

Threads



UNIFEI

Universidade Federal de Itajubá

IMC – Instituto de Matemática e Computação

Prof. Carlos Minoru Tamaki

Threads

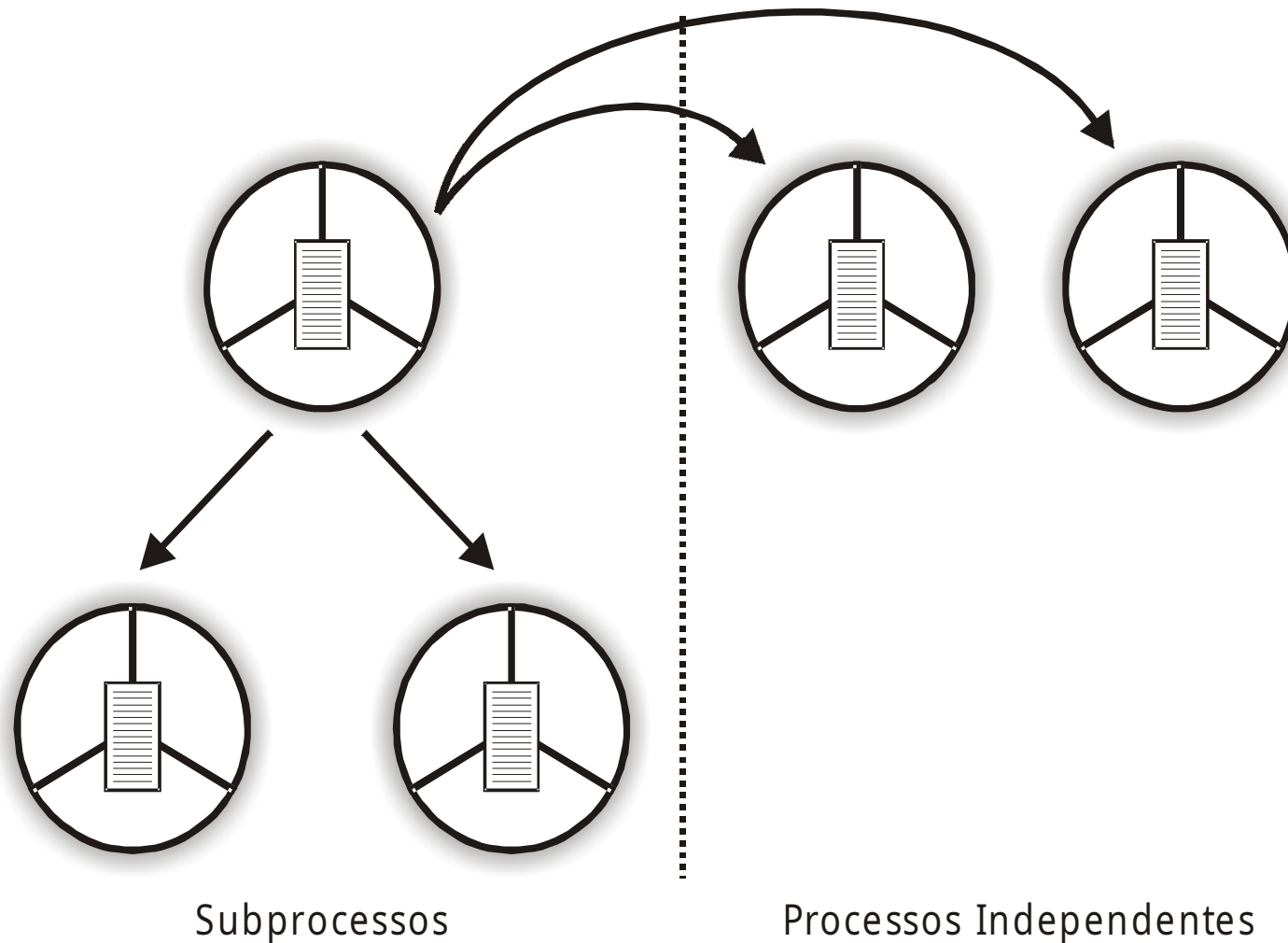
- Até o final da década de 1970, SO's, como Tops-10 (DEC), MVS (IBM) e Unix (Bell Labs) - um único thread (monothread)
- Em 1979, SO Toth, foi introduzido o conceito de processo lightweight (peso leve), onde o espaço de endereçamento de um processo era compartilhado por vários programas.

- Com o conceito de multithread é possível projetar e implementar aplicações concorrentes de forma eficiente, pois o processo pode ter partes diferentes de seu código sendo executadas concorrentemente, com um menor overhead do que utilizando múltiplos processos.
- O desenvolvimento de programas que explorem os benefícios da programação multithread não é simples. A presença do paralelismo introduz um novo conjunto de problemas, como comunicação e sincronização de threads.
- Atualmente, o conceito de multithread pode ser encontrado em diversos SO's como no Solaris da Sun e Windows da Microsoft.

