FATEC

CENTRO PAULA SOUZA
COMPETÊNCIA EM EDUCAÇÃO PÚBLICA PROFISSIONAL
Faculdade de Tecnologia Prof. Jessen Vidal
São José dos Campos

GOVERNO DO ESTADO DE SÃO PAULO

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Disciplina: Sistemas Operacionais I

Aula 10: Thread P1

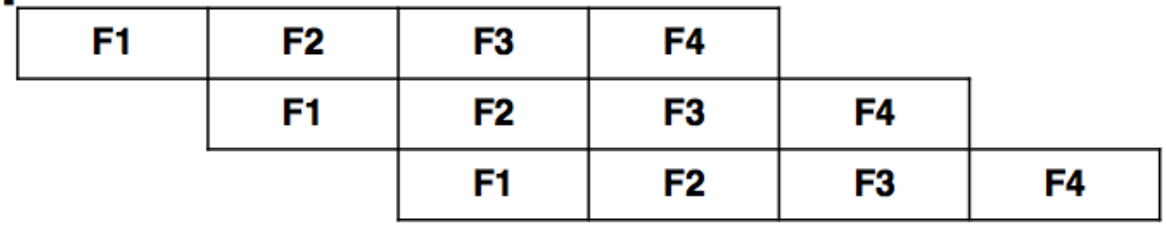
Prof. Diogo Branquinho Ramos
diogo.branquinho@fatec.sp.gov.br
São José dos Campos - SP

Roteiro

- Visão geral de arquiteturas computacionais
- Definição de thread
- Threads de usuário e de kernel

Evolução das arquiteturas

- **Pipeline: busca de mais de uma instrução além da próxima**

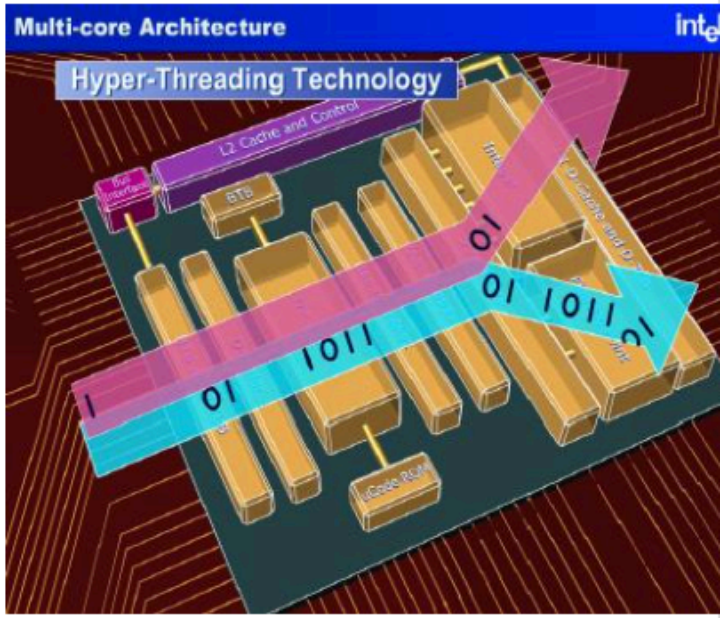


Instruções em fases diferentes

- **Hierarquia de memórias: uso de cache**
- **Arquiteturas com mais de um pipeline**
  - Pipelines por função.
- **Superescalares**
  - Possuem pipelines que permitem a execução de mais de uma instrução no mesmo clock.
    - Obtido pelas múltiplas unidades funcionais.

Evolução das arquiteturas

- **Execução Multithreaded**
  - Execução de mais de um fluxo de execução simultaneamente.
  - Simula dois processadores.
    - TLP: Paralelismo a Nível de Thread.
    - Usa partes da CPU não aproveitadas na previsão de desvio do pipeline.
    - Cada CPU lógica recebe seu controlador de interrupção programável (APIC) e um conjunto de registradores.

