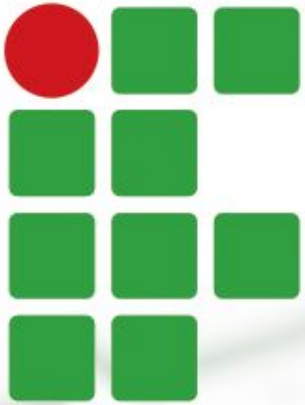


INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Sistemas Distribuídos

- Conceitos Básicos -

Conceitos Básicos

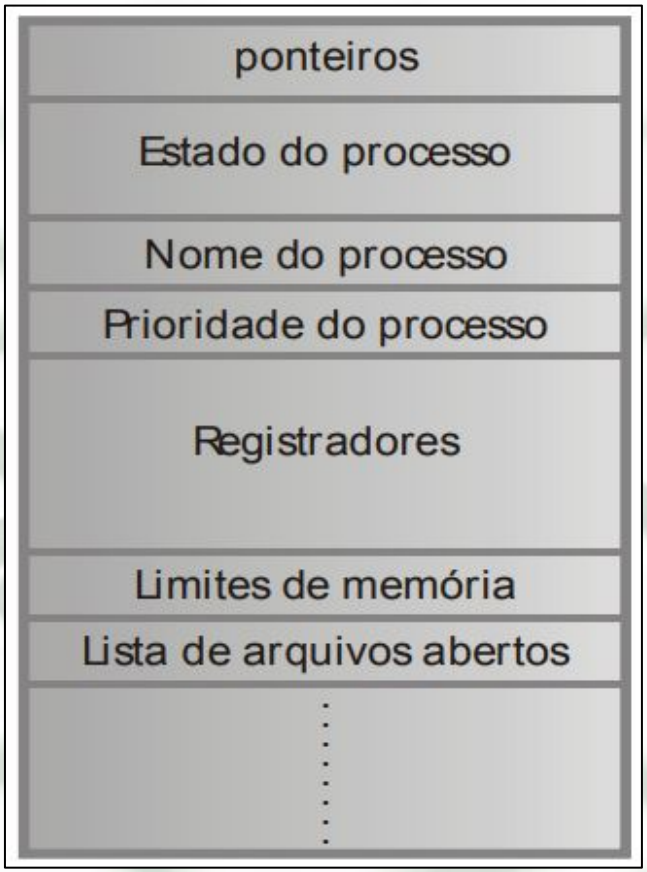
- Processo
- Threads
- Concorrência
- Paralelismo
- Computação Centralizada
- Computação Distribuída
- Comunicação Síncrona
- Comunicação Assíncrona

Processo

- Processos são, resumidamente, **programas em execução**! (*aka*. *job, task*)
- Estudo do ponto de vista de Sistemas Operacionais:
  - Gerenciamento dos Processos (Início, Meio e Fim)
  - Escalonamento de Processos (Trocas de Contexto)
- Estudo do ponto de vista dos **Sistemas Distribuídos**:
  - Como aproveitar ao máximo os recursos computacionais disponíveis para atender os processos?

PCB

- ***Process Control Block***
  - Estrutura de Dados mantida pelo S.O para cada processo executado.
  - Armazena todas as informações necessárias para a execução.
  - Totalmente independentes entre si.
  - Grande quantidade de dados atualizada a cada instante.
- Os PCB residem na memória principal do dispositivo => ***Kernel Space*** (Memória dedicada ao S.O.)

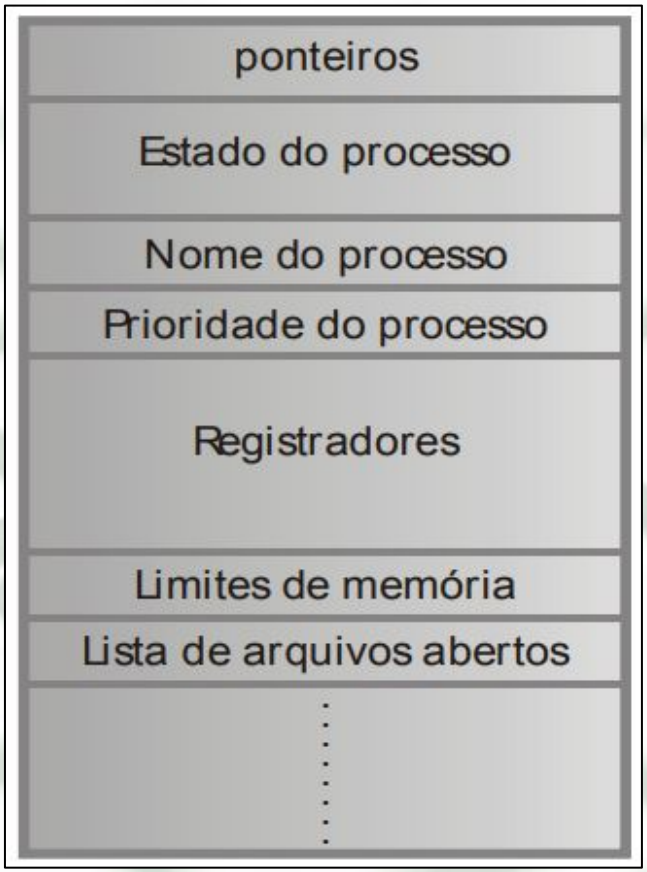


INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

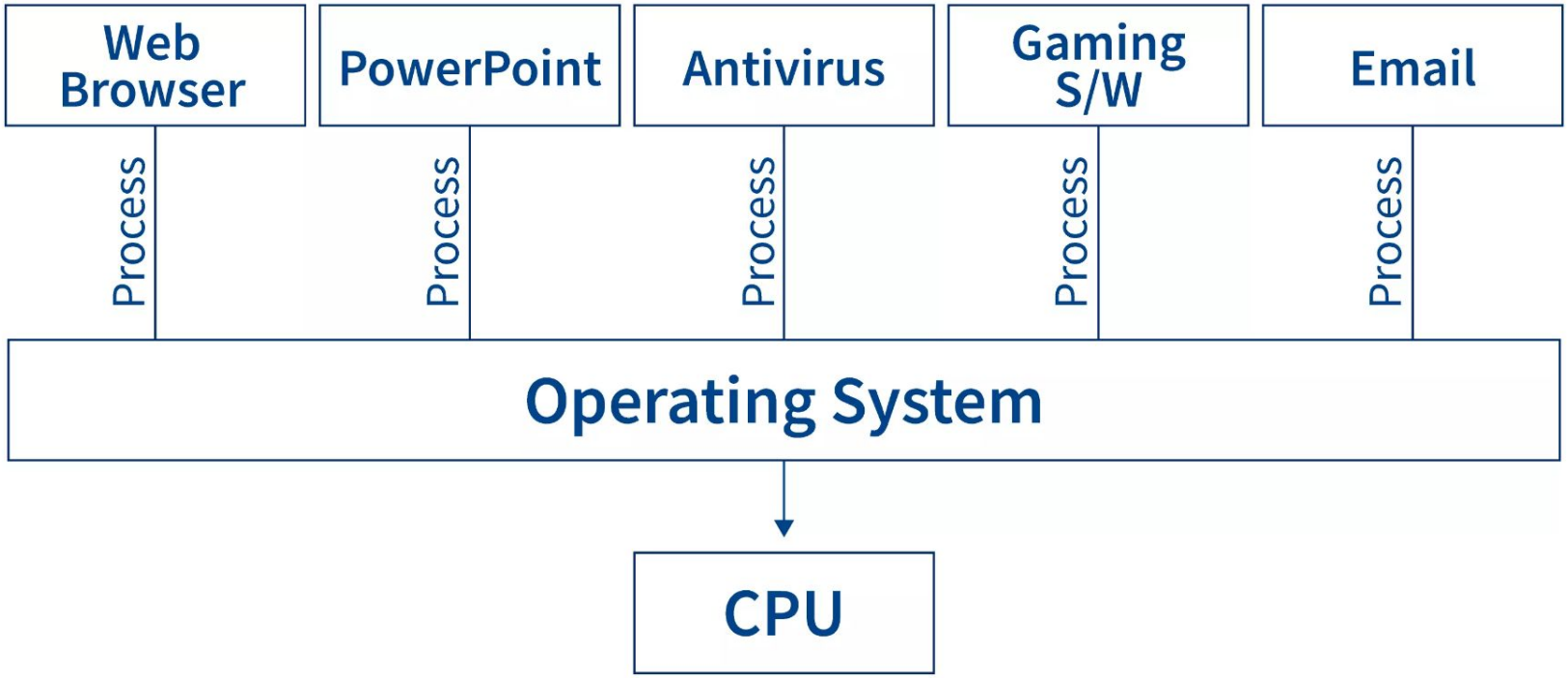
PCB

- **Ao sistema operacional cabe a tarefa de gerenciar os PCBs...**
- Os processos devem compartilhar os *hardwares* entre si (memória, CPU, disco, I/O, rede, etc...)
- Este compartilhamento é feito **pela alternância dos PCBs em execução num dado instante de tempo**.

