

Computação Concorrente (DCC/UFRJ)

Aula 3: Visão geral dos sistemas de computação e histórico da concorrência

Prof. Silvana Rossetto

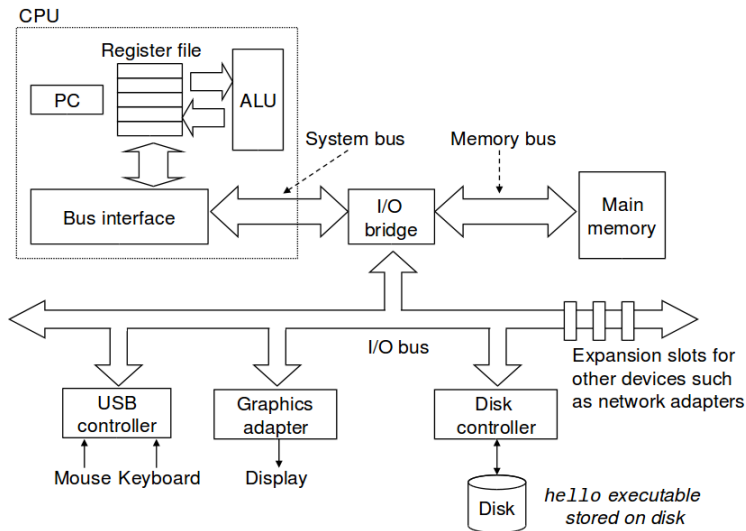
20 de março de 2012

- 1 Visão geral dos sistemas de computação
- 2 O papel do sistema operacional
- 3 Histórico da computação concorrente
 - Concorrência dentro do SO
 - Concorrência entre aplicações
 - Concorrência dentro das aplicações
 - Construção de programas concorrentes

Sistemas de computação

Consistem de **hardware** e **software** que funcionam juntos para executar aplicações do usuário

Organização do hardware de um sistema



Organização do hardware de um sistema

Dispositivos de E/S

Conexão do sistema com o mundo externo (monitor, disco, teclado, interface de rede, etc.)

Memória principal

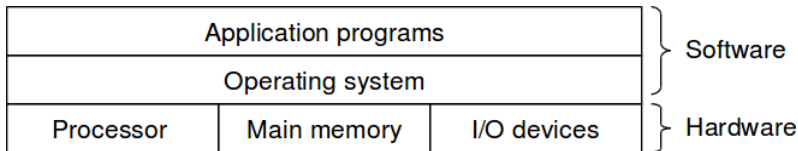
Dispositivo de armazenamento temporário que **armazena programas e os dados** usados por eles enquanto o processador está executando o programa (**logicamente a memória é organizada como um vetor de bytes, cada um com seu próprio endereço**)

Processador

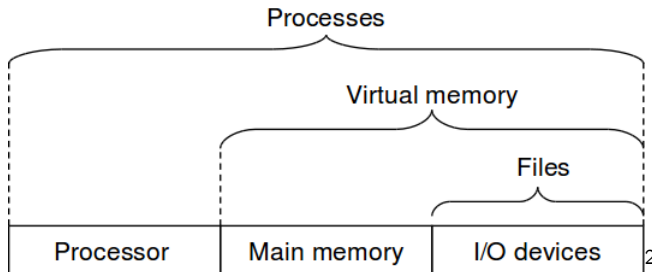
A CPU (Unidade Central de Processamento) interpreta e executa as instruções armazenadas na memória principal

Funcionalidade dos sistemas operacionais

- O SO é uma camada de software interposta entre o hardware e as aplicações, e visa **proteger e simplificar o acesso ao hardware**
- Os programas do usuário usam os serviços oferecidos pelo Sistema Operacional (SO) para acessar o hardware da máquina



Abstrações providas pelo sistema operacional



Vários processos podem **executar concorrentemente** no mesmo sistema de computação e **cada processo “parece” ter uso exclusivo do hardware**