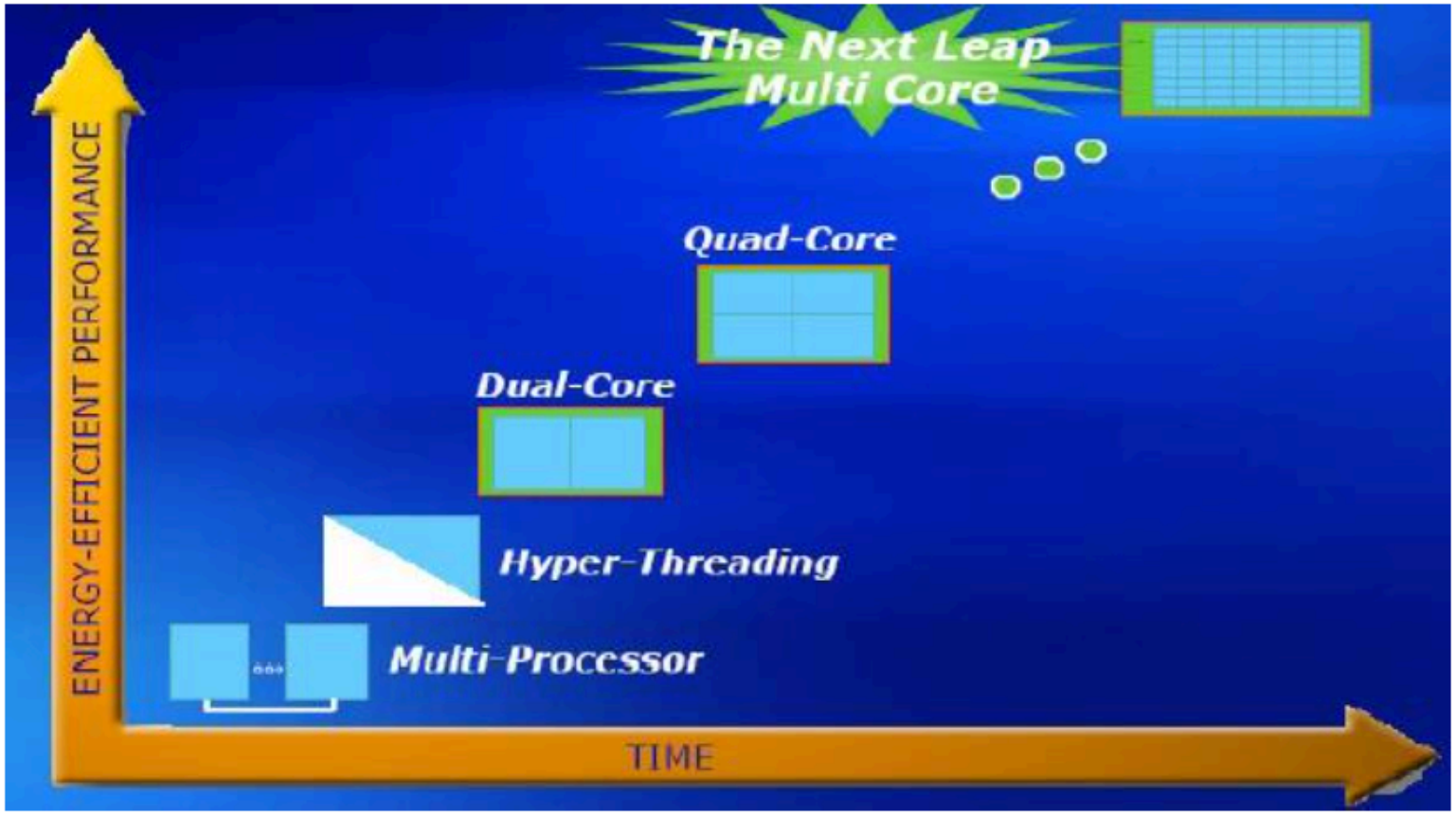


Primeiro HT:
Pentium 4
Northwood

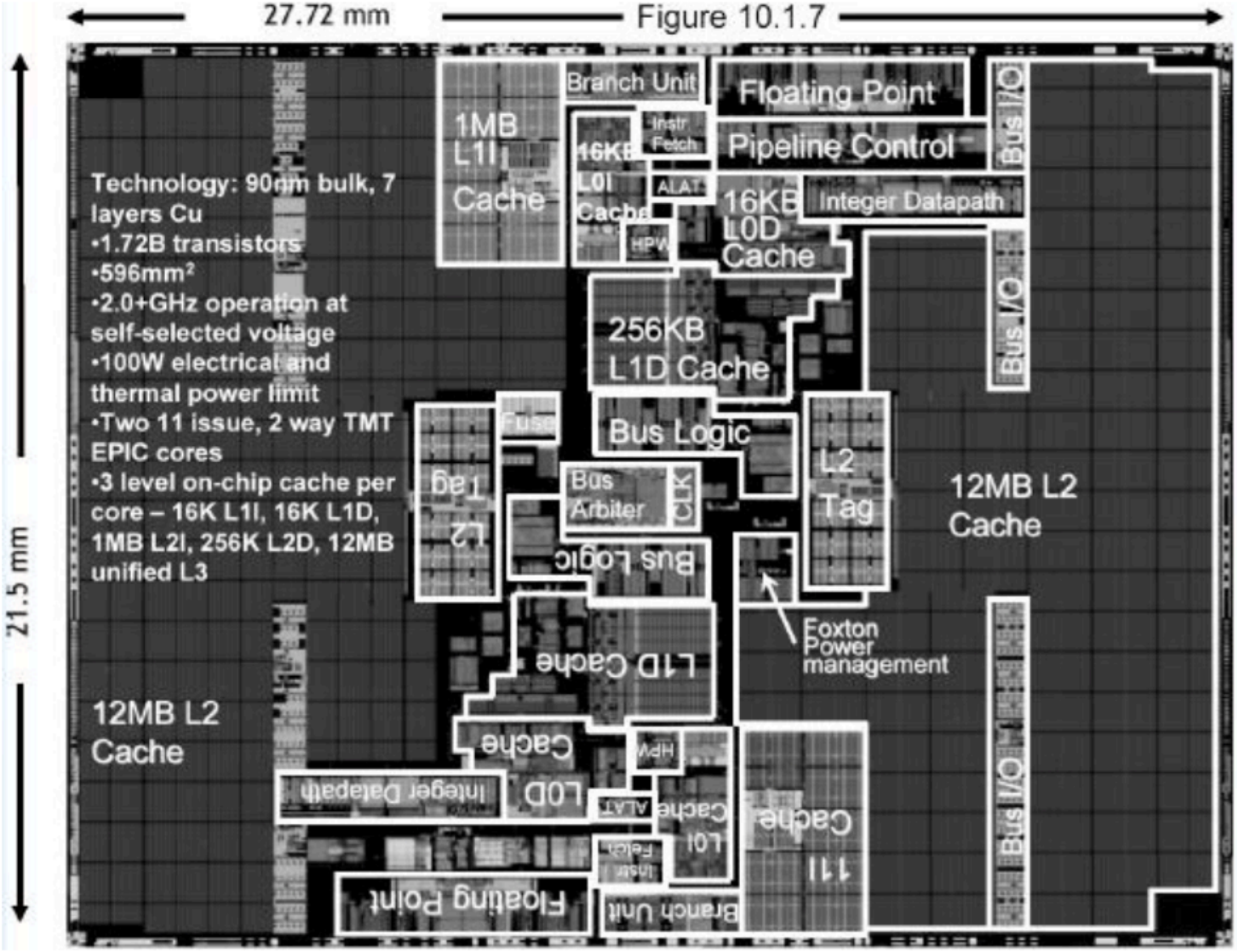
Evolução das arquiteturas

- **Multicore**
  - Colocar dois ou mais "processadores físicos" num mesmo processador/chip.
    - O trabalho de processamento ficará dividido entre os núcleos.
  - O software deve ser escrito para poder suportar este paralelismo.
  - É um tipo de multiprocessamento simétrico (SMP).

