO	EI	EO
0,00	I1	-	-	-	-	-
0,33	I2	I1	-	-	-	-
0,66	I3	I2	I1	-	-	-
1,00	I4	I3	I2	I1	-	-
1,33	I5	I4	I3	I2	I1	-
1,66	I6	I5	I4	I3	I2	I1
2,00	I7	I6	I5	I4	I3	I2

Pipeline de Múltiplos Estágios

- Como visto na aula passada...

Tempo	BI	DI	CO	BO	EI	EO
0,00						
0,33						
0,66						
1,00						
1,33						
1,66						
2,00	17	16	15	14	13	12

**Será que todos estes
estágios demoram
exatamente o mesmo
tempo para serem
concluídos?**

Superpipeline

- No pipeline tradicional
 - O tempo dos estágios mais rápidos é limitado pelo tempo dos estágios mais lentos
- Tempo de cada estágio: controlado pelo clock
 - Exemplo: estágio lento ocupa 2 ciclos de clock... uma instrução a cada 2 ciclos
- Acelerar o clock geral... resolve?
 - Acelera todos os estágios, mas os estágios lentos continuam lentos
 - Exemplo: instruções continuam usando dois ciclos de clock!

Superpipeline

- Acelerar o **clock interno**
 - Permite execução mais rápida dos estágios mais lentos
 - Exemplo: clock interno = 2x clock externo...
 - Agora estágio lento é cumprido em 2 ciclos **internos**
 - Mas cada ciclo externo equivale a dois internos...
 - Logo, no exemplo, permite uma instrução por ciclo
- Superpipeline: pipeline com clock interno da CPU maior que o clock externo



ARQUITETURA SUPERESCALAR

Paralelismo em Nível de Instruções

- Arquitetura pipeline: domínio inicial da Intel
- AMD vinha “colada”... Intel tinha que inovar
 - Que tal executar várias instruções ao mesmo tempo?
 - Múltiplos pipelines: instruções distribuídas
- Quaisquer instruções podem ser executadas ao mesmo tempo?
- **Instruções Independentes**
- O que são?

Instruções Independentes

- Exemplo...
 - LD A, 17 ; Carrega A com 17
 - ADD A,20 ; Soma 20 em A
 - Quanto vale A?
-
- ADD A, 20 ; Soma 20 em A
 - LD A,17 ; Carrega A com 17
 - Quanto vale A?
-
- Para produzir o primeiro efeito, a instrução LD A,17 precisa ter sido executada **antes** de ADD A,20
 - O resultados dependem da ordem das instruções

Instruções Independentes

- Vejamos um outro programa
- LD A, 17 ; Carrega A com 17
- ADD A, 20 ; Soma 20 em A
- LD C, A ; Guarda resultado (37) em C
- LD A, 30 ; Carrega A com 30
- ADD A, 10 ; Soma 10 em A
- LD D, A ; Guarda resultado (40) em D
- Essas linhas são dependentes?

Instruções Independentes

- Vejamos um outro programa
- LD A, 17 ; Carrega A com 17
- ADD A, 20 ; Soma 20 em A
- LD C, A ; Guarda resultado (37) em C
- LD A, 30 ; Carrega A com 30
- ADD A, 10 ; Soma 10 em A
- LD D, A ; Guarda resultado (40) em D
- Essas linhas são dependentes?
- Vamos trocar os registradores...

Instruções Independentes

- Vejamos um outro programa
- LD A, 17 ; Carrega A com 17
- ADD A, 20 ; Soma 20 em A
- LD C, A ; Guarda resultado (37) em C
- LD B, 30 ; Carrega B com 30
- ADD B, 10 ; Soma 10 em B
- LD D, B ; Guarda resultado (40) em D
- Essas linhas são dependentes?
- E vamos reorganizar as linhas...

Instruções Independentes

- Vejamos um outro programa
- LD A, 17 ; Carrega A com 17
- LD B, 30 ; Carrega B com 30
- ADD A, 20 ; Soma 20 em A
- ADD B, 10 ; Soma 10 em B
- LD C, A ; Guarda resultado (37) em C
- LD D, B ; Guarda resultado (40) em D
- Essas linhas são dependentes?
- Ainda são dependentes?

Instruções Independentes

- Vejamos um outro programa

- LD A, 17

- LD B, 30

- ADD A, 20

- ADD B, 10

- LD C, A

- LD D, B

- Essas linhas são dependentes?

- Ainda são dependentes?