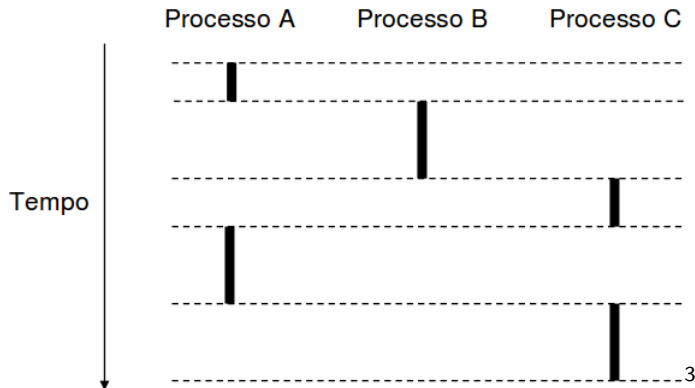
²Fonte: <http://csapp.cs.cmu.edu>

Execução concorrente

Executar **concorrentemente** significa que se observarmos o comportamento do sistema ao longo do tempo, veremos que a execução das instruções de um processo alternam com a execução das instruções de outros processos

Um único processador **aparenta executar vários processos concorrentemente**, executando um pedaço de cada até terminarem

Atenção de execução entre processos



³Fonte: <http://csapp.cs.cmu.edu>

Troca de contexto

- Quando o SO transfere o controle do processo atual para algum novo processo, ele executa uma **troca de contexto**:
 - 1 salva o contexto do processo atual
 - 2 restaura o contexto do novo processo
 - 3 passa o controle para o novo processo
- O novo processo retoma a sua execução exatamente do ponto onde ele parou anteriormente

Concorrência aparece nos vários níveis dos sistemas de computação

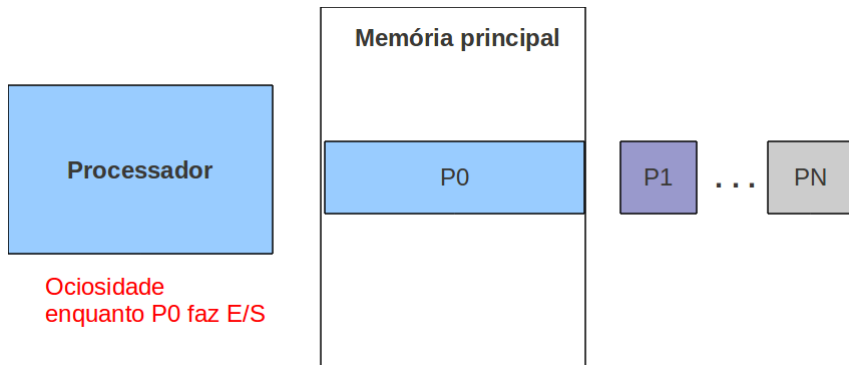
- 1 Circuitos digitais que operam em paralelo
- 2 Execução de instruções usando o mecanismo de pipelining
- 3 Construções semanticamente visíveis ao programador (processos, threads, co-rotinas, etc.)

Concorrência dentro do código do Sistema Operacional

Antes da concorrência...

- **Máquinas monousuário:** carga manual de um programa após o outro
- **Processamento em lote:** **jobs** (programas + dados) enfileirados e controlados por um **monitor**

Processamento em lote

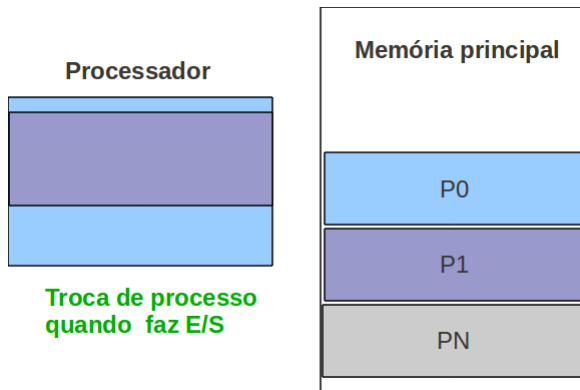


Concorrência dentro do código do Sistema Operacional

Introdução da concorrência nos Sistemas Operacionais

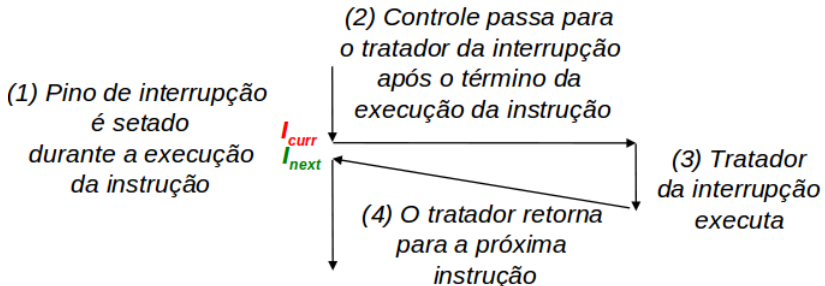
E/S dirigido a interrupção (elimina a necessidade de espera ocupada por E/S) + **Multiprogramação** (permite mais de um programa na memória principal ao mesmo tempo)