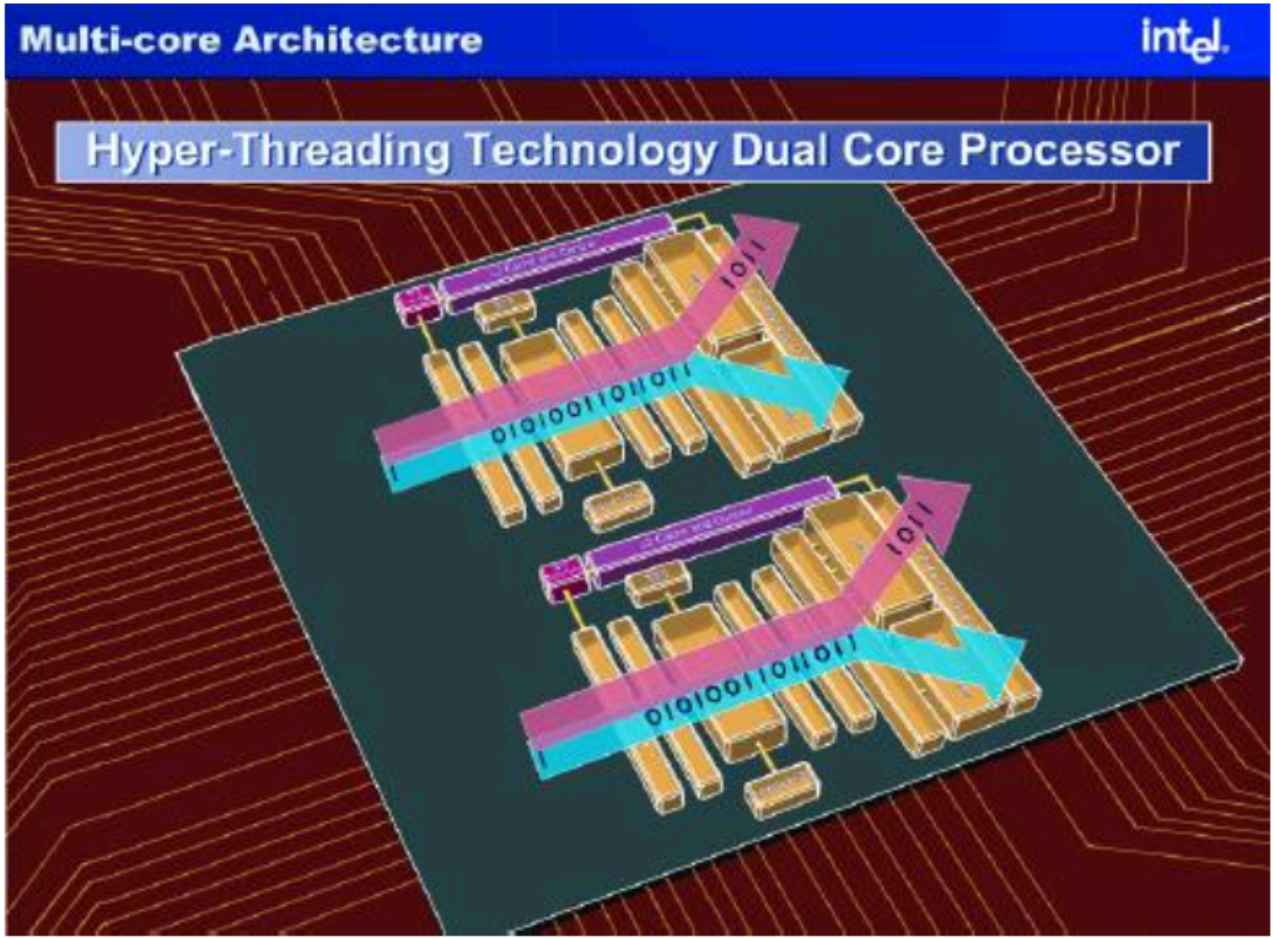
Evolução do Multicore