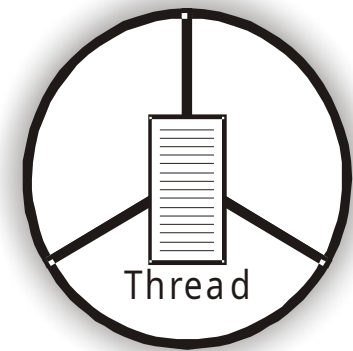
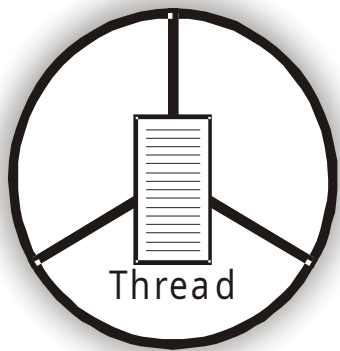
Ambiente Monothread

- Um programa é uma sequência de instruções
 - desvios, repetições e chamadas a procedimentos e funções.
- Em um ambiente monothread, um processo suporta apenas um programa no seu espaço de endereçamento.

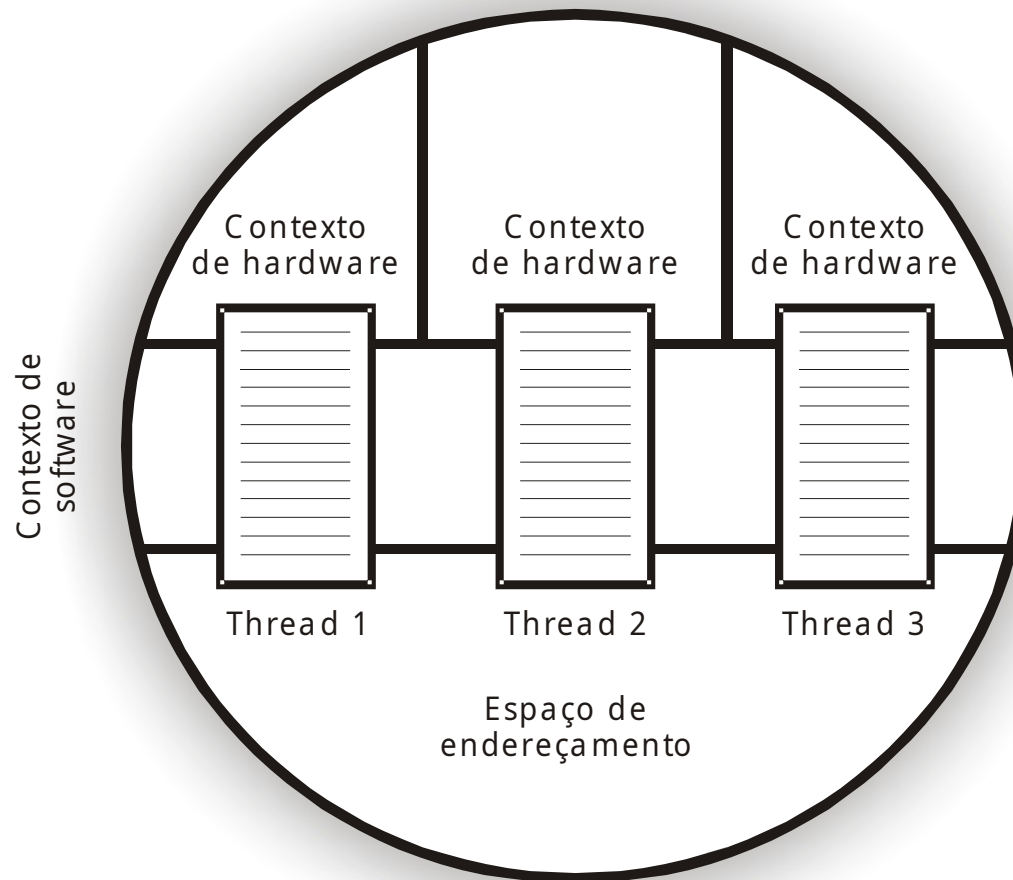
Subprocessos e processos independentes



