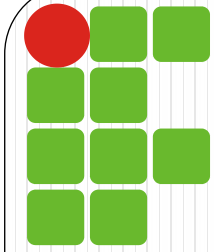


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA
Campus Salvador

Gerência de Processos

Instituto Federal da Bahia
Campus Salvador
INF009 - Sistemas Operacionais
Profª Flávia Maristela

Quantas operações você realiza simultaneamente no seu computador?

Processos

- O que é um processo?

- Programa *vs.* Processos
  - Processo é uma atividade de um programa

- Exemplo: vamos fazer um bolo!
  - Fazer o bolo corresponde a fazer um programa! Então, quais os principais elementos de um programa?

Vamos considerar um exemplo

- Vamos fazer um bolo!

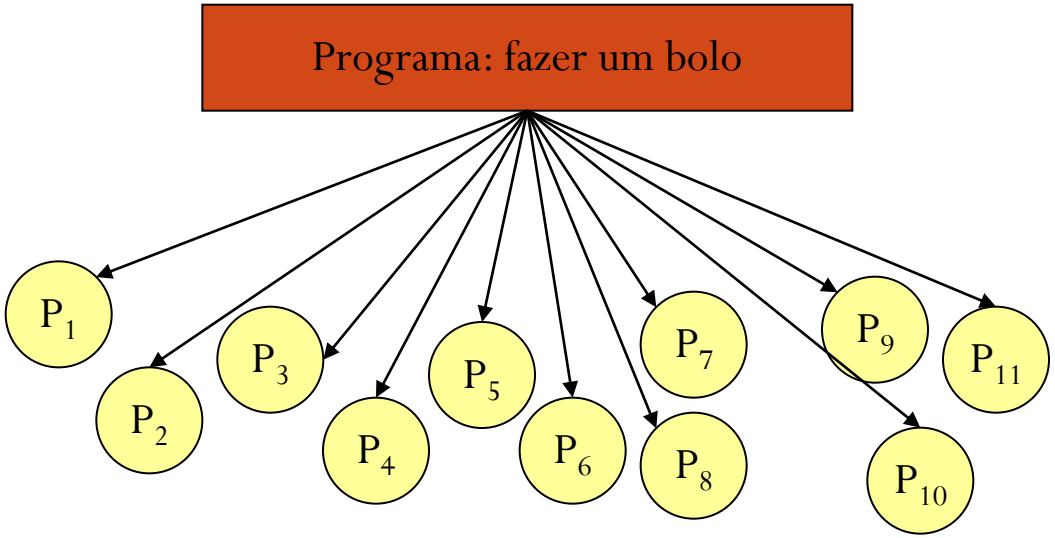
ENTRADA
(-- ingredientes --)

PROCESSAMENTO
(-- algoritmo --)

SAÍDA
(-- bolo --)

1. Bater as claras em neve
2. Misturar açúcar e manteiga
3. Adicionar as gemas
4. Adicionar metade da farinha de trigo
5. Adicionar metade do leite
6. Adicionar a outra metade da farinha de trigo
7. Adicionar a outra metade do leite
8. Ligar o forno a 180°
9. Adicionar as claras
10. Despejar bolo na forma
11. Colocar a forma no forno

Vamos considerar um exemplo



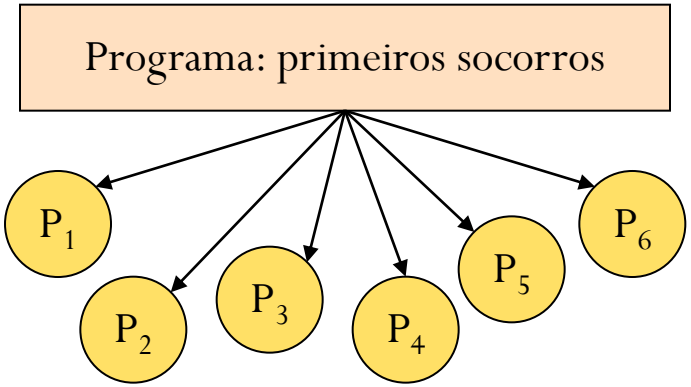
Estados de um processo

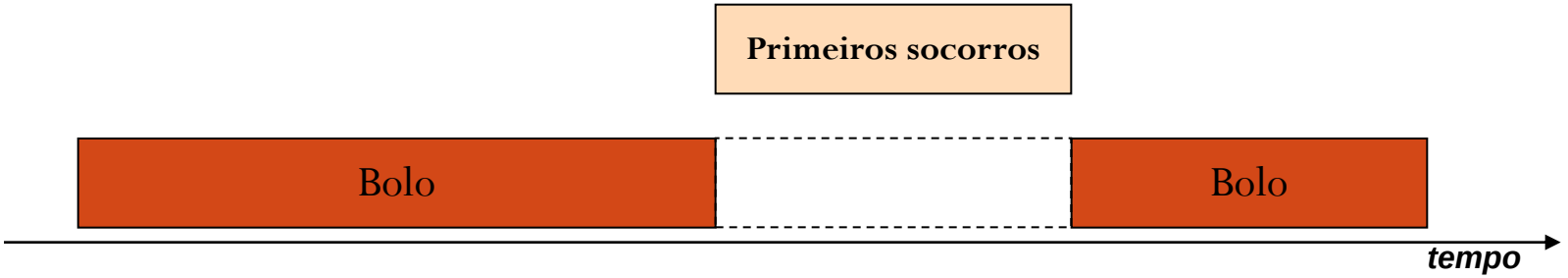
- Clássicos
  - Execução
  - Bloqueado
  - Pronto (Apto)