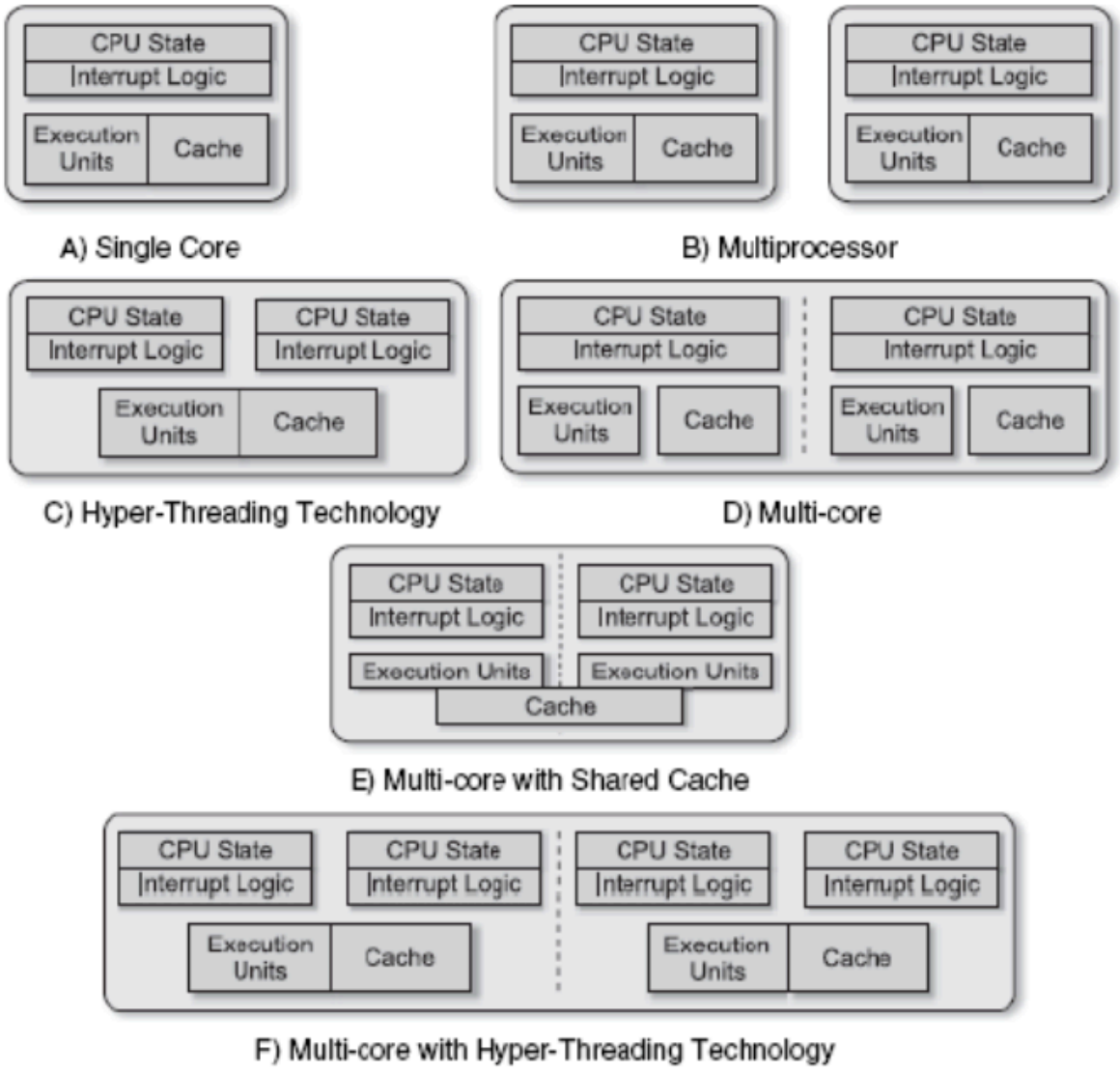
Exemplo de CPU dual core (Montecito)