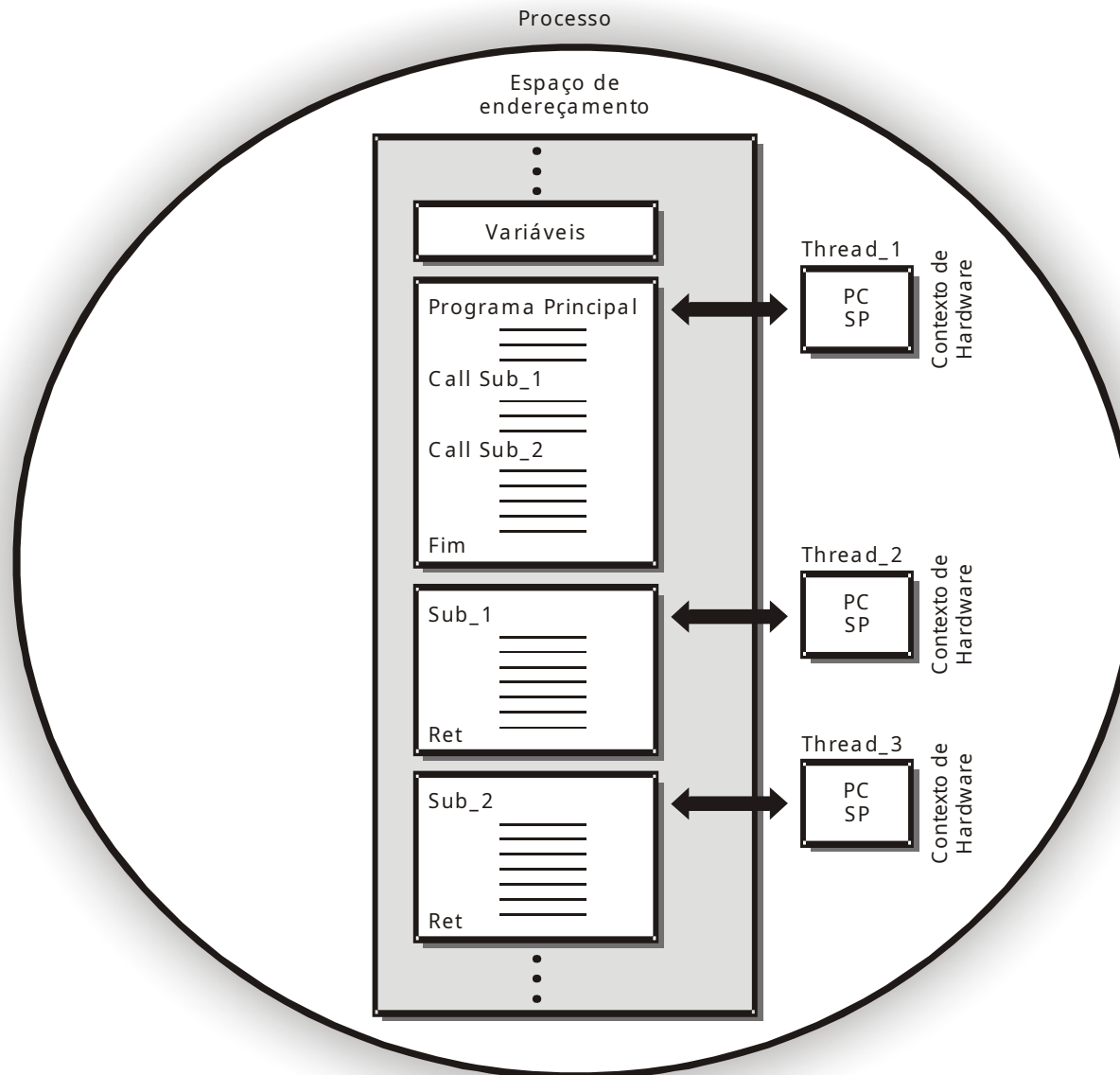
Ambiente Monothread



Ambiente Multithread



Aplicação Multithread



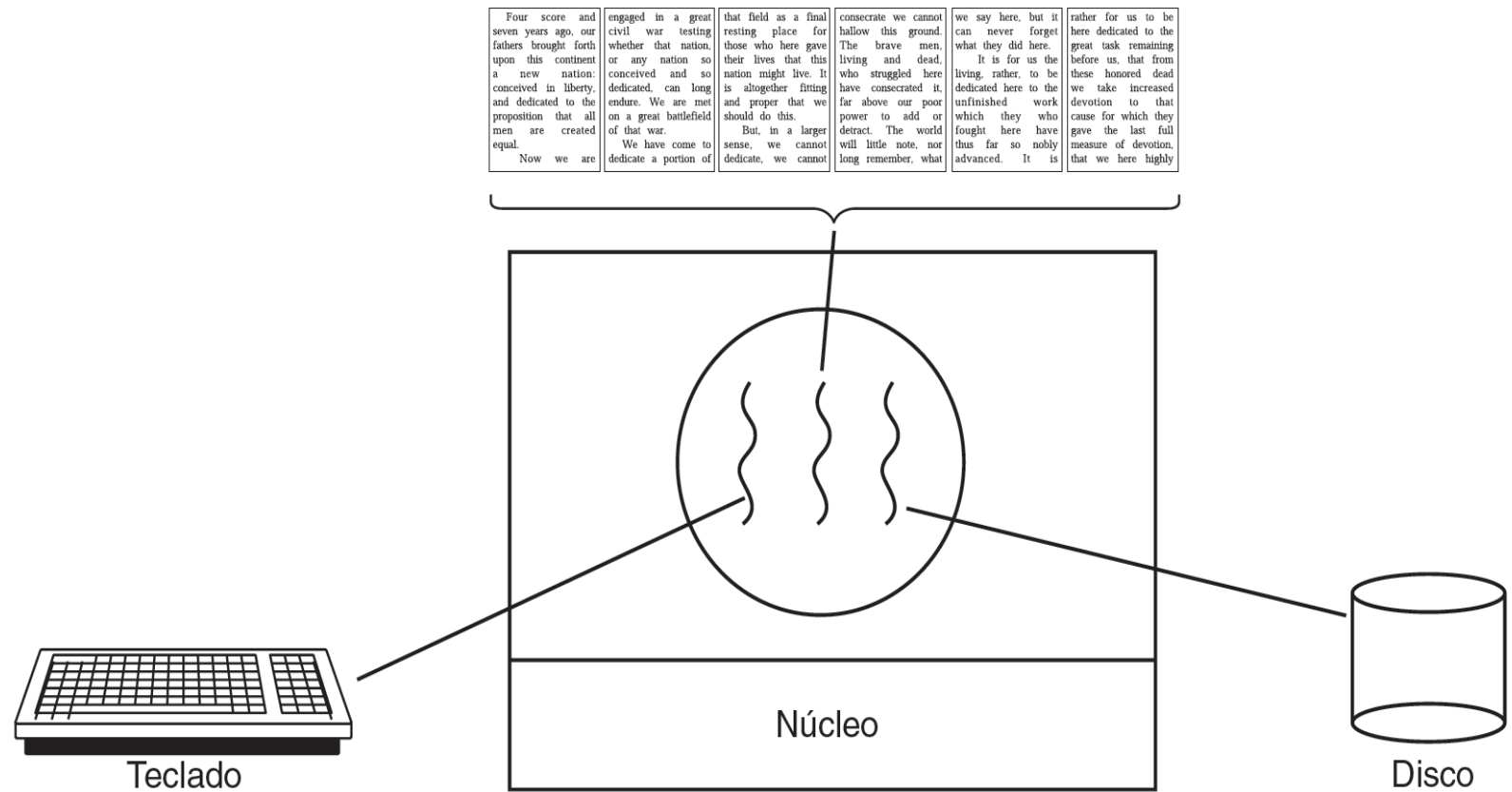
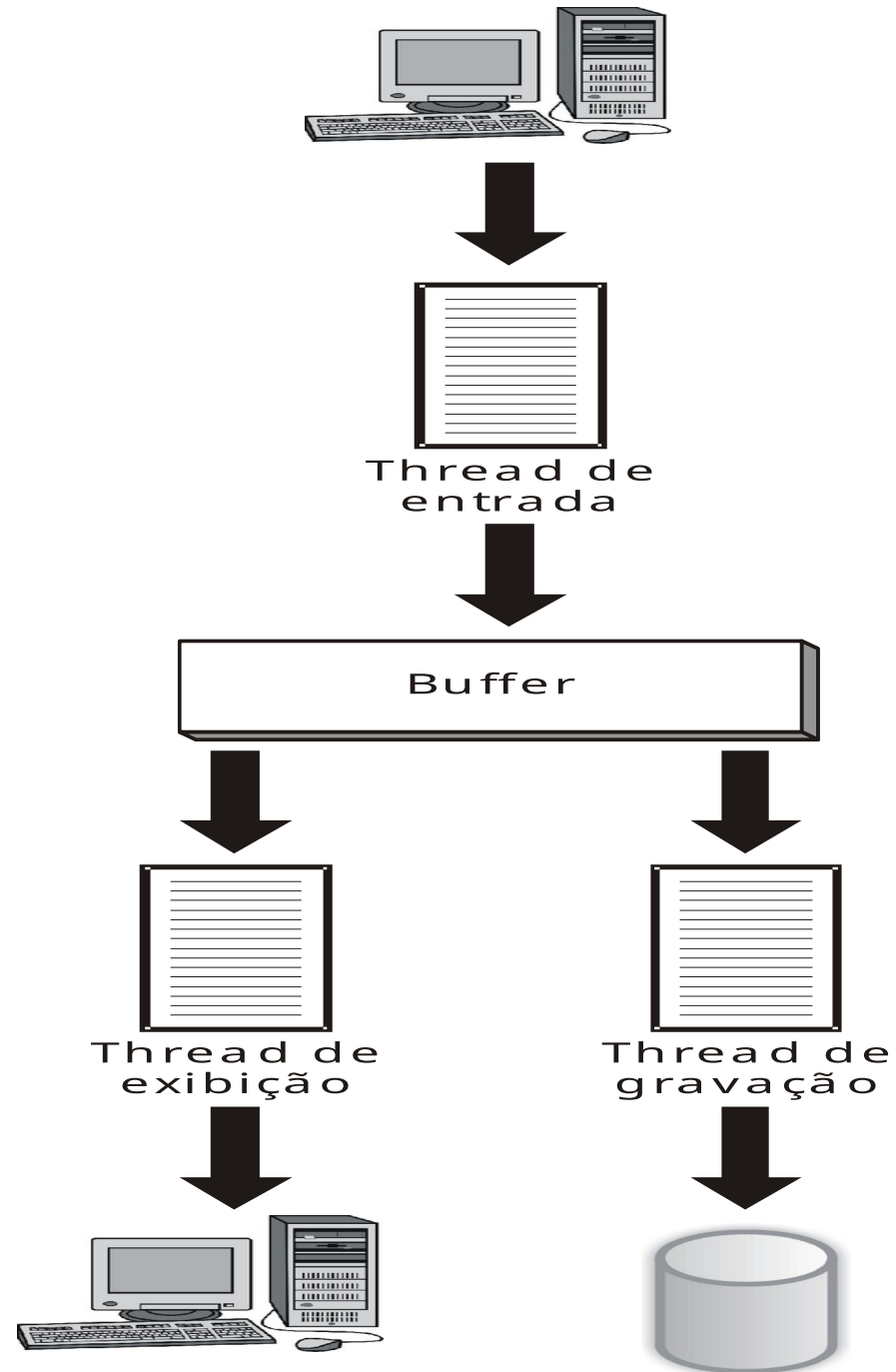


