

Computação Concorrente (DCC/UFRJ)

Aula 1: Apresentação e motivação da disciplina

Prof. Silvana Rossetto

6 de março de 2011

- 1 Programação concorrente
 - Motivação para a programação concorrente
 - Desafios para a computação concorrente

- 2 Visão geral da disciplina
 - Objetivos e ementa
 - Metodologia
 - Material bibliográfico

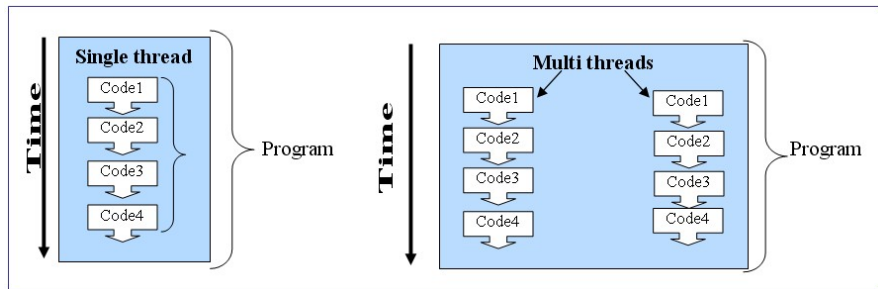
Programação concorrente

- Atividade de construir programas de computador que incluem **linhas de controle distintas**, as quais podem executar simultaneamente
- As diferentes linhas de controle **cooperam para a execução da tarefa principal**

Exemplo de concorrência



Exemplo de implementação concorrente



²Fonte: <http://programmingforfuture.com>

A tactical diagram of a football team in a 4-3-3 formation on a pitch. The goalkeeper is Valdes (1) in the center. Defenders are Pique (3) on the left, Abidal (22) on the right, and Alves (2) at the bottom. Midfielders are Puyol (5) on the left, Iniesta (8) in the center, and Xavi (6) at the bottom. Forwards are Villa (7) on the left, Messi (10) in the center, and Pedro (17) on the right. Arrows indicate player movements and passing lanes.

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡

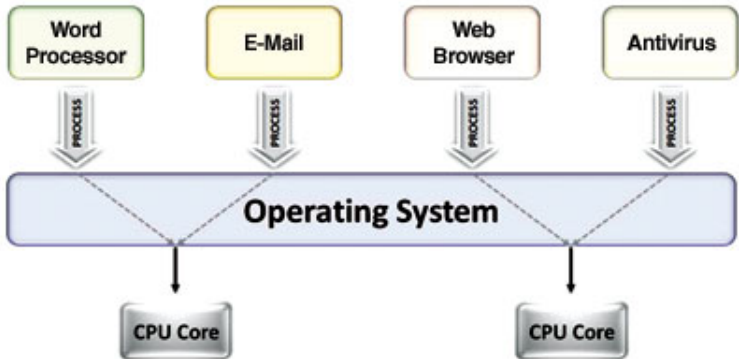
Motivação para a programação concorrente

Para que serve a **programação concorrente** e onde ela já é usada?

Motivações para a computação concorrente

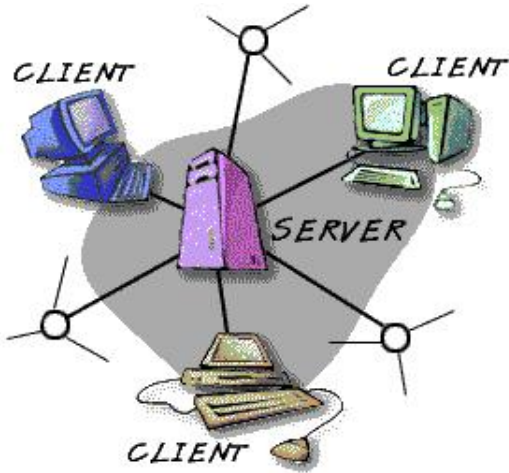
- **Necessidade de capturar a estrutura lógica de um problema:** ex., aplicações Web, aplicações gráficas
- **Necessidade de lidar com dispositivos independentes:** ex., sistemas operacionais, sistemas de controle de automóveis, roteamento de msgs na Internet
- **Necessidade de aumentar o desempenho das aplicações:** ex., computação científica

Explorar o paralelismo do hardware



⁴ Fonte: <http://www.evaluationengineering.com>

Desenvolver aplicações distribuídas



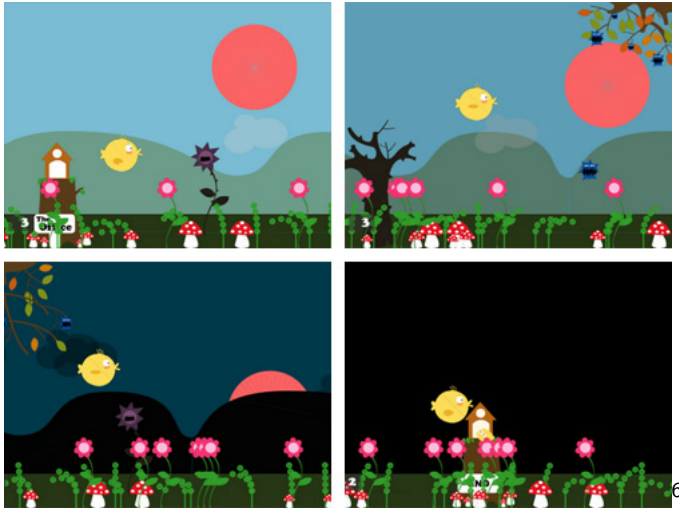
5

⁵Fonte: <http://buzzparas.com>

Permitir sobreposição de processamento de E/S



Suprir requisitos das próprias aplicações



⁶Fonte: <http://workshop.evolutionzone.com>

Desafios para a computação concorrente

- **Visão ideal da programação concorrente:** partes independentes do programa podem executar em paralelo, e sempre que uma parte depende dos resultados produzidos por outra, a comunicação e a sincronização ocorrem do forma implícita.
- Nas linguagens de programação imperativas é necessário escrever **programas concorrentes imperativos**: a **concorrência**, a **comunicação** e a **sincronização** são explícitas