Multicore


Multi-core Architecture
intel
Hyper-Threading Technology Dual Core Processor

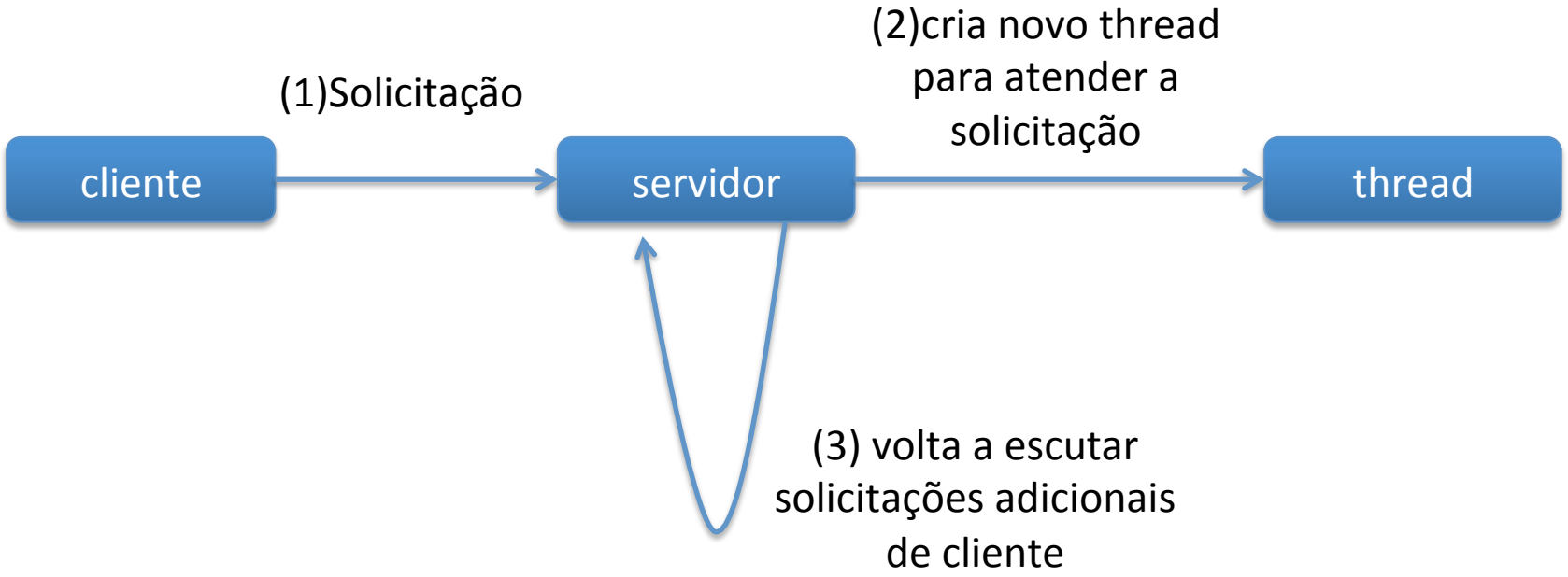
Comparação entre tipos de processamento



Definição de thread

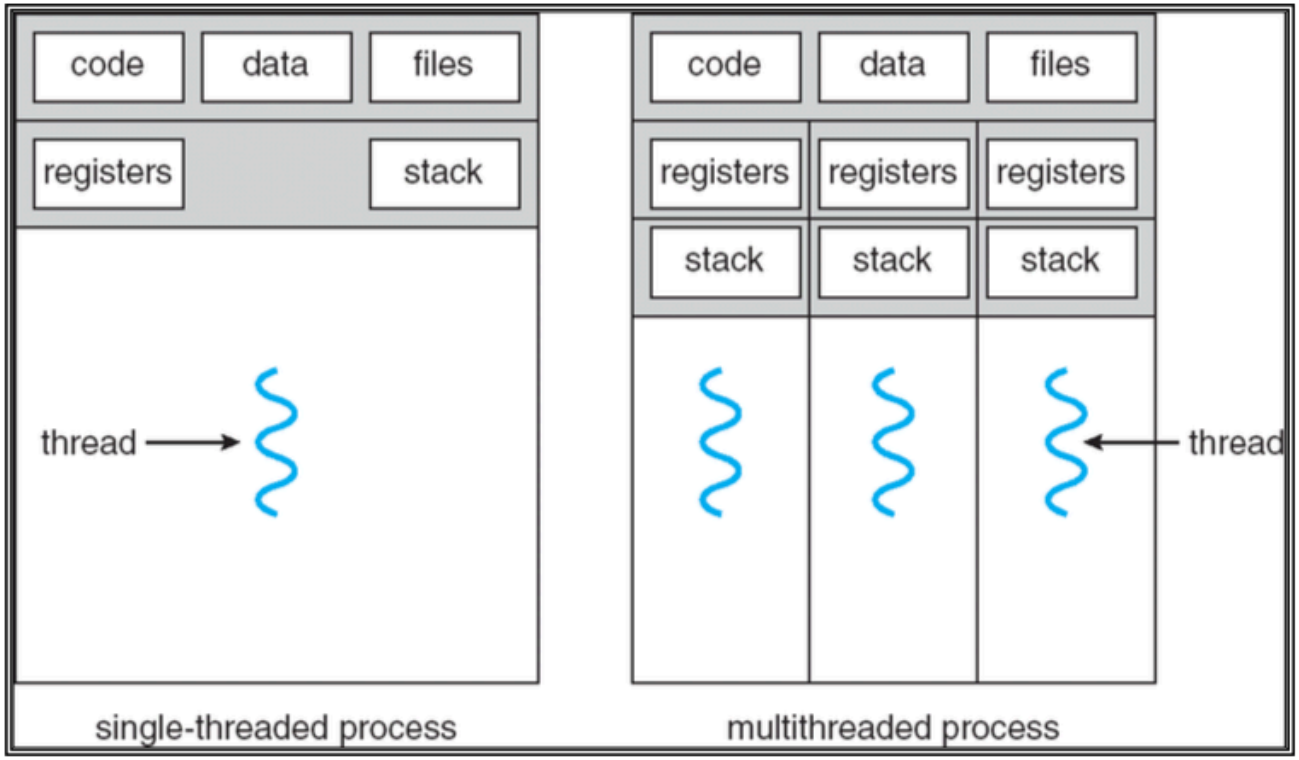
- **Unidade básica de utilização de CPU (linha de execução).**
- **"Subdivisões" de um processo que podem executar concorrentemente.**
  - São chamados por vezes de "processos leves".
- **Compreende um ID de thread, um contador de programa, um conjunto de registradores e uma pilha.**
- **Compartilha com outras threads**
  - Seção de código, seção de dados e outros recursos, pertencentes ao mesmo processo.