- Outros estados da literatura
  - Iniciado
  - Finalizado (terminado)

Voltando ao nosso exemplo...

Enquanto o bolo está sendo preparado, o filho do cozinheiro
cai da bicicleta e se machuca! O que fazer?
O cozinheiro vai **interromper** o bolo para dar os primeiros socorros ao filho!





Voltando ao nosso exemplo...

- Estados dos processos:
  - Em execução:
    - Selecionar ingredientes
    - Misturar açúcar e manteiga
    - Fazer curativo
  - Bloqueado
    - Acabou o leite
    - Faltou algodão
  - Apto
    - O bolo pode ser finalizado enquanto ele atende o filho

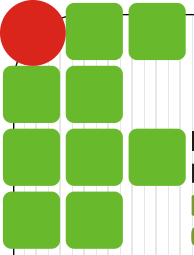
- Mas afinal, quem é que coloca ordem na casa?
  - ESCALONADOR: implementação de baixo nível do sistema operacional que controla o acesso a CPU
  - Em nosso exemplo:
    - ESCALONADOR: a consciência do cozinheiro
    - PROCESSADOR: cozinheiro
    - PROCESSOS: atividades que ele precisava realizar

Será que pode piorar?

Imagine agora uma padaria, onde além do bolo é necessário fazer pães, salgados, doces…

compartilhando os mesmos ingredientes

Isso se chama PROGRAMAÇÃO CONCORRENTE



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA
Campus Salvador

Processos

Porque eu preciso entender este assunto?

- Para entender como um computador consegue executar várias tarefas simultaneamente e **qual o impacto que isso pode ter em meus programas**!

Os programas de ontem...

- Antigamente, os computadores eram máquinas dedicadas:
  - Possuíam apenas um usuário
  - Executavam apenas um programa por vez
    - Programas em execução tinham total controle dos recursos do computador

E os programas de hoje!

- Hoje os computadores:
  - Executam vários programas simultaneamente
  - Podem ser usados por vários usuários
- Isso gerou a necessidade de compartilhar recursos...
- ... e por isso os programas foram divididos em unidades menores.

Sobre os programas...

- Quando ligamos o computador, vários programas começam a ser executados.
  - Programas ativados pelo Sistema Operacional
  - Programas ativados pelo usuário

- Cada um destes programas possui vários ***processos***.

Processos
(-- definição --)

- Definição:
  - Programa em execução

Silberschatz, Tanenbaum

- Processos são entidades independentes entre si, mas *concorrem* aos mesmos recursos do computador.

Processos
(-- estados --)

- Novo
  - O processo está sendo criado, ou seja, seu código está sendo carregado em memória, junto com suas bibliotecas;
  - As estruturas de dados do *kernel* estão sendo atualizadas para permitir sua execução.

- Pronto
  - Processo está em memória, pronto para ser executado, aguardando a disponibilidade do processador;
  - IMPORTANTE: Os processos "prontos" são organizados em uma fila cuja ordem é determinada por algoritmos de escalonamento.

Processos
(-- estados --)

- Executando:
  - Processo está executando suas instruções.

- Bloqueado
  - Processo não pode executar porque depende de recursos ainda não disponíveis (dados, algum tipo de sincronização, a liberação de algum recurso compartilhado);
  - Processo simplesmente espera o tempo passar (em estado de "*sleeping*").

- Terminado
  - A execução do processo foi encerrada e ele pode ser removido da memória do sistema.

Processos
(-- transições --)

- … → **Novo**
  - um novo processo é criado e começa a ser preparado para executar.

- **Novo → Pronto**
  - o novo processo termina de ser carregado em memória, estando pronto para executar.

- **Pronto → Executando**
  - o processo é escolhido pelo escalonador para ser executado, entre os demais processos prontos.

Processos
(-- transições --)

- **Executando → Pronto**
  - esta transição ocorre quando se esgota a fatia de tempo destinada ao processo (***quantum***);
  - Nesse momento o processo não precisa de outros recursos além do processador e por isso volta à fila de "pronto" para esperar novamente a disponibilidade do processador.

- **Executando → Terminada**
  - O processo encerra sua execução ou é abortado em consequência de algum erro (acesso inválido à memória, instrução ilegal, divisão por zero).

  - Em geral, o processo que deseja terminar avisa ao sistema operacional através de uma chamada de sistema.