SISTEMA OPERACIONAL MULTITAREFAS



Multitarefas

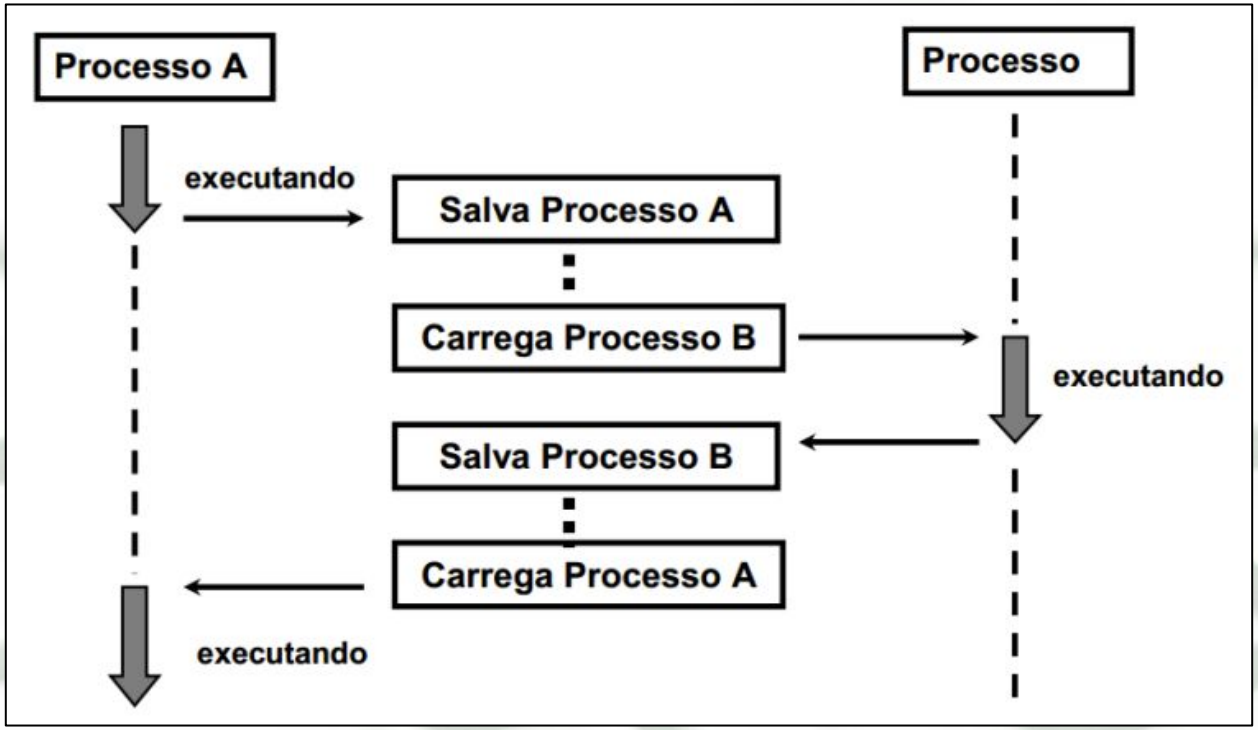


Troca de Contexto

- **Troca de Contexto** é o procedimento de alternar os processos em execução na CPU.
- Cada CPU (core) executa apenas um processo por vez.
- Isso gera uma sobrecarga de operações custosas (*overhead*) para o sistema (é o preço pago para ter um sistema multitarefas).
- S.O. deve colocar e retirar processos em execução:
  - **Ao retirar**, deve guardar todo o estado atual do PCB na memória RAM => *kernel space*.
  - **Ao colocar**, deve restaurar todo o PCB da memória RAM para execução na CPU.

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Troca de Contexto



Sistemas Atuais: de 100 a 1000 trocas / segundo.

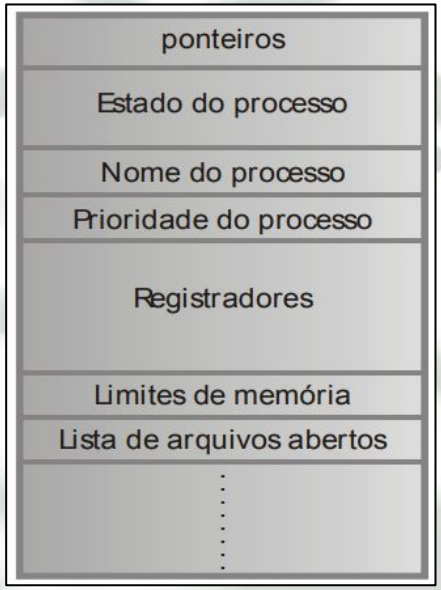
REALMENTE INCRÍVEL!!!

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Informações do PCB

- Estado
  - Ativo/Executando => Na CPU
  - Pronto/Esperando => Aguardando sua vez
  - Em espera/Bloqueado => Aguardando I/O, Sync.
  - Inativo/Zumbi => Finalizado
- PID (Process ID)
- Usuário do Processo
- Prioridade de Execução
  - [-20, +20] #Conceito *Nice do Linux*

Etc...

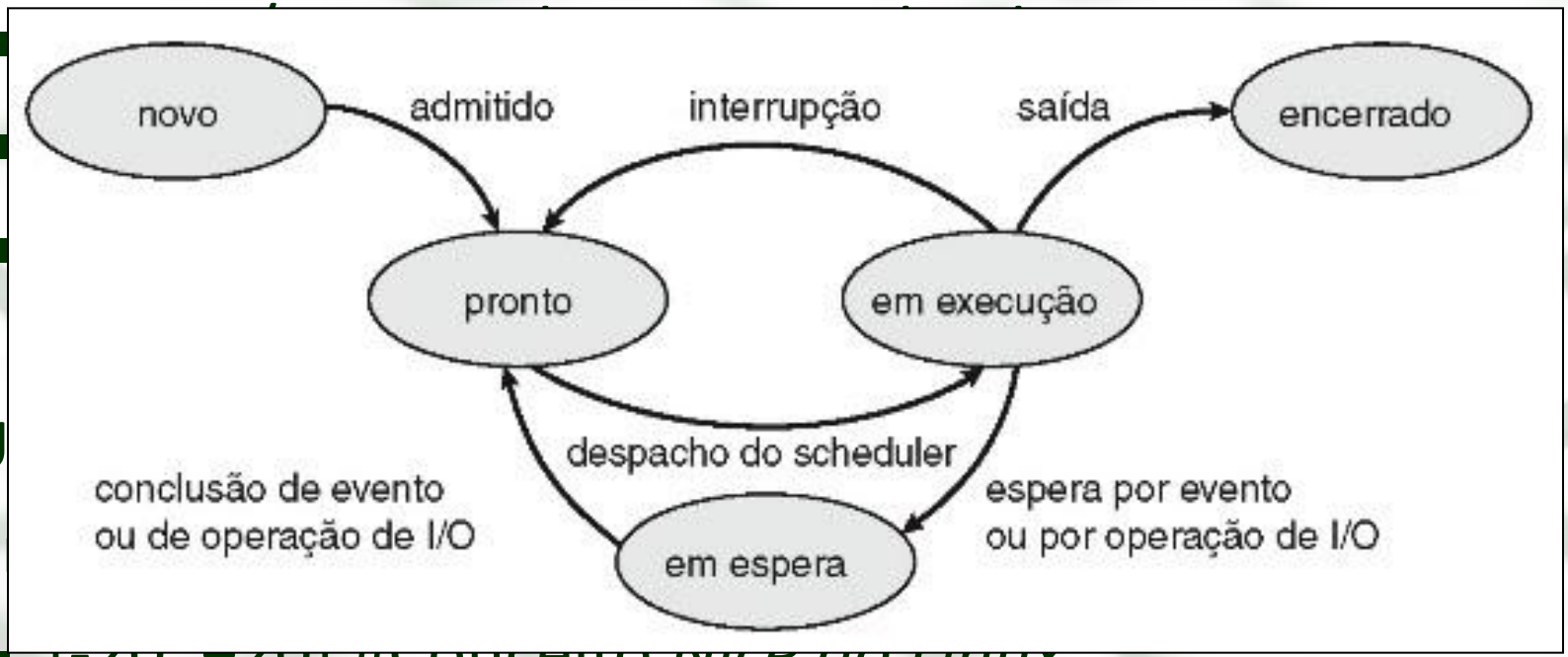


INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Estado do Processo

- Estado
  - Ativo/Executando => Na CPU
  -
  -
  -
- P
- U
- P
  - [-20, +20] #Conceito *Nice do Linux*

Etc...



INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Gerenciador de Processos

- E
- P
- N
- P

Etc.