E/S dirigido a interrupção e multiprogramação



Concorrência dentro do código do Sistema Operacional

Como uma interrupção pode acontecer a qualquer momento, incluindo quando o controle já está com o Sistema Operacional, os **tratadores de interrupção** e o **código principal do Sistema Operacional** são **linhas de controle concorrentes**



Problema da concorrência dentro do código do SO

main program:

read head

A->next := head

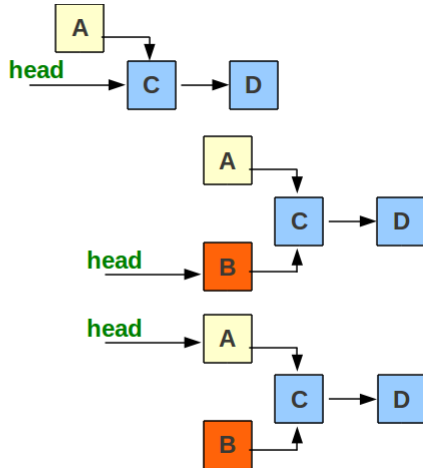
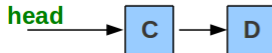
Handler:

read head

B->next := head

head := B

head := A



4

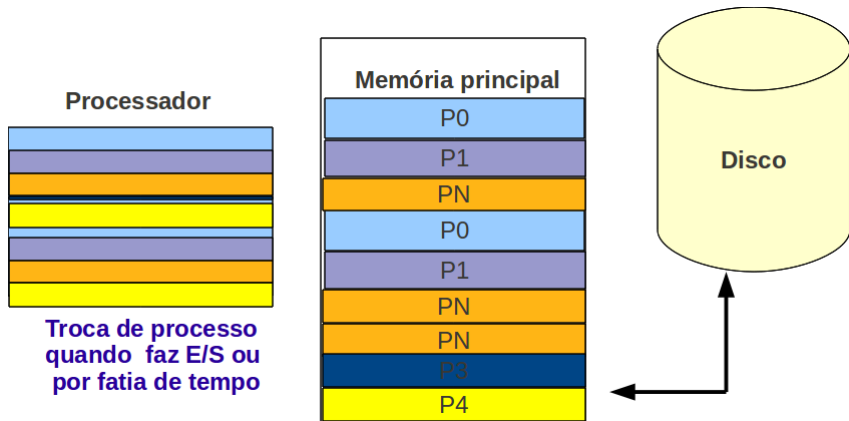
⁴ Fonte: "Programming Language Pragmatics", M.L.Scott, 2006

Concorrência entre aplicações

Sistemas de tempo compartilhado

- Com o incremento do espaço de memória, tornou-se possível construir **sistemas com um número arbitrário de programas carregados simultaneamente na memória**
- Ao invés de submeter *jobs* em modo *offline*, o usuário pode interagir com o computador diretamente
- Para prover **respostas interativas**, implementou-se a idéia de **preempção**: **fatias de tempo para uso da CPU**
- Esses sistemas ficaram conhecidos como **sistemas de tempo compartilhado** (ou *timesharing*)

Sistemas de tempo compartilhado



Concorrência dentro das aplicações

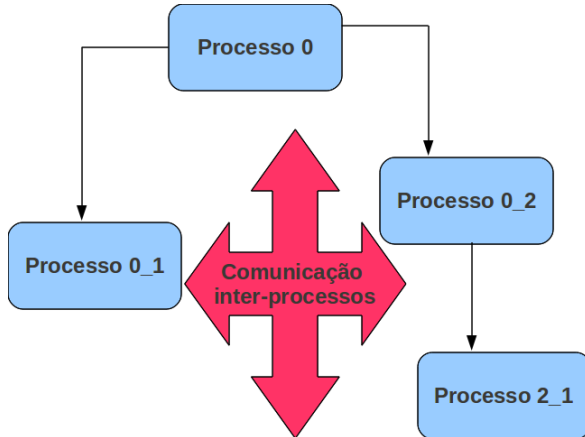
Aplicações concorrentes

Quando acrescentados de mecanismos que permitiram **compartilhamento de dados** e outras formas de comunicação entre os programas em execução, a **concorrência foi introduzida nas aplicações no nível do usuário**