Definição de thread

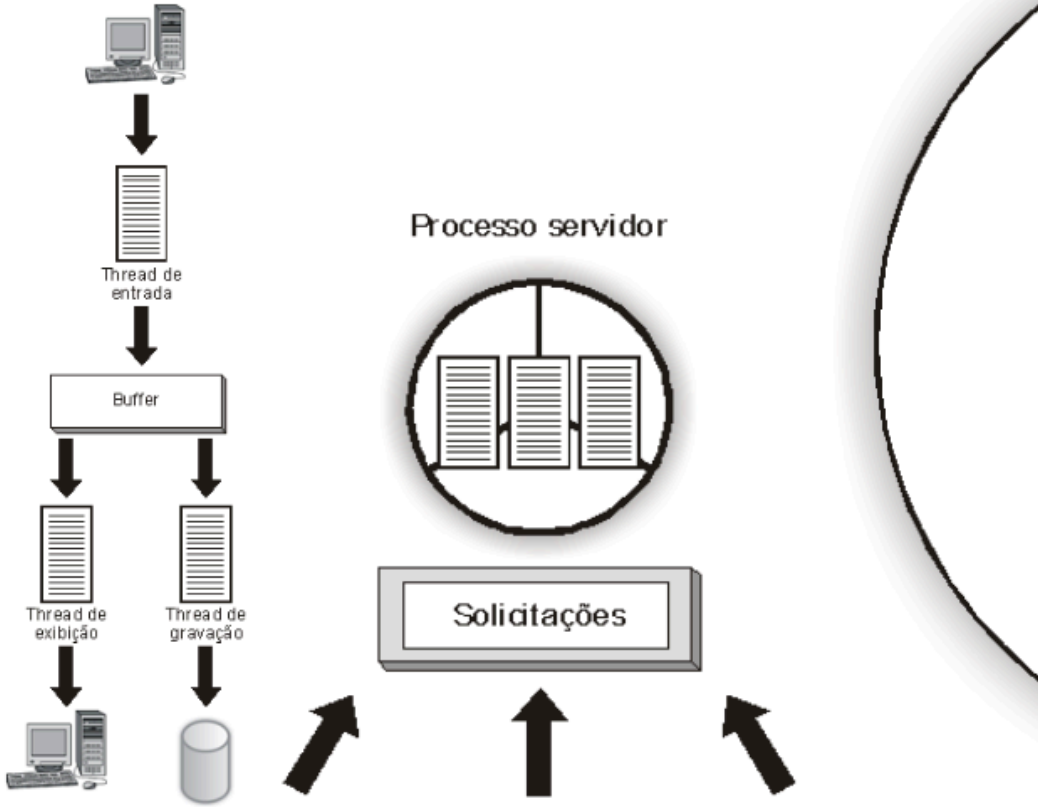


Processos de único e múltiplos threads

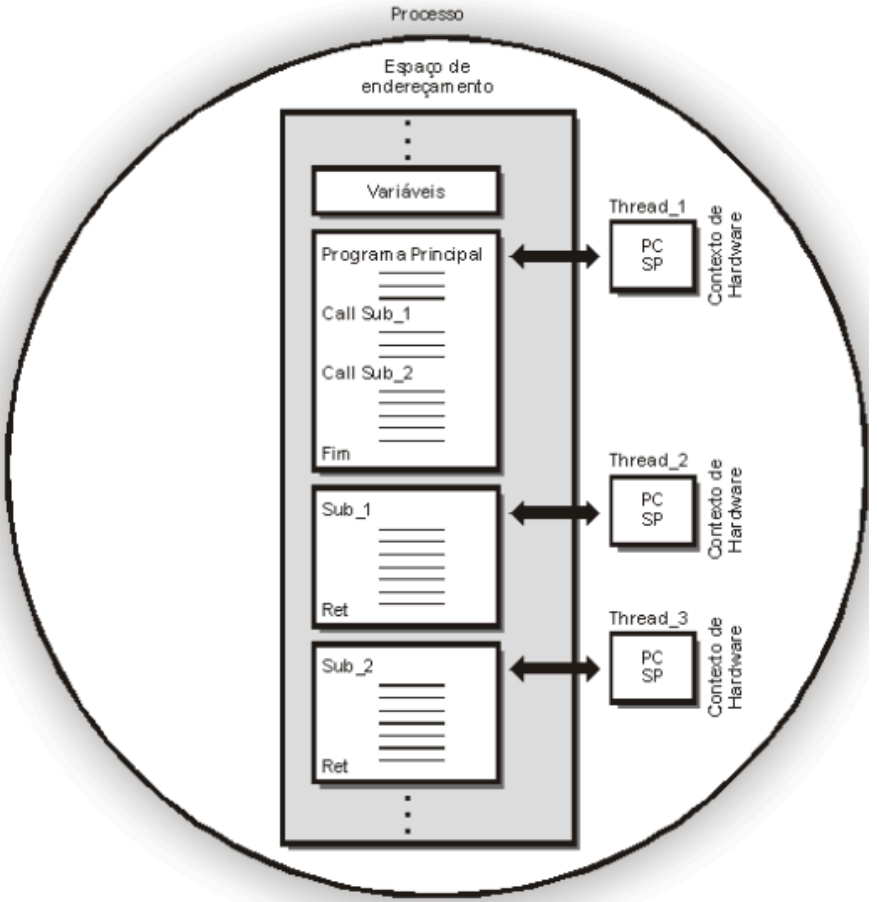
- O thread tem o mesmo contexto de software
- Compartilha o mesmo espaço de memória
- **Mas** o contexto de hardware é diferente