adriano@adriano-All-Series: ~

Arquivo Editar Ver Pesquisar Terminal Ajuda

```
  1  [|||                                    7.9%]   Tasks: 187, 815 thr; 1 running
  2  [|||||||                               18.7%]   Load average: 0.71 0.63 0.55
  3  [|||||||||                             24.7%]   Uptime: 03:43:20
  4  [||||                                  10.0%]
  Mem[||||||||||||||||||||5.39G/7.70G]
  Swp[|                                    24.5M/2.00G]

    PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
  14679 adriano    20   0 5229M 1728M 1646M S 11.3 21.9 42:22.62 /usr/lib/virtualb
  14723 adriano    20   0 5229M 1728M 1646M S  9.3 21.9 37:26.13 /usr/lib/virtualb
  15008 adriano    20   0 32.6G  364M  120M S  3.3  4.6  1:12.15 /opt/google/chrom
   2796 adriano    20   0 3882M  232M 47492 S  3.3  2.9 16:27.15 /usr/bin/gnome-sh
   2853 adriano     9 -11 2327M 16232 12024 S  2.0  0.2  0:14.64 /usr/bin/pulseaud
  29302 adriano    20   0 27348  4680  3432 R  2.0  0.1  0:00.73 htop
   2854 adriano    -6   0 2327M 16232 12024 S  2.0  0.2  0:08.52 /usr/bin/pulseaud
  27872 adriano    20   0 28.5G  144M  102M S  2.0  1.8  0:03.06 /opt/google/chrom
  15089 adriano    20   0 32.6G  364M  120M S  1.3  4.6  0:03.19 /opt/google/chrom
  14780 adriano    20   0 5229M 1728M 1646M S  1.3 21.9  0:26.23 /usr/lib/virtualb
   7219 adriano    20   0 16.8G  286M  116M S  0.7  3.6  4:20.07 /opt/google/chrom
  29341 adriano    20   0 17.0G 50688 39672 S  0.7  0.6  0:00.19 /opt/google/chrom
   7584 adriano    20   0 17.0G 50688 39672 S  0.7  0.6  0:05.85 /opt/google/chrom
   7259 adriano    20   0 16.8G  148M 95472 S  0.7  1.9  5:51.79 /opt/google/chrom
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

Outros Aspectos

- **Outros aspectos relevantes para Sistemas Distribuídos...**
- Em que estado um processo qualquer passa maior parte do tempo?
  - CPU-Bound
  - IO-Bound
- Quantos processos podem estar em estado Running simultaneamente?
- Processos podem se comunicar entre si?
  - IPC

Outros Aspectos

- **Outros aspectos relevantes para Sistemas Distribuídos...**
- Em que estado um processo qualquer passa maior parte do tempo?
  - CPU-Bound
  - IO-Bound
- Quantos processos podem estar em estado Running simultaneamente?
- Processos podem se comunicar entre si?
  - IPC

Veremos tudo isso em detalhes à frente...
MAS ANTES E MAIS IMPORTANTE...