Caminhos para construção de programas concorrentes

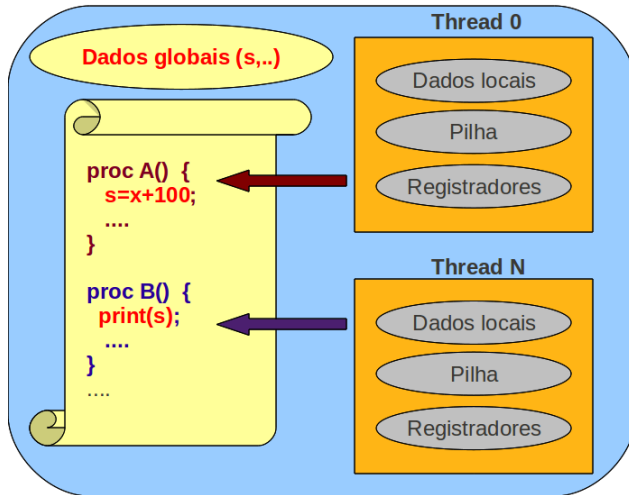
- 1 Aplicação com **vários processos** (**maior custo com troca de contexto e comunicação entre processos**)
- 2 Aplicação com **várias threads** (multitarefa preemptiva)
- 3 Aplicação com **várias co-rotinas** (multitarefa cooperativa) (**veremos depois!**)
- 4 Aplicação com **multiplexação de E/S** (transições de estado ativadas por interrupções de E/S) (**veremos depois!**)

Aplicações com vários processos



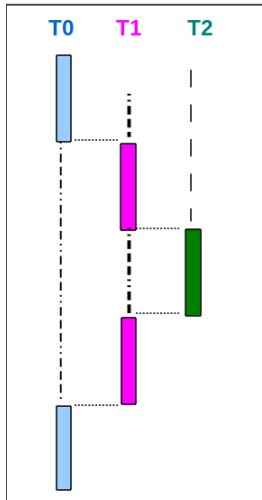
Aplicações com várias threads

Processo 0

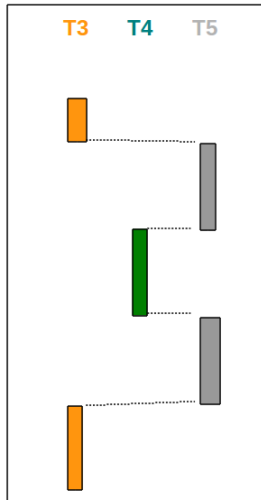


Alternância **implícita** entre threads

Processador 1



Processador 2



Concorrência dentro do código de uma aplicação

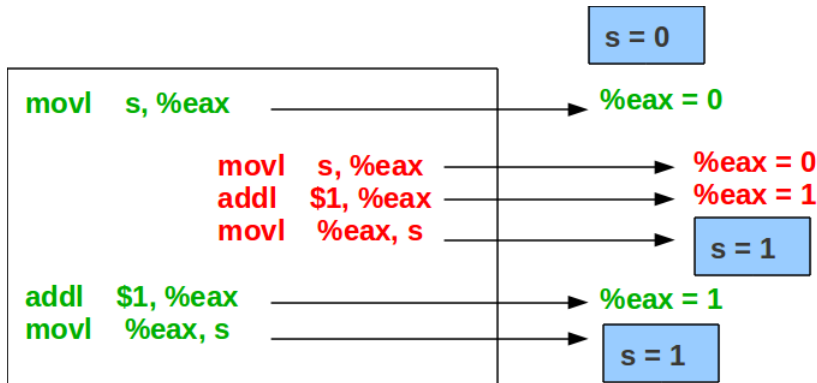
```
int s;  
  
void soma() {  
    s++;  
}
```

```
.comm s,4,4  
soma: (...  
    movl  s, %eax  
  
    addl  $1, %eax  
  
    movl  %eax, s  
    (...)
```

Interrupção do tempo

Interrupção do tempo

Condição de corrida dentro do código de uma aplicação



Referências bibliográficas

- *Computer Systems - A Programmer's Perspective* (**Cap.1**)
- *Programming Language Pragmatics*, M.L.Scott, Morgan-Kaufmann, ed. 2, 2006
- *Operating Systems – Internals and Design Principles*, W. Stallings, Pearson, ed. 6, 2009
- *Modern Multithreading*, Carver e Tai, Wiley, 2006