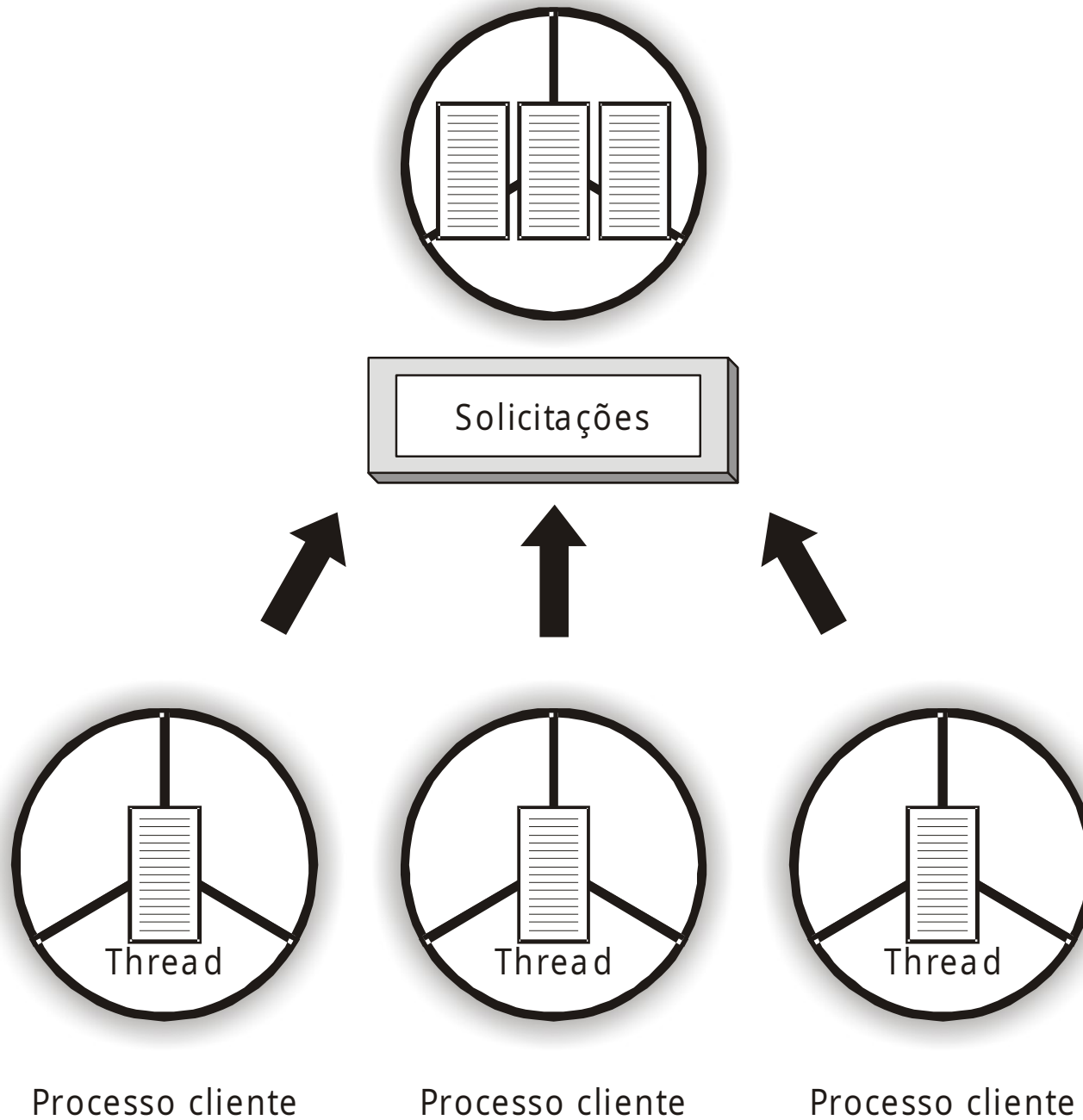
Figura 2.5 Um processador de textos com três threads.