Concorrência e Paralelismo

Concorrência e **Paralelismo** são sinônimos?

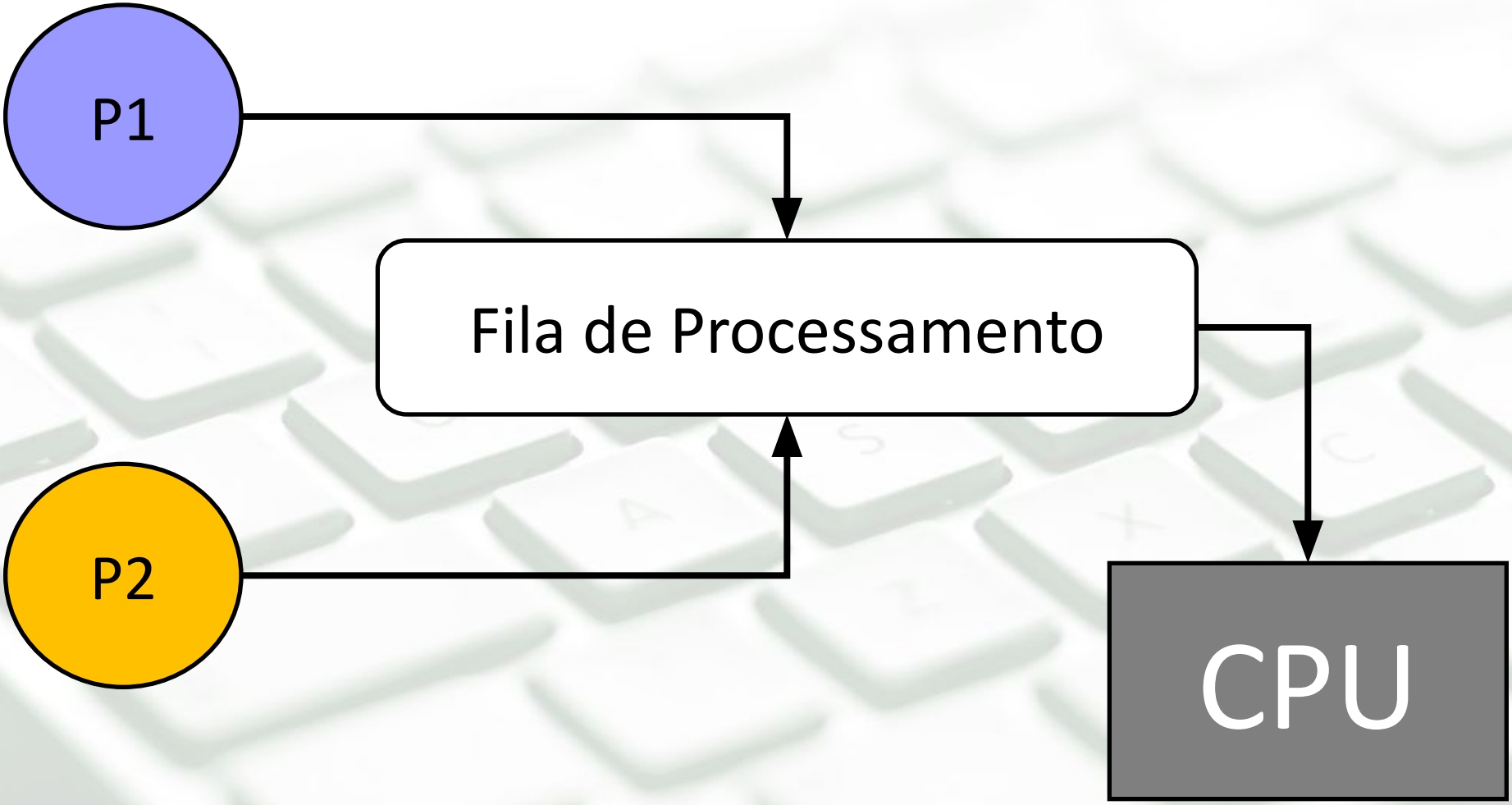
Concorrência

- **Computação multitarefa** é obtida a partir do suporte à **Concorrência**.
- **Concorrência** é o termo que utilizado quando processos **disputam o acesso a recursos compartilhados** (p.ex. CPU).

- Graças a este conceito, você tem a impressão que um computador "*faz várias coisas ao mesmo tempo*", o que não é necessariamente verdade.

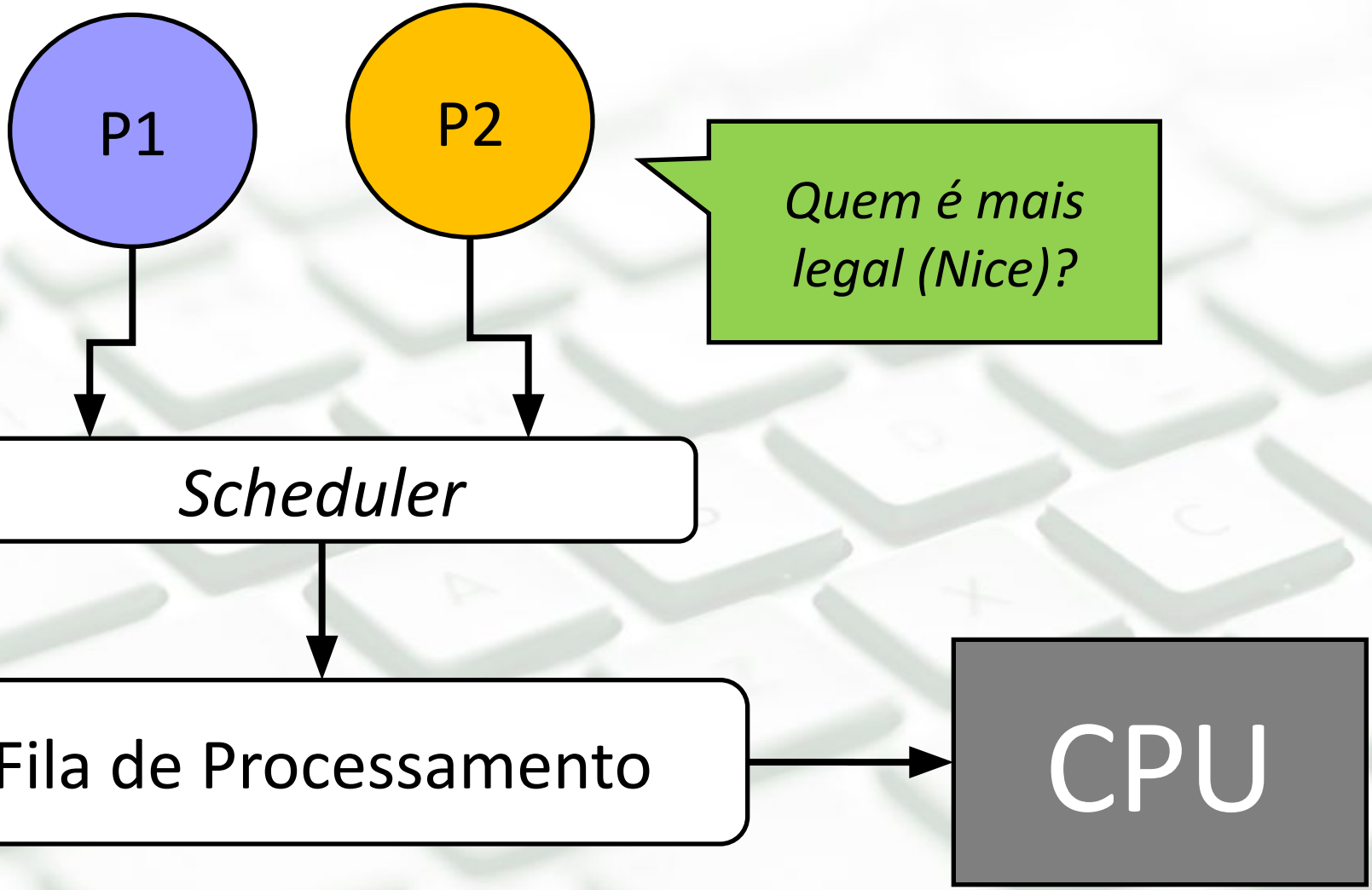
INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Concorrência



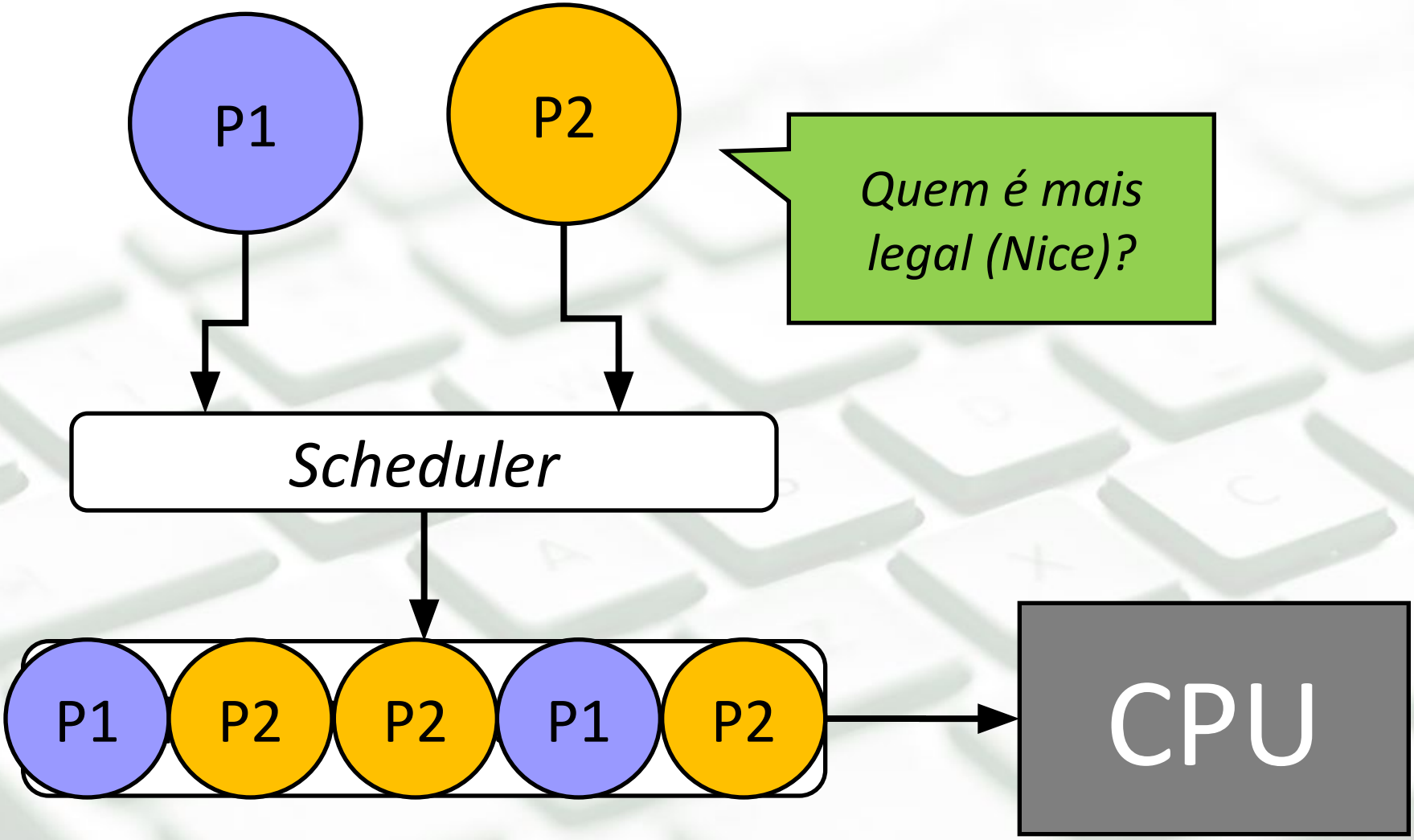
INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Escalonador

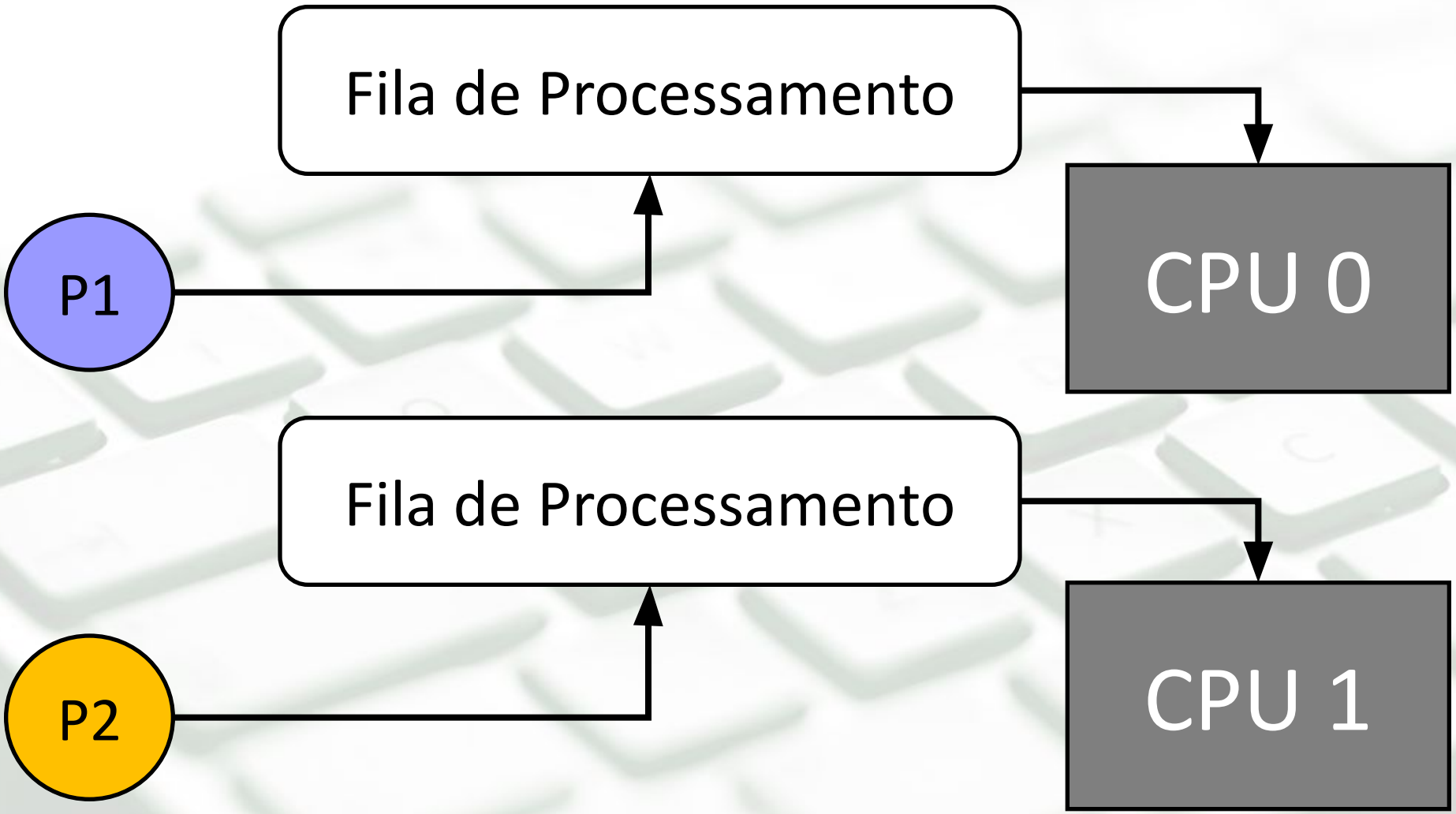


INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Escalonador

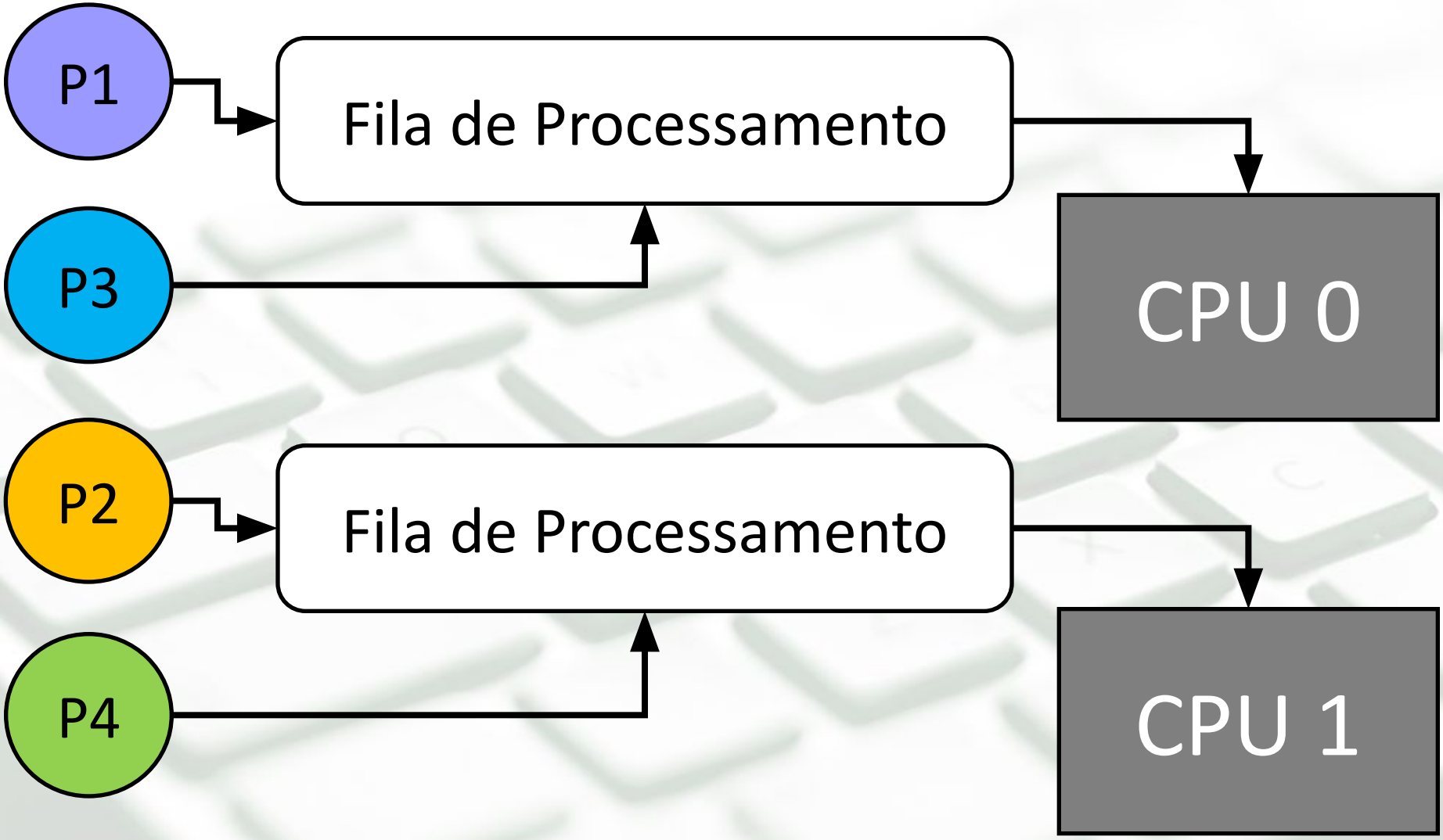


Paralelismo



INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Paralelismo com Concorrência



Filosofando...

- ***Concorrência*** *é sobre* ***lidar*** *com várias coisas ao mesmo tempo...*

- ***Paralelismo*** *é sobre* ***fazer*** *várias coisas ao mesmo tempo...*

Sendo mais formal...

- ***Concorrência*** *é a capacidade de se executar duas ou mais tarefas em* ***um período de tempo****;*

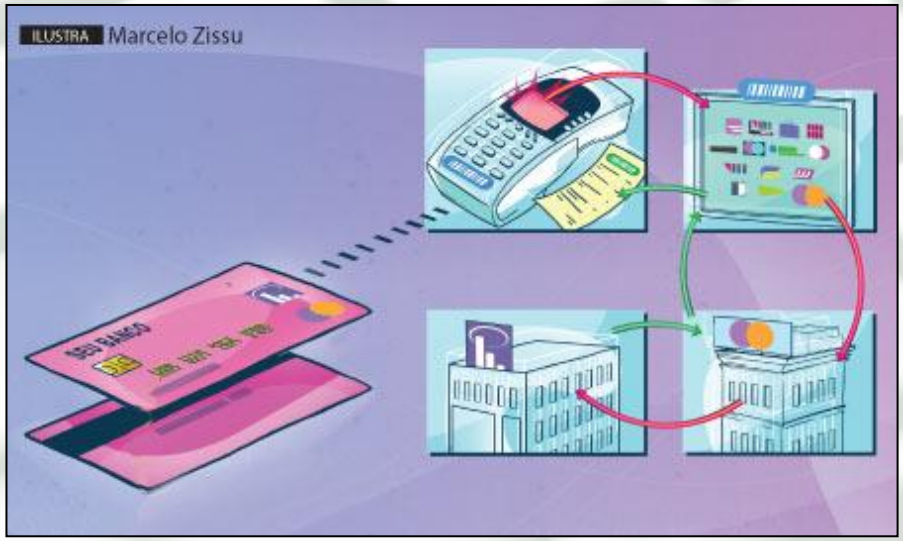
- ***Paralelismo*** *é a capacidade de se executar duas ou mais tarefas de* ***forma simultânea****;*

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Computação Distribuída

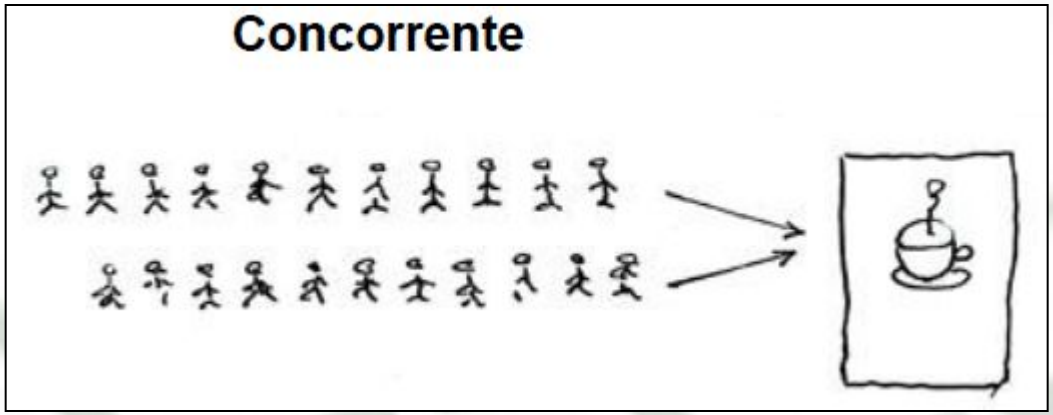
- Um **aglomerado de computadores (ou sistemas)** trabalhando **em conjunto** e que se apresentam ao usuário **como um sistema único**.

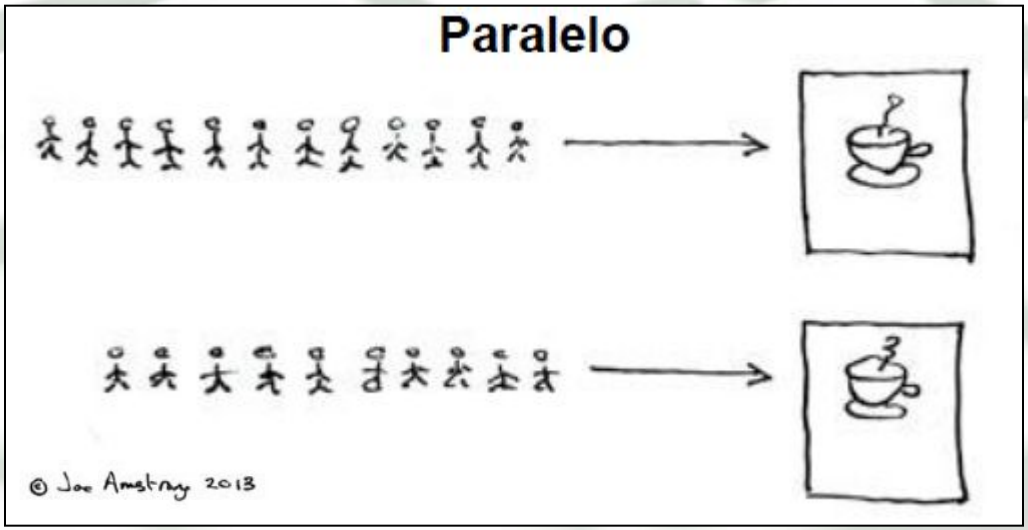
"Concorrência e Paralelismo para além da sua própria máquina".



INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Resumindo...

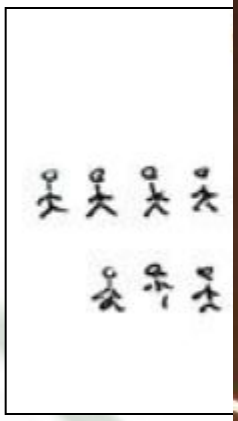


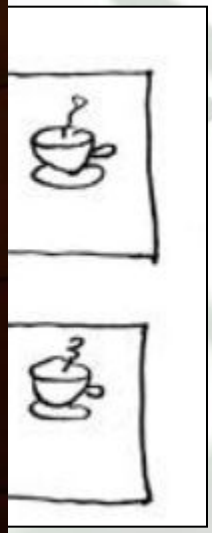


INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Resumindo...

Distribuído





Analisando um Caso…

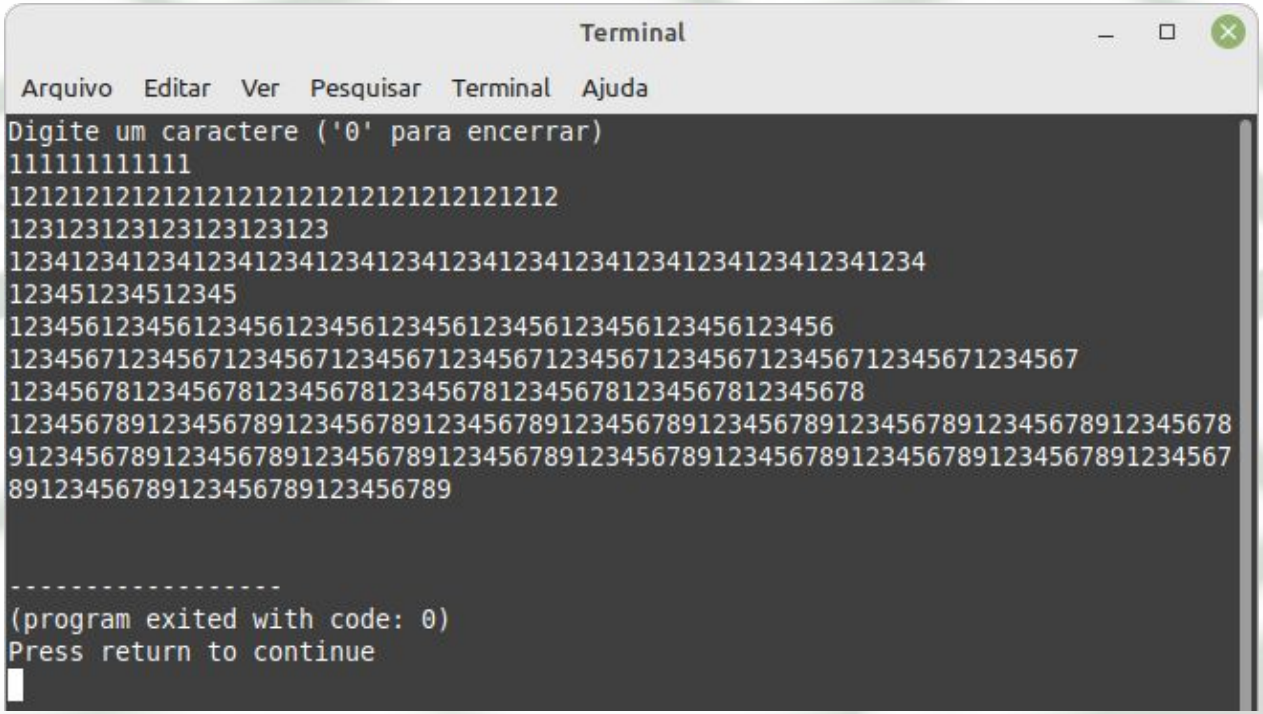
- Imagine o desenvolvimento de um servidor WEB, que deve oferecer concorrência para atender vários clientes simultaneamente…

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Analisando um Caso…

- **Imagine o desenvolvimento de um servidor WEB, que deve oferecer concorrência para atender vários clientes simultaneamente…**

Lembram-se?