**Otimizador de
código para
processadores
superescalares**

Instruções Independentes

- Vejam os outros programas

Superescalar:

Excelente para

processamento de imagens:

Pixels são Independentes!

- LD D, B ; Guarda resultado (40) em D
- Essas linhas são dependentes?
- Ainda são dependentes?

Instruções Independentes

Fim da Festa:

CP A,B

RET Z

ADD A,10

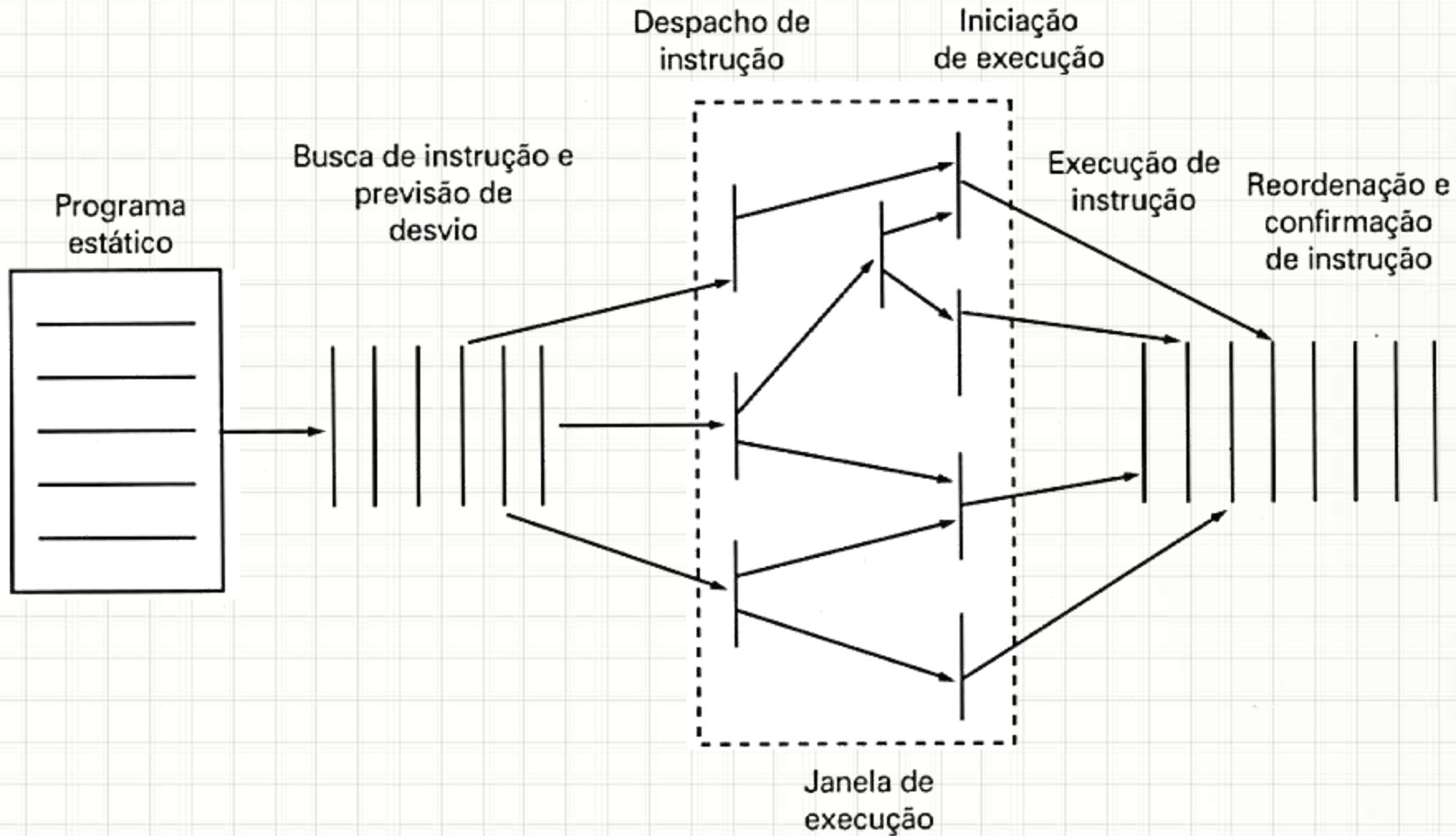
Dependência difícil de resolver!