Implementação	Tempo de Criação (μS)	Tempo de Sincronização (μS)
Processo	1700	200
Processo Lightweight	350	390
Thread	52	66

Latência de processos e threads
(Vahalia, 1996)



Processo servidor



Modelo Clássico de Threads

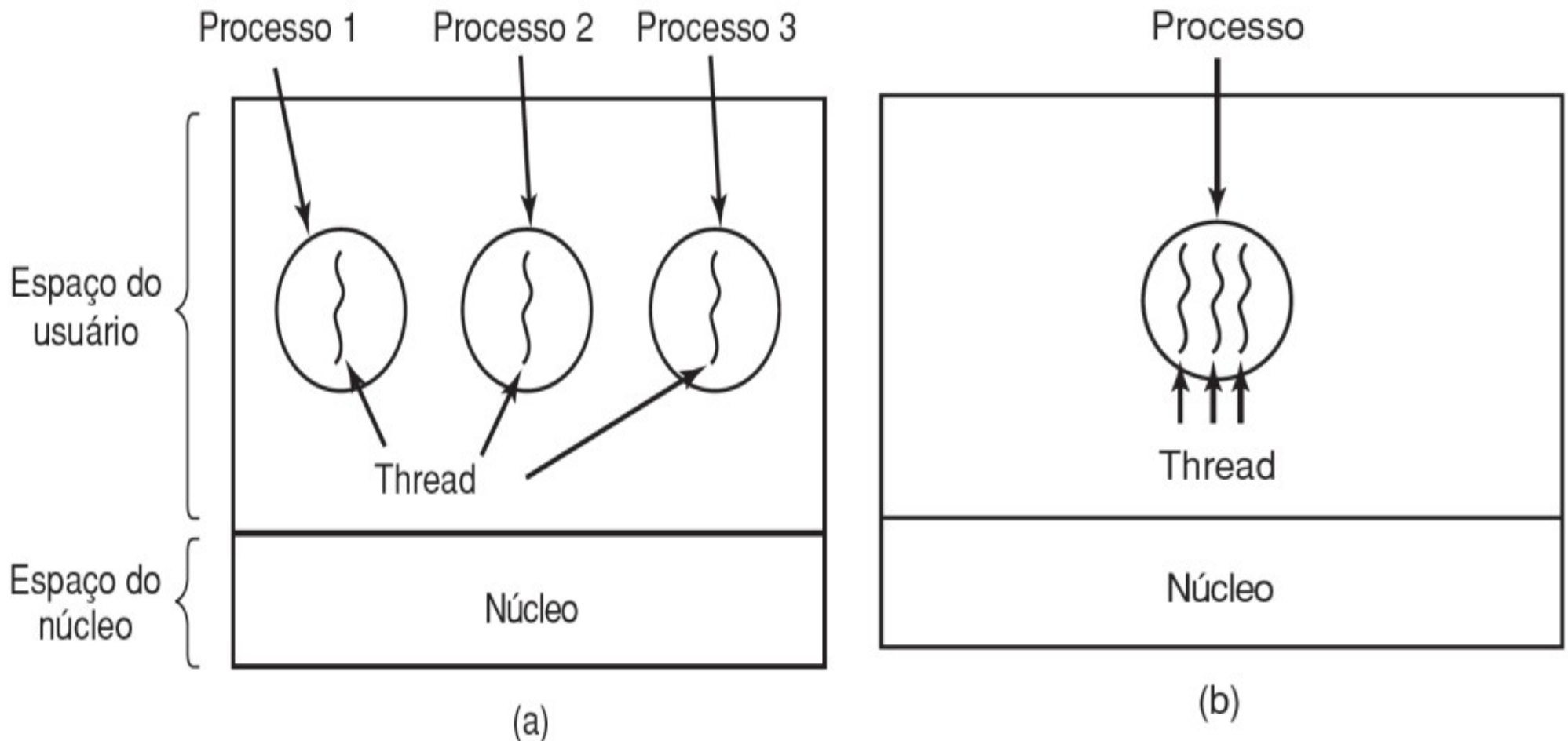


Figura 2.8 (a) Três processos, cada um com um thread.
(b) Um processo com três threads.

Itens por processo	Itens por thread
Espaço de endereçamento	Contador de programa
Variáveis globais	Registradores
Arquivos abertos	Pilha
Processos filhos	Estado
Alarmes pendentes	
Sinais e manipuladores de sinais	
Informação de contabilidade	

Tabela 2.4 A primeira coluna lista alguns itens compartilhados por todos os threads em um processo. A segunda lista alguns itens específicos a cada thread.