Analisando um Caso…

- Imagine o desenvolvimento de um servidor WEB, que deve oferecer concorrência para atender vários clientes simultaneamente…

Seria interessante criar um novo processo para atender cada cliente que requisita a página?

Analisando um Caso…

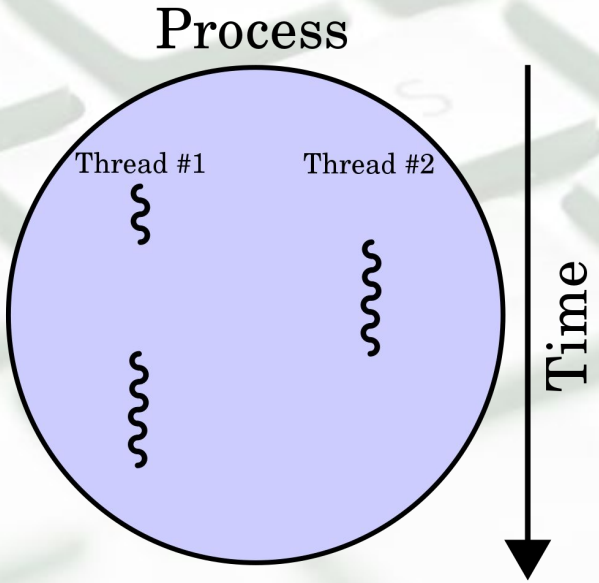
- **Imagine o desenvolvimento de um servidor WEB, que deve oferecer concorrência para atender vários clientes simultaneamente…**

Seria interessante criar um novo processo para atender cada cliente que requisita a página?

- Uso de memória: PCBs distintos e isolados.
- Custo para gerência dos processos:
  - Trocas de contexto.
- No fim das contas, **a mesma página que será enviada a todos os clientes…**

Threads

- **Solução: Separar o conceito de Processo do conceito de Linha de Execução.**
- Imagine ***threads*** como **linhas de execução distintas e concorrentes** dentro de **um mesmo processo**.

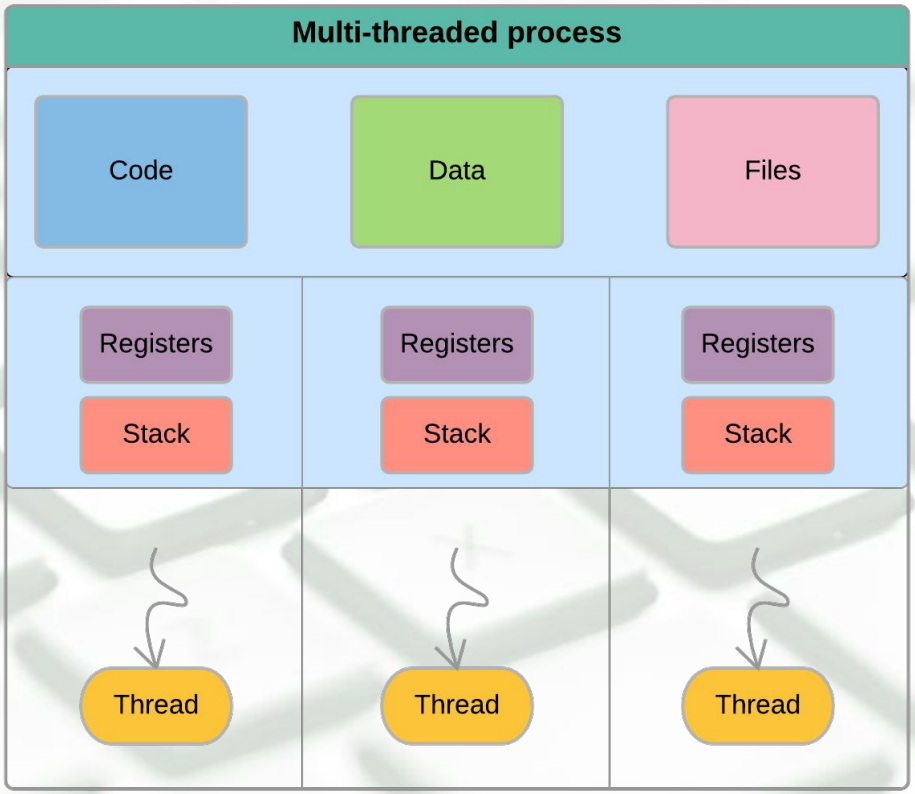


Threads

- Assim como em um processo, uma **thread** executa uma porção bem definida de código, e também **independente** de outras threads.
- Entretanto, no caso de *threads,* o S.O. não se preocupa com alto grau de transparência na concorrência entre essas threads.
- As **threads** passam a ser a unidade de escalonamento no contexto de um processo.
- Cada CPU (core) executa um processo por vez, e dentro deste contexto, **uma thread por vez**.

Threads

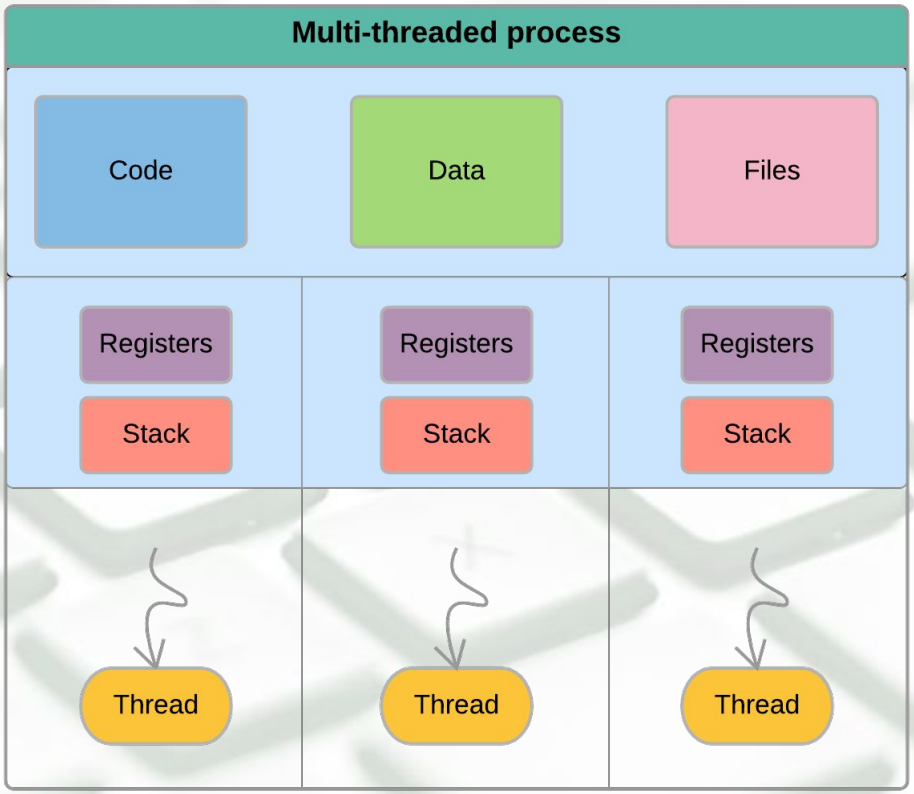
- Cada **processo** contém, no mínimo, uma thread (*Main Thread*).
- Processos podem porém, implementar um conjunto de outras **threads** (controles de fluxos distintos) que serão executadas **concorrentemente** no mesmo escopo do processo.



INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Threads

- Perceba que as *threads* **compartilham código, arquivos e dados (variáveis)** do processo como um todo.



- É algo POSITIVO:

  Economia e Praticidade

- E NEGATIVO:

  Segurança e Sincronização

NEGATIVO???

- Imagine que um sistema bancário utiliza *threads* para atender as requisições de movimentação financeira de uma conta-corrente.

- Em que aspecto isso se tornaria um grande problema?

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Acompanhe...

Saldo Atual da Conta: R$ 1.000	
Thread A	**Thread B**
Consulta Saldo da Conta	
	Consulta Saldo da Conta
Realiza Saque de R$ 1.000,00	
	Paga conta de R$ 500,00