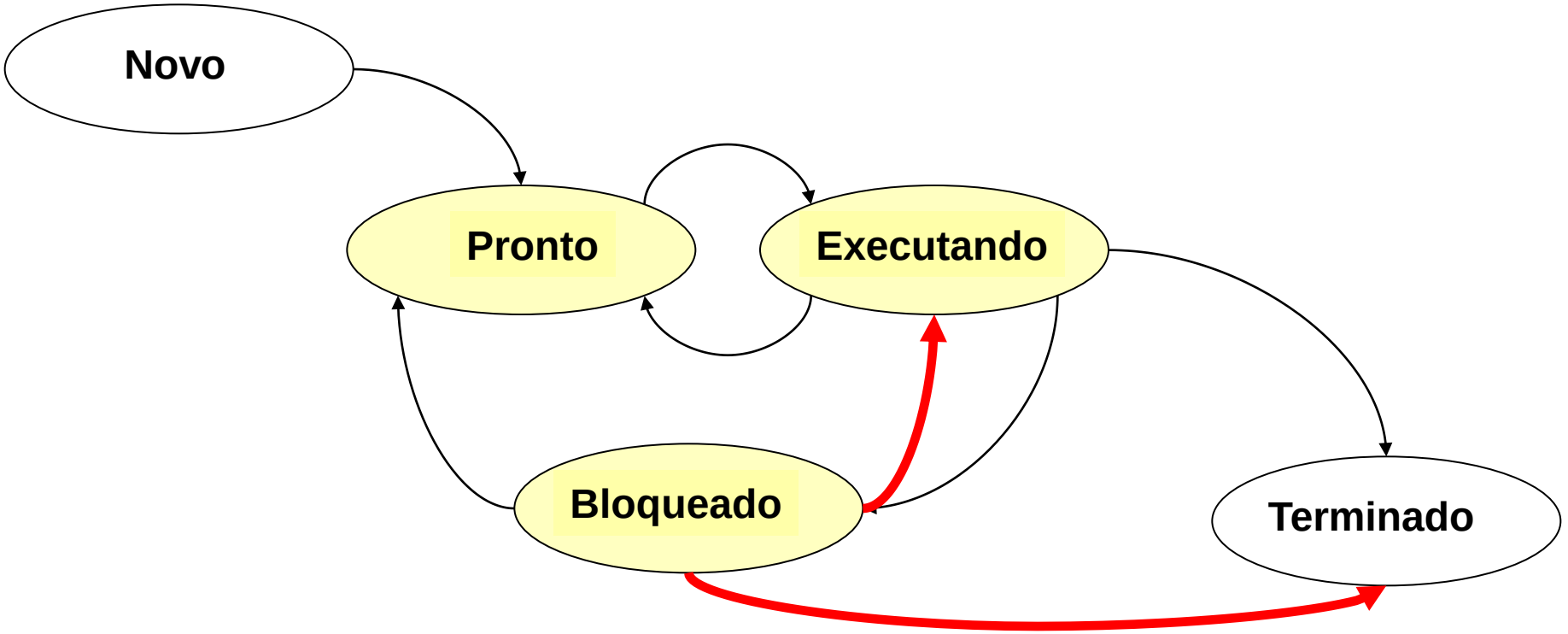
Processos
(-- transições --)

- **Terminado → …**
  - Quando terminado, um processo é removido da memória e seus registros e estruturas de controle no *kernel* são apagados.

- **Executando → Bloqueado**
  - caso o processo em execução solicite acesso a um recurso não disponível, ele abandona o processador e fica bloqueado até o recurso ficar disponível.

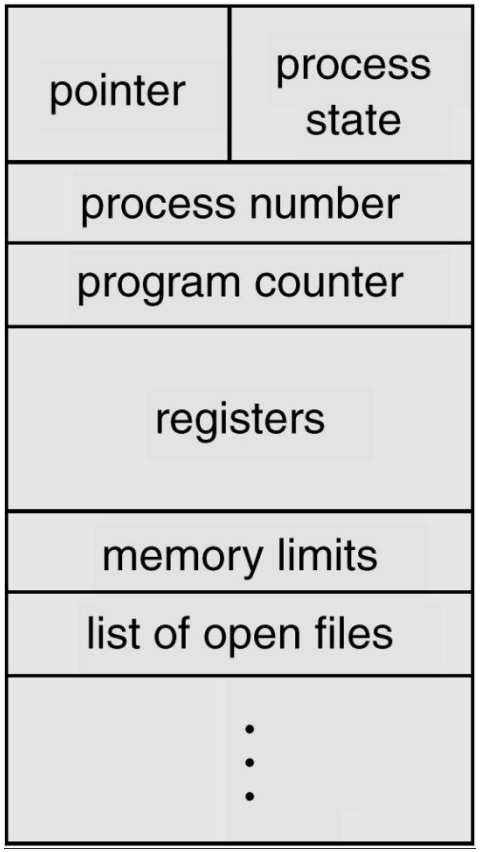
- **Bloqueado → Pronto**