Programação Multithread

```
program ADD1( Input, Output );  
var X : integer;
```

```
procedure p1;  
var k : integer;  
begin  
  for k := 1 to 50 do X := X + 1;  
end;
```

```
procedure p2;  
var k : integer;  
begin  
  for k := 1 to 50 do X := X + 1;  
end;
```

```
begin { main }  
  X := 0;  
  cobegin  
    p1; p2  
  coend;  
  writeln( X );  
end.
```

Programa escrito em Co-Pascal

Arquitetura e Implementação

<i>Ambientes</i>	<i>Arquitetura</i>
Distributed Computing Environment (DCE)	Modo usuário
Compaq Open VMS versão 6	Modo usuário
Microsoft Windows 2000	Modo kernel
Compaq Unix	Modo kernel
Compaq Open VMS versão 7	Modo kernel
Sun Solaris versão 2	Modo híbrido
University of Washington FastThreads	Scheduler activations

Threads em Modo Usuário



Threads em Modo Kernel

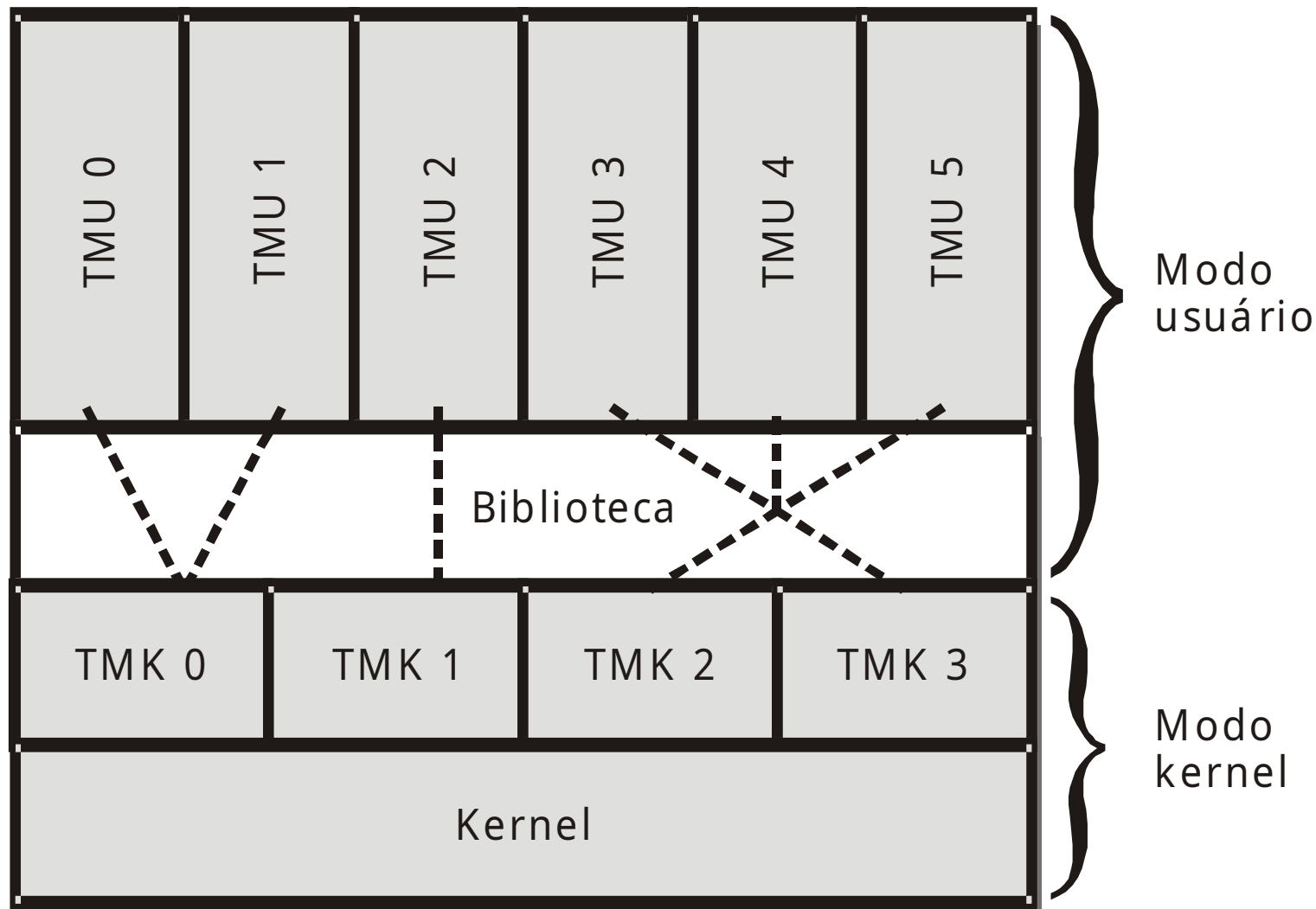


Tempos de Latência

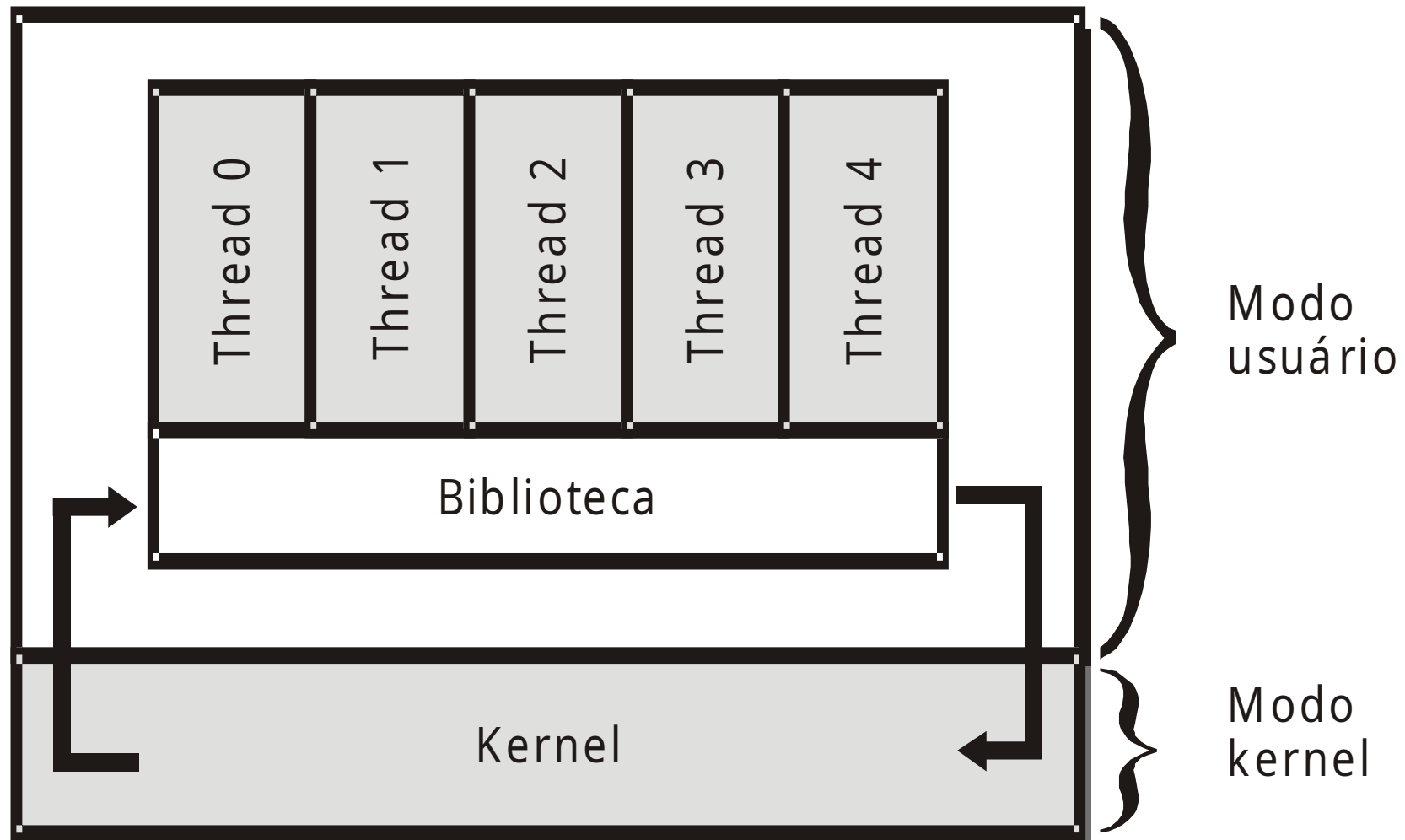
<i>Implementação</i>	<i>Operação 1 (μS)</i>	<i>Operação 2 (μS)</i>
Subprocessos	11.300	1.840
Threads em modo kernel	948	441
Threads em modo usuário	34	37

Comparação entre tempos de latência
(Anderson et al., 1992)

Threads em Modo Híbrido



Scheduler Activations



Threads Posix

- Criação de programas com threads portáteis, com padrão **IEEE 1003.1c**
- Pacote de threads denominado **Pthreads**
- Mais de 60 chamadas de funções

Algumas chamadas da PThreads

Chamada de thread	Descrição
pthread_create	Cria um novo thread
pthread_exit	Conclui a chamada de thread
pthread_join	Espera que um thread específico seja abandonado
pthread_yield	Libera a CPU para que outro thread seja executado
pthread_attr_init	Cria e inicializa uma estrutura de atributos do thread
pthread_attr_destroy	Remove uma estrutura de atributos do thread

■ **Tabela 2.5** Algumas das chamadas de função de Pthreads.

Exemplo de PThreads

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#define NUMBER OF THREADS 10
void *print hello world(void *tid)
{
    /* Esta funcao imprime o identificador do
    thread e sai. */
    printf("Ola mundo. Boas vindas do thread %d\
n", tid);
    pthread exit(NULL);
}
int main(int argc, char *argv[ ])
{
    /* O programa principal cria 10 threads e sai.
    */
```

```
pthread t threads[NUMBER OF THREADS];
int status, i;
for(i=0; i < NUMBER OF THREADS; i++) {
    printf("Metodo Main. Criando thread %d\n",
i);
    status = pthread create(&threads[i], NULL,
print hello world, (void *)i);
    if (status != 0) {
        printf("Oops. pthread create returned error
code %d\n", status);
        exit(-1);
    }
}
exit(NULL);
}
```


Compilar um programa com PThreads

- Programas podem ser escritos em C/C++
- Para compilação utilize:
 - gcc -Wall -o nome nome.c **-lpthread**

Dúvidas?