Desafios para a computação concorrente

A programação concorrente é desafiadora porque os sistemas de computação modernos são inerentemente **assíncronos**:

- as atividades podem ser retardadas sem qualquer previsão devido a eventos de: interrupção, preempção, atualização da memória cache, atraso de comunicação, e outros.

Ângulos de visão

- 1 **computabilidade**, conjunto de princípios que abordam o que pode ser computado em um ambiente concorrente e assíncrono
- 2 **desempenho**, avaliação prática do ganho em desempenho das soluções implementadas

Esforço de paralelização

- Alguns problemas são facilmente paralelizáveis (ex., aplicação gráfica)
- Em geral, para um dado problema e uma máquina com 10 processadores, mesmo que paralelizando 90% da solução, o ganho de tempo será de apenas 5-vezes e não de 10 como seria esperado
- **O esforço principal da programação concorrente é conseguir paralelizar o máximo possível esses 10% restantes**

Foco do estudo em concorrência

- O foco do **curso de Computação Concorrente** é o estudo das ferramentas e técnicas para codificar as partes do código que requerem **comunicação e sincronização** entre as linhas de execução distintas
- Os ganhos alcançados com essa paralelização podem trazer um grande impacto no desempenho final da aplicação

Objetivos e página da disciplina

Objetivos

- 1 Apresentar a motivação e estudar as **regras, ferramentas e estratégias** para a construção de aplicações concorrentes
- 2 Discutir e mostrar **exemplos de modelagem e implementação** de problemas computacionais que são concorrentes por natureza
- 3 Apresentar **bibliotecas e mecanismos** oferecidos pelas linguagens de programação para o desenvolvimento de aplicações concorrentes

Página da disciplina e horário de atendimento

Página da disciplina

[www.dcc.ufrj.br/ silvana/compconc](http://www.dcc.ufrj.br/silvana/compconc)

Lista de discussão

groups.google.com/group/mab117

Horário de atendimento

Terça-feira, de 15:00h às 16:00h (sala E2010)

Ementa da disciplina

- Motivação e histórico da programação concorrente
- Ferramentas e construções para a programação concorrente
- Programação com variáveis compartilhadas (seção crítica e exclusão mútua)
- Métodos de trava (locks, semáforos, monitores).
- Programação com troca de mensagens (primitivas de troca de mensagens send/receive)
- Introdução a objetos distribuídos e invocação remota de métodos.
- Teste e depuração de programas concorrentes
- Modelagem e implementação de programas concorrentes

Metodologia de trabalho

Método de ensino e atividades discentes

- Explanação dos tópicos de estudo e aulas práticas no laboratório (**LAB-1, quinta-feira**)
- As atividades discentes incluirão:
 - Participação nas aulas e estudo de material suplementar
 - Resolução de exercícios propostos (teóricos e práticos)

Instrumentos de acompanhamento e avaliação

- P_1 (24 de abril), P_2 (26 de junho) e P_s (10 de julho)
- Dois trabalhos práticos de implementação ao longo do curso: T_1 e T_2
- Média final:
$$M_f = (((P_1 + P_2)/2) * 0.7) + (((T_1 + T_2)/2) * 0.3)$$
- Se M_f **igual ou superior** a 7.0 dispensado da P_s e $N_f = M_f$
- Senão se M_f **maior ou igual a 3.0 e menor que 7.0** deverá fazer a P_s e $N_f = (M_f + P_s)/2$
- N_f **igual ou superior** a 5.0 e frequência mínima de 75%:
APROVADO

Instrumentos de acompanhamento e avaliação

- Caso o aluno necessite faltar a uma das provas parciais por motivo de saúde, ele deverá fazer a prova de **segunda chamada** (P_3) (3 de julho)
- O aluno poderá optar por fazer a P_3 para substituir a nota da P_1 ou da P_2 (a substituição da nota será realizada mesmo que a nota substituta seja menor que a nota inicial)
- Casos excepcionais serão tratados individualmente

Instrumentos de acompanhamento e avaliação

Laboratórios

- As atividades realizadas em cada laboratório (com exceção do primeiro) serão pontuadas com nota entre 0.0 e 0.1
- A soma das pontuações obtidas nos laboratórios será acrescentada na média final (M_f)

Bibliografia

- ❶ **The Art of Multiprocessor Programming**, M. Herlihy, N. Shavit, Morgan-Kaufmann, 2008.
- ❷ **Modern Multithreading : Implementing, Testing, and Debugging Multithreaded Java and C++/Pthreads/Win32 Programs**, R. H. Carver and K.-C. Tai, Wiley-Interscience, 2005.
- ❸ **Java Concurrency in Practice**, Brian Goetz, Addison-Wesley, 2006.
- ❹ **Synchronization Algorithms and Concurrent Programming**, G. Taubenfeld, Pearson/Prentice Hall, 2006.
- ❺ **Computer Organization and Design - the hardware/software interface**, D. A. Patterson e J. L. Hennessy, 4ed, Morgan Kaufmann, 2009.