Exemplos


Thread de entrada
Buffer
Thread de exibição
Thread de gravação

Processo servidor
Solicitações


Processo
Espaço de endereçamento
Variáveis
Programa Principal
Call Sub_1
Call Sub_2
Fim
Sub_1
Ret
Sub_2
Ret
Thread_1
PC
SP
Contexto de Hardware
Thread_2
PC
SP
Contexto de Hardware
Thread_3
PC
SP
Contexto de Hardware


Thread
Thread
Thread
Processo cliente
Processo cliente
Processo cliente

Benefícios

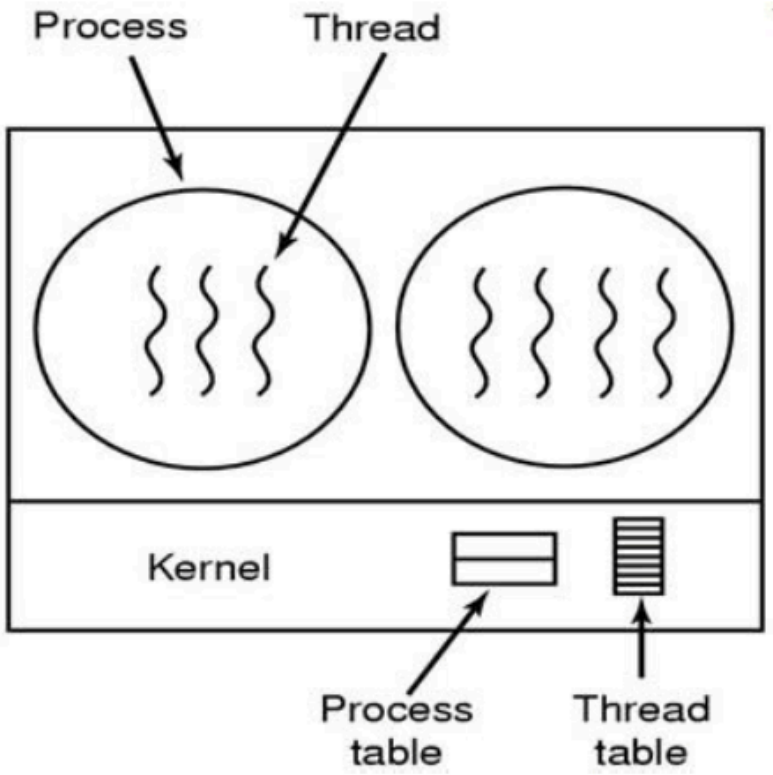
- **Responsividade**
  - Se houver demora ou travamento na execução de uma parte do código.
- **Compartilhamento de recursos**
  - Várias execuções diferentes dentro do mesmo espaço de endereços.
- **Economia**
  - Evita realocação de recursos.
    - Solaris: criação 30x1; troca de contexto 5x1.
- **Utilização de arquiteturas de MP**
  - Permite o uso de mais de um processador.

Threads

- **Motivação**
  - Muitos programas são dotados de funções diferenciadas.
    - Navegador Web
      - Exibir imagens, receber dados da rede...
    - Processador de textos
      - Exibir gráficos, ler teclado, verificar ortografia...
    - Diferentes paradigmas para um serviço
      - Processo tradicional: um cliente por vez.
      - Criação de um processo filho: custo adicional.
      - Solução: criação de threads para atender clientes.
- **Tipos: threads de kernel e threads de usuário**

Threads de kernel

- **Kernel-Level Thread (KLT)**
- **Admitidos pelo kernel diretamente**
  - A gestão dos threads é feita pelo SO.
- **Multiprocessamento mais efetivo**
- **Ao escalonar, é preciso mudar o modo da CPU**
  - Aumenta o overhead.



Threads de usuário

- **User-Level Thread (ULT)**
- **Implementada por uma biblioteca de linguagem**
  - Pode rodar num SO sem threads.
- **Não é preciso mudar o modo da CPU**
  - Diminui o overhead.
  - Programador que escalona.
- **Possuem as mesmas operações dos KLT**
  - Criar, terminar e join, etc.