n C

n D

• An

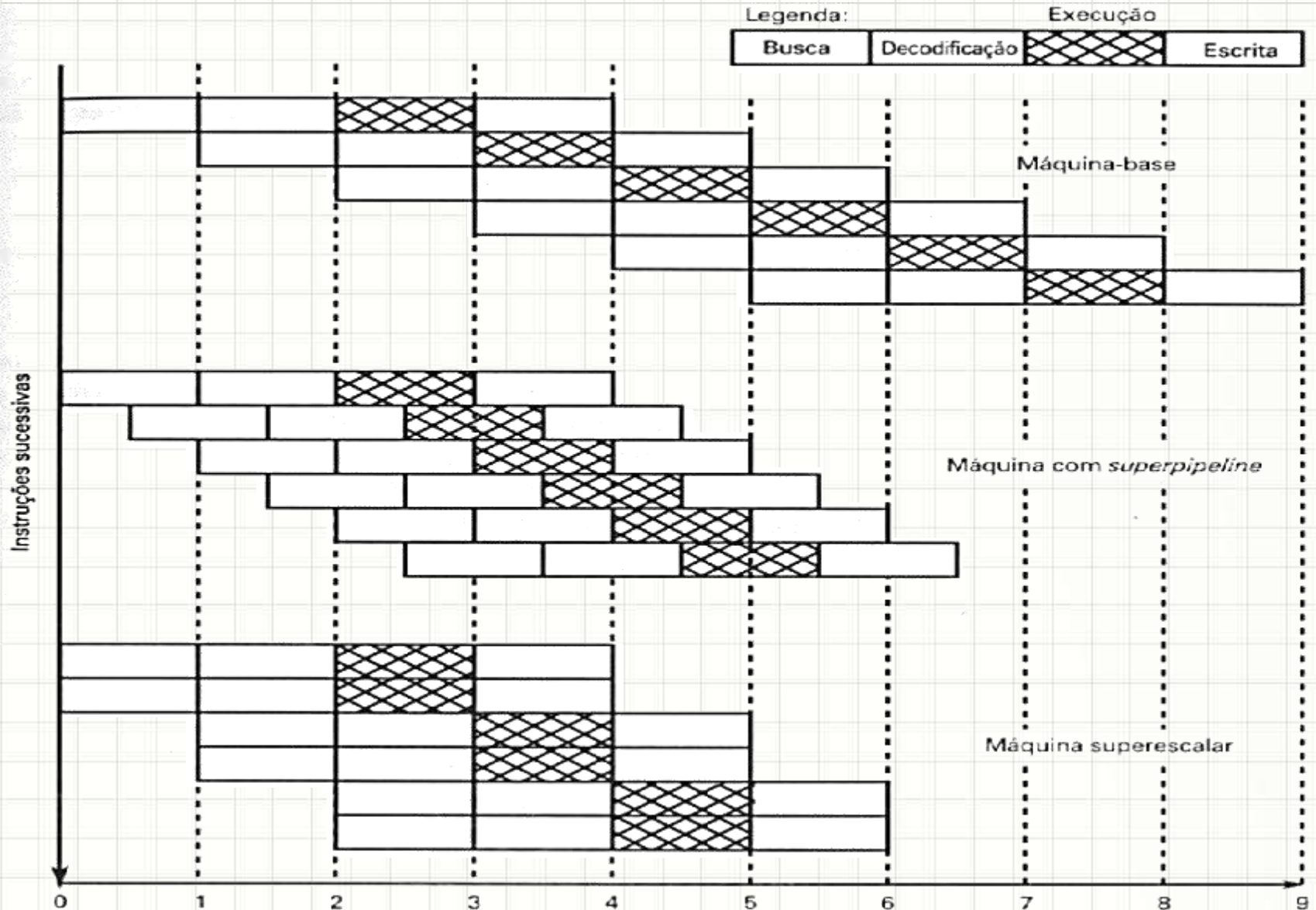
Lógica Superescalar





COMPARATIVO

(Super)Pipeline x Superescalar





FUNDAMENTOS DO MULTIPROCESSAMENTO

Tipos de Multiprocessamento

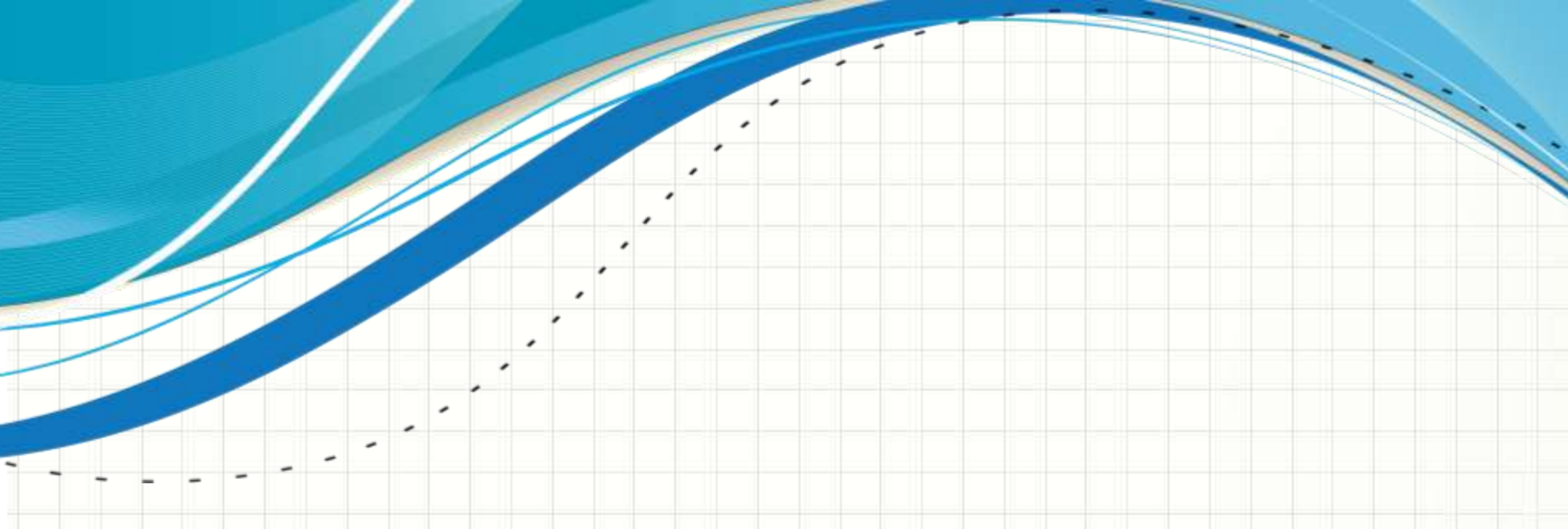
- Superescalar
 - Paralelismo em nível de instruções
- Múltiplos programas...
 - Paralelismo em nível de processos?
- Os tipos mais conhecidos são
 - **SMP**: Symetric MultiProcessing
 - **Clusters**
 - **NUMA**: Non-Uniform Memory Access
- Detalhes na próxima aula...!

Tipos de Multiprocessamento

- Classificação do Processamento:
 - **SISD**: Single Instruction, Single Data
 - Pipelines e Processadores Escalares
 - **SIMD**: Single Instruction, Multiple Data
 - Processamento Vetorial e Matricial
 - **MISD**: Multiple Instruction, Single Data
 - Aplicação apenas teórica
 - **MIMD**: Multiple Instruction, Multiple Data
 - SMP e Clusters
- Processamento vetorial?
 - Supercomputadores: imagens, simulação de partículas etc.

Tipos de Multiprocessamento





COMPUTADOR COMPLETO

Conceito

- O que é um computador completo?
 - Whole Computer
- Computador com processamento, unidades de entrada e saída e memória próprias
- **SMP** e **NUMA**: um único computador, com várias unidades de processamento
- **Clusters**: vários computadores completos interligados por rede
 - Cada computador completo é um **nó** de processamento

Consequência

- Um cluster com N nós...
 - **Pode** ser dividido em N computadores independentes
- Um sistema SMP/NUMA com N processadores
 - **NÃO pode** ser dividido em N computadores independentes



CONCLUSÕES

Resumo

- O multiprocessamento busca aumento do desempenho de processamento
- O multiprocessamento pode ser em nível de instruções e em nível de processos
- Para aproveitar o multiprocessamento em nível de instruções, muitos cuidados são exigidos com o programa
- No multiprocessamento em nível de processos, muitos cuidados são exigidos com o hardware
- **TAREFA**
 - Lista de Exercícios 2!

Próxima Aula



- Como funcionam o multiprocessamento em nível de processos?
 - Qual a diferença entre SMP/NUMA e Clusters?
 - Qual deles é melhor?



PERGUNTAS?



**BOM DESCANSO
A TODOS!**