Saldo Atual da Conta: R$???	

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Acompanhe...

Saldo Atual da Conta: R$ 1.000	
Thread A	**Thread B**
Consulta Saldo da Conta	
	Consulta Saldo da Conta
Realiza Saque de R$ 1.000,00	
	Paga conta de R$ 500,00
Saldo Atual da Conta: R$???	

Vocês já devem ter estudado o conceito de **ACID** em outras disciplinas...

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Acompanhe...

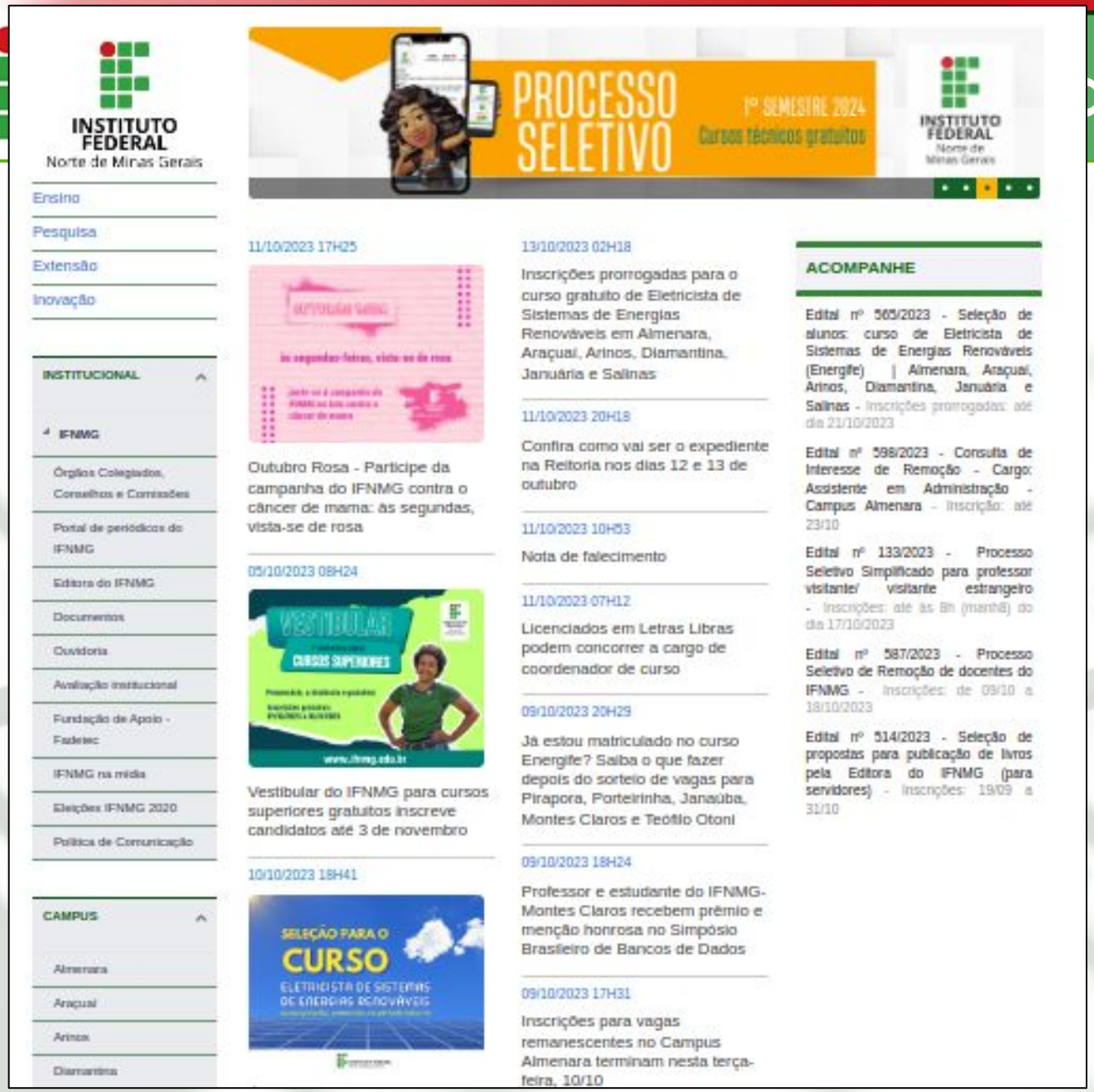
Saldo Atual da Conta: R$ 1.000	
Thread A	**Thread B**
Consulta Saldo da Conta	
	Consulta Saldo da Conta
Realiza Saque de R$ 1.000,00	
	Paga conta de R$ 500,00
Saldo Atual da Conta: R$???	

Vocês já devem ter estudado o conceito de **ACID** em outras disciplinas...
ATOMICIDADE, CONSISTÊNCIA, ISOLAMENTO E DURABILIDADE

Threads

- Contudo, as ***threads*** são muito importantes para a grande maioria das aplicações…

- Vantagens:
  - Facilidade para desenvolvimento de concorrência.
  - Sobreposição de operações de I/O e computação.
  - Melhor aproveitamento da CPU.
  - …



Imagine se para toda página WEB, apenas uma imagem pudesse ser carregada por vez.

INSTITUTO FEDERAL
Norte de Minas Gerais

- Ensino
- Pesquisa
- Extensão
- Inovação

INSTITUCIONAL

- IFNMG
- Órgãos Colegiados, Conselhos e Comissões
- Portal de periódicos do IFNMG
- Editora do IFNMG
- Documentos
- Ouvidoria
- Avaliação institucional
- Fundação de Apoio - Fadetec
- IFNMG na mídia
- Eleições IFNMG 2020
- Política de Comunicação

CAMPUS

- Almenara
- Araçuaí
- Arinos
- Diamantina

11/10/2023 17H25

Outubro Rosa - Participe da campanha do IFNMG contra o câncer de mama: às segundas, vista-se de rosa

05/10/2023 08H24

Vestibular do IFNMG para cursos superiores gratuitos inscreve candidatos até 3 de novembro

10/10/2023 18H41

13/10/2023 02H18

Inscrições prorrogadas para o curso gratuito de Eletricista de Sistemas de Energias Renováveis em Almenara, Araçuaí, Arinos, Diamantina, Januária e Salinas

11/10/2023 20H18

Confira como vai ser o expediente na Reitoria nos dias 12 e 13 de outubro

11/10/2023 10H53

Nota de falecimento

11/10/2023 07H12

Licenciados em Letras Libras podem concorrer a cargo de coordenador de curso

09/10/2023 20H29

Já estou matriculado no curso Energife? Saiba o que fazer depois do sorteio de vagas para Pirapora, Porteirinha, Janaúba, Montes Claros e Teófilo Otoni

09/10/2023 18H24

