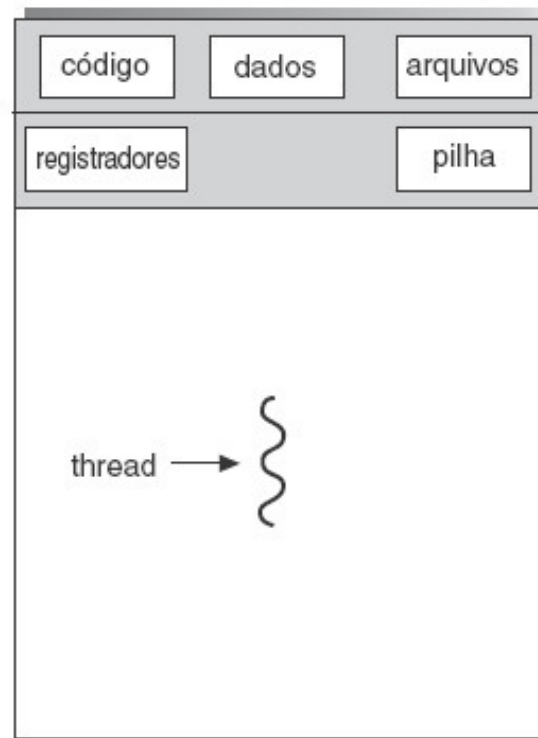


Sistemas Operacionais I

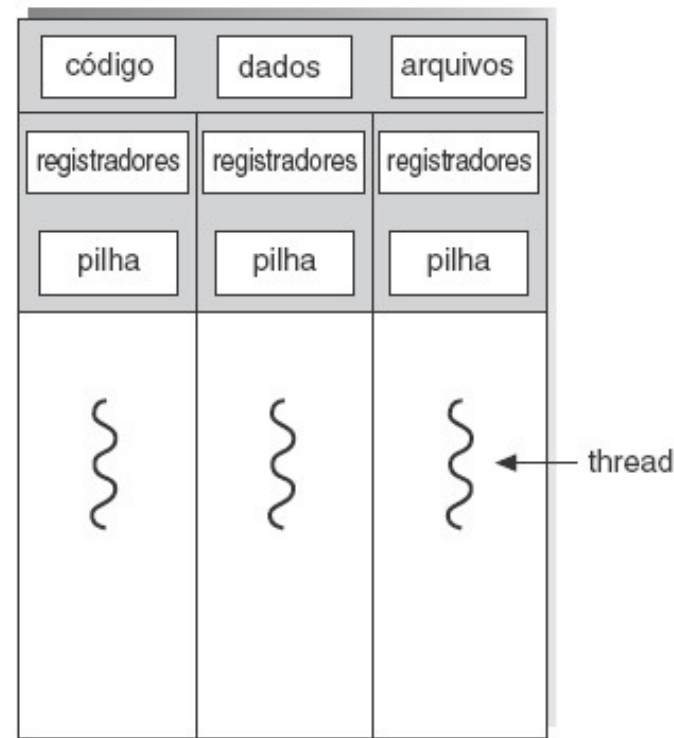
Threads

Prof. Carlos Eduardo de B. Paes
Departamento de Ciência da Computação
Pontifícia Universidade Católica de São Paulo

Processos com Thread Única e Múltithread



processo dotado de única thread



processo dotado de múltiplas threads

Vantagens

- Responsividade
- Compartilhamento de recursos
- Economia
- Utilização de arquiteturas multiprocessadas

Threads de Usuário

- Gerenciadas por biblioteca de threads no nível do usuário
- Três bibliotecas de threads principais:
 - POSIX Pthreads
 - Threads Java
 - Threads Win32

Threads de kernel

- Suportado pelo kernel
- Exemplos
 - Windows XP/2000
 - Solaris
 - Linux
 - Tru64 UNIX
 - Mac OS X

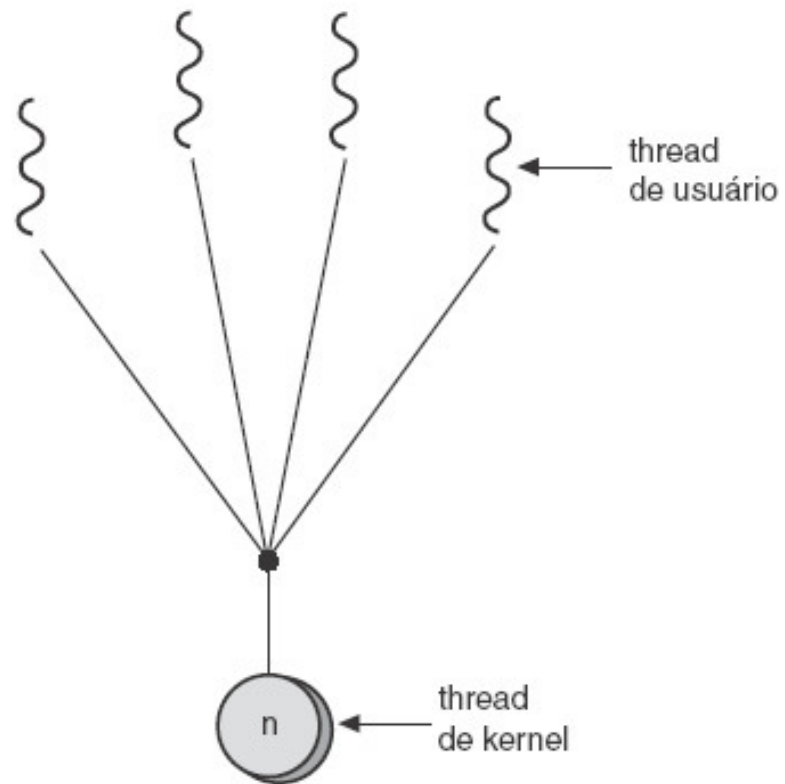
Modelos Multithreads

- Muitos-para-um
- Um-para-um
- Muitos-para-muitos

Muitos-para-Um

- Muitas threads no nível do usuário associadas a uma única thread de kernel
- Exemplos
 - Solaris Green Threads
 - GNU Portable Threads

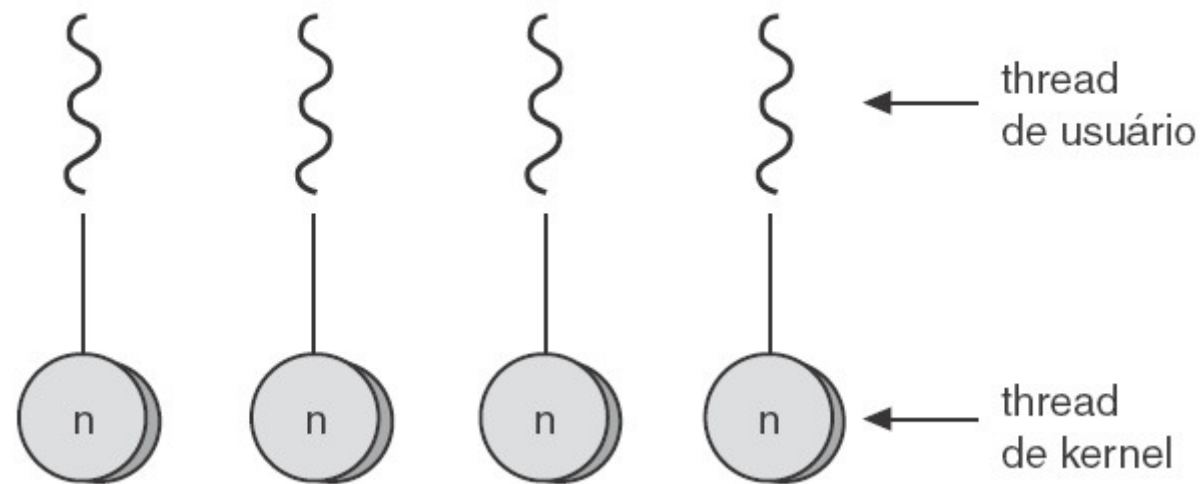
Modelo Muitos-para-Um



Um-para-Um

- Cada thread do usuário associada a uma thread de kernel
- Exemplos
 - Windows NT/XP/2000
 - Linux
 - Solaris 9 e acima

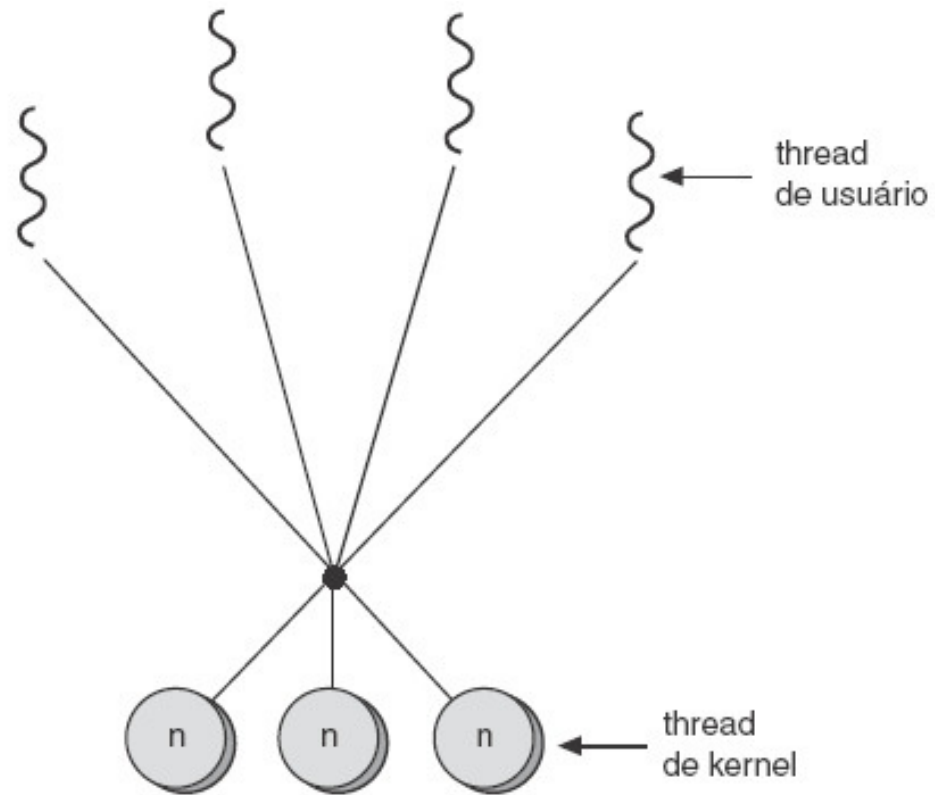
Modelo Um-para-Um



Muitos-para-Muitos

- Permite que muitas threads no nível do usuário sejam associadas a muitas threads no nível do kernel
- Permite que o sistema operacional crie um número suficiente de threads de kernel
- Exemplos
 - Solaris antes da versão 9
 - Windows NT/2000 com o pacote *ThreadFiber*

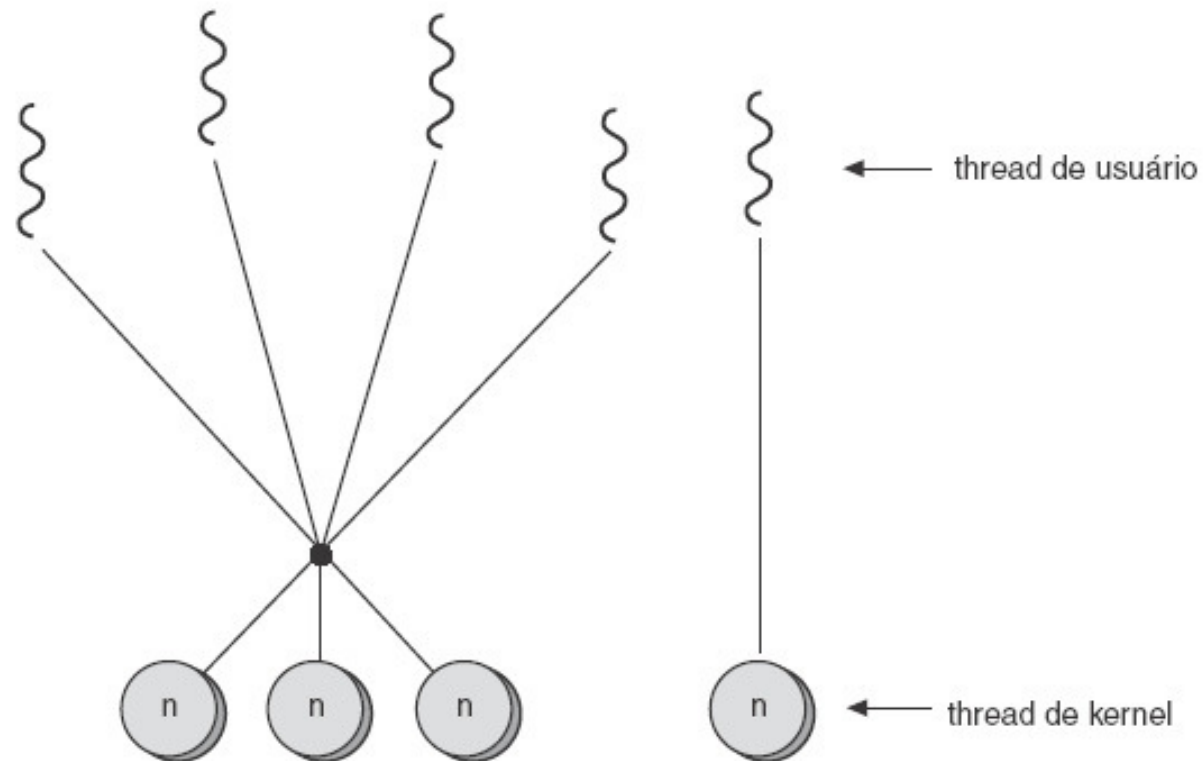
Modelo Muitos-para-Muitos



Modelo em Dois Níveis

- Semelhante ao modelo M:M, exceto que permite que uma thread do usuário seja associada a thread do kernel
- Exemplos
 - IRIX
 - HP-UX
 - Tru64 UNIX
 - Solaris até a versão 8

Modelo em Dois Níveis



Aspectos do Uso de Threads

- Semântica das chamadas de sistema **fork()** e **exec()**
- Cancelamento da thread
- Tratamento de sinais
- Bancos de threads
- Dados específicos da thread
- Ativações de escalonador

Threads em Java

- São gerenciadas pela JVM
- Podem ser criadas através de:
 - Extensão da classe Thread
 - Implementação da interface Runnable

Estendendo a classe Thread

```
class Worker1 extends Thread
{
    public void run() {
        System.out.println("Eu sou a thread trabalhadora");
    }
}

public class First
{
    public static void main(String args[]) {
        Worker1 runner = new Worker1();
        runner.start();
        System.out.println("Eu sou a Thread principal");
    }
}
```

A interface Runnable

```
public interface Runnable  
{  
    public abstract void run();  
}
```

Implementando a interface Runnable

```
class Worker2 implements Runnable
{
    public void run() {
        System.out.println("Eu sou a Thread trabalhadora ");
    }
}

public class Second
{
    public static void main(String args[]) {
        Runnable runner = new Worker2();
        Thread thrd = new Thread(runner);
        thrd.start();
        System.out.println("Eu sou a thread principal");
    }
}
```

Estados da thread