Professor e estudante do IFNMG-Montes Claros recebem prêmio e menção honrosa no Simpósio Brasileiro de Bancos de Dados

09/10/2023 17H31

Inscrições para vagas remanescentes no Campus Almenara terminam nesta terça-feira, 10/10

ACOMPANHE

Edital nº 565/2023 - Seleção de alunos: curso de Eletricista de Sistemas de Energias Renováveis (Energife) | Almenara, Araçuaí, Arinos, Diamantina, Januária e Salinas - Inscrições prorrogadas: até dia 21/10/2023

Edital nº 598/2023 - Consulta de Interesse de Remoção - Cargo: Assistente em Administração - Campus Almenara - Inscrição: até 23/10

Edital nº 133/2023 - Processo Seletivo Simplificado para professor visitante/ visitante estrangeiro - Inscrições: até às 8h (manhã) do dia 17/10/2023

Edital nº 587/2023 - Processo Seletivo de Remoção de docentes do IFNMG - Inscrições: de 09/10 a 18/10/2023

Edital nº 514/2023 - Seleção de propostas para publicação de livros pela Editora do IFNMG (para servidores) - Inscrições: 19/09 a 31/10

Imagine se para

Com *multi threading*, podemos fazer com que cada arquivo da página seja **baixado e renderizado de forma concorrente**.

Implementação de Threads

- Existem duas formas de implementação de **Threads**
  - Kernel Level Threads
  - User Level Threads

Kernel Level Threads

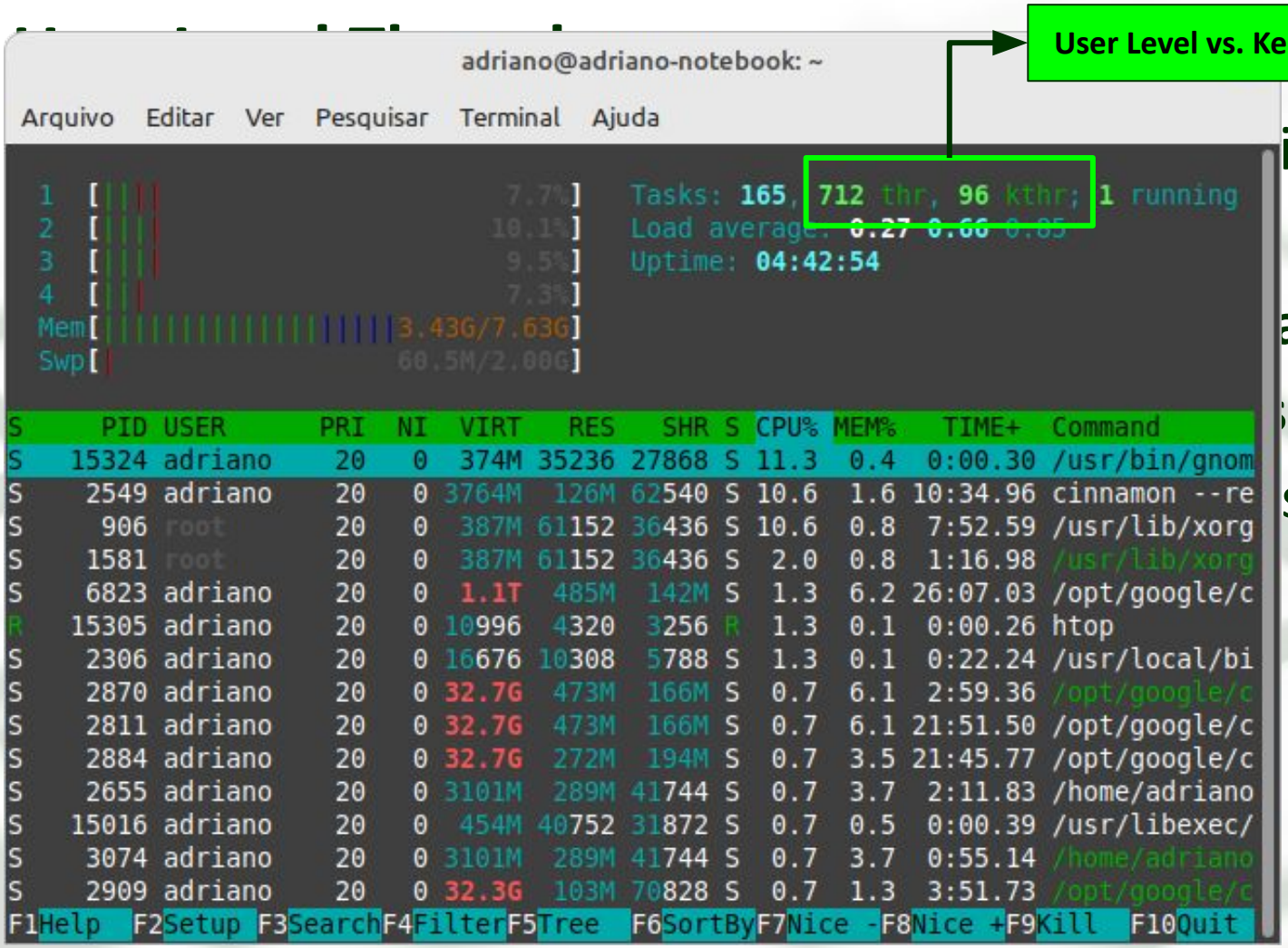
- **Kernel Level Threads**
  - Também chamado de *Lightweight Process* (LWP)
- Criadas através de *System Calls* do próprio S.O.
- Neste modelo, **as threads são reconhecidas pelo Sistema Operacional**.
- O S.O. faz escalonamento das threads (e não dos processos vinculados)
- Consegue resolver problema de justiça entre processos:
  - P.Ex.: P1 tem 2 threads e P2 tem 10 threads. A CPU possui 2 cores. Qual forma mais justa de escalonar?

User Level Threads

- **User Level Threads**
  - Implementado através de Bibliotecas específicas.
- Threads são invisíveis ao Sistema Operacional.
- **Criação de threads, troca de contexto e sincronização é feito por chamadas de funções do próprio processo.**
- Por não depender do S.O., são mais leves e rápidas.
  - Gerenciamento chega a ser 100x mais rápido do que Kernel Level Threads.
- Desvantagem: Escalonamento das threads. (O S.O não possui controle sobre elas)

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

User Level Threads



- icas.
- ação é
- so.
- s.
- Kernel
- não

Python Threading

- Módulo **Threading** no **Python.**

```
import threading

def olaMundo(idThread):
    print(f'Olá Mundo! Sou a thread {idThread}')

threads = []

for i in range(5):
    t = threading.Thread(target=olaMundo, args=(i+1,))
    t.start()
    threads.append(t)
```

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Python Threading

- Implemente cada modificação a seguir, em etapas (em cada modificação, execute várias vezes a solução e observe o comportamento):

1. Faça com que cada thread imprima a mesma mensagem 10 vezes.
2. Faça com que o tempo entre uma impressão e outra seja de 2 segundos (sleep)
3. Use o argumento `(target=olaMundo, args=(i+1,), daemon=True)`
4. Retire o passo anterior, e faça com que o programa encerre a execução com a frase "**ATÉ MAIS**"

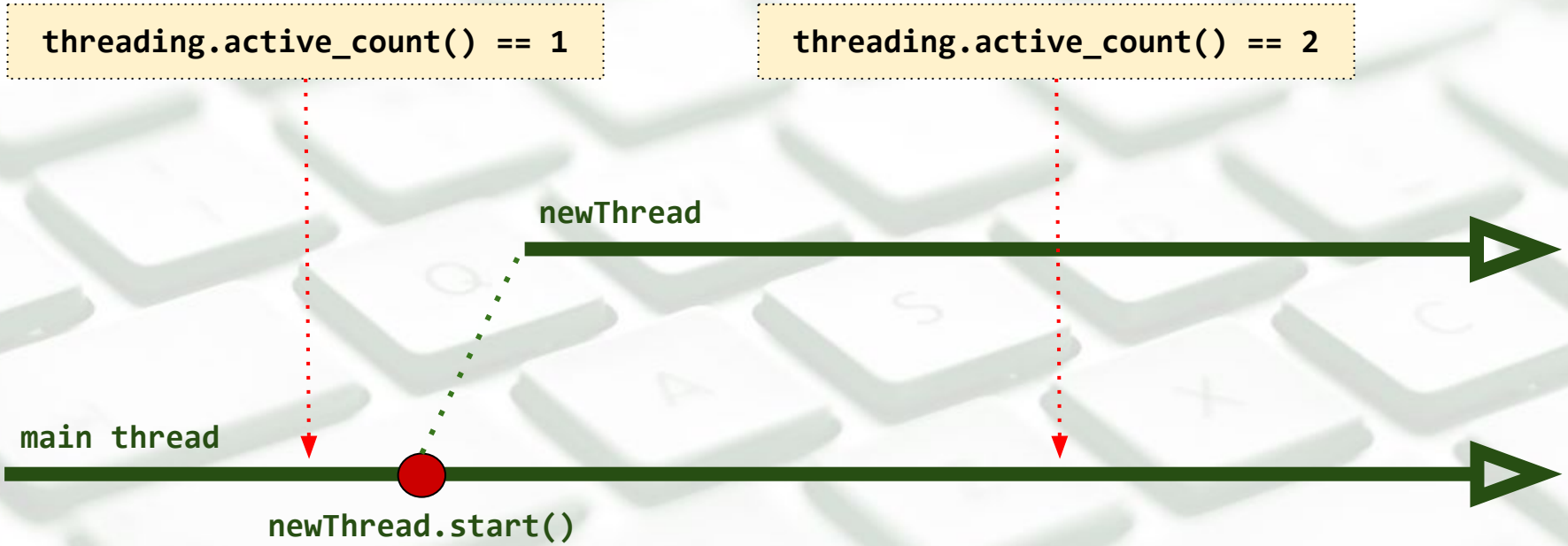
Python Threading

- Comportamento *multithreading*.

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Python Threading

- Comportamento *multithreading*.



INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Python Threading

- Comportamento *multithreading*.

O que acontece com o programa se a **mainThread** finaliza a sua execução antes da **newThread**?

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Python Threading

- Comportamento *multithreading*.

O que acontece com o programa se a **mainThread** finaliza a sua execução antes da **newThread**?

Em modo padrão, o programa só encerra quando **todas as threads finalizam** sua execução!

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Python Threading

- Comportamento *multithreading*.

O que acontece com o programa se a **mainThread** finaliza a sua execução antes da **newThread**?

É um problema quando precisamos de uma thread em loop infinito, p.ex.: aceitar requisições de conexão…

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Python Threading

- Comportamento *multithreading*.

O que acontece com o programa se a **mainThread** finaliza a sua execução antes da **newThread**?

Threads com parâmetro daemon=True são encerradas assim que a main thread é finalizada.

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Python Threading

- Comportamento *multithreading*.

O que acontece com o programa se a **mainThread** finaliza a sua execução antes da **newThread**?

MAS... Como garantir agora que a newThread fez seu trabalho até o fim???

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Python Threading

- Comportamento *multithreading*.

O que acontece com o programa se a **mainThread** finaliza a sua execução antes da **newThread**?

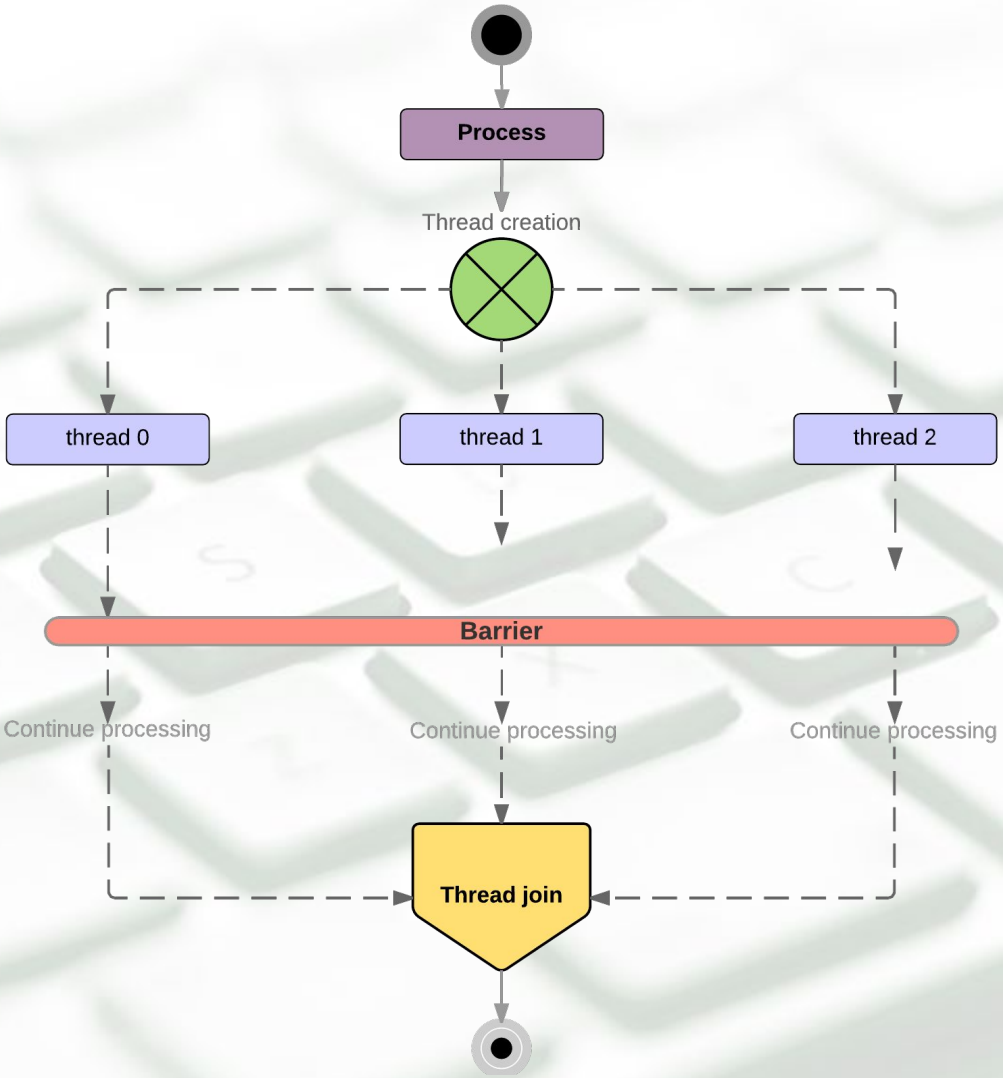
É necessário técnicas de sincronização!

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Método Join( )

Join é uma técnica de sincronização de threads que estabelece um **ponto de barreira.**

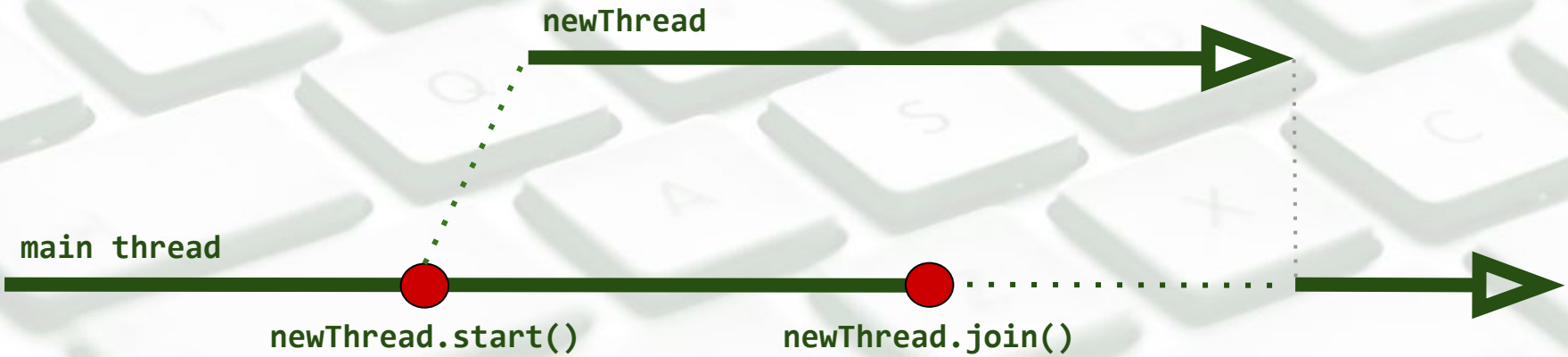
Se uma **thread t1** executa a instrução **t2.join()**, **t1** será bloqueado até a finalização de **t2**.



Python Threading

- Comportamento *multithreading*.

Sincronização com Join( )



INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Python Threading

- Comportamento *multithreading*.

Sincronização com Event( )

Laboratório 02-01

- **Laboratórios Práticos**
  - Faça uma aplicação que simule o *download* de uma lista com 10 arquivos, em **modo serial** e depois em **modo concorrente**.
  - Cada *download* demora cerca de 5 segundos para ser finalizado.
  - Utilize o módulo **threading** para fazer a concorrência.
  - Calcule o tempo de execução em cada versão, e compare.

Laboratório 02-02

- **Corrida Maluca**
  - Desenvolva um game que simula a corrida entre 5 threads.
  - Cada thread percorre a distância de 1 casa com tempo aleatório uma das outras - use random.random( ).
  - Atualize a tela constantemente para visualizar a corrida e acompanhar a evolução dos competidores.
  - Ganha a thread que chegar primeiro à casa 80.
  - Ao final, seu programa deve informar a thread vencedora, ou se houve um empate.

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Laboratório 02-02

```
**********************************************************************************
**********************************************************************************
**********************************************************************************
**********************************************************************************
**********************************************************************
THREAD 4 VENCEDOR

Process finished with exit code 0
```

```
**********************************************************************************
**********************************************************************************
**********************************************************************************
**********************************************************************************
**********************************************************************
HOUVE EMPATE

Process finished with exit code 0
```

Laboratório 02-03

- Imagine uma pizza gigante, com 128 pedaços disponíveis.
- Imagine que 3 threads que irão consumir toda essa pizza, começando do primeiro pedaço, até o último.
- Cada thread demora aproximadamente 3 segundos para comer cada pedaço.
- Faça a implementação de uma aplicação em Python que simula esse cenário…

INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Laboratório 02-03

- Imagine
  disponí
- Imagine ssa
  pizza, c mo.
- Cada th dos
  para co
- Faça a i hon
  que sim

```
adriano@adriano: ~/Dropbox/0. IFNMG/AULAS/_SLIDES DE AULA/SUPERIOR - SISTE...
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
Thread 1 comendo o pedaço: 1
Thread 2 comendo o pedaço: 2
Thread 3 comendo o pedaço: 3
Thread 2 comendo o pedaço: 4
Thread 3 comendo o pedaço: 5
Thread 1 comendo o pedaço: 6
Thread 2 comendo o pedaço: 7
Thread 3 comendo o pedaço: 8
Thread 1 comendo o pedaço: 9
Thread 2 comendo o pedaço: 10
Thread 3 comendo o pedaço: 11
Thread 1 comendo o pedaço: 12
Thread 3 comendo o pedaço: 13
Thread 2 comendo o pedaço: 14
Thread 1 comendo o pedaço: 15
Thread 3 comendo o pedaço: 16
Thread 2 comendo o pedaço: 17
Thread 1 comendo o pedaço: 18
Thread 2 comendo o pedaço: 19
Thread 1 comendo o pedaço: 20
Thread 3 comendo o pedaço: 21
Thread 3 comendo o pedaço: 22
Thread 1 comendo o pedaço: 23
Thread 2 comendo o pedaço: 24
Thread 2 comendo o pedaço: 25
Thread 3 comendo o pedaço: 26
Thread 1 comendo o pedaço: 27
Thread 2 comendo o pedaço: 28
Thread 1 comendo o pedaço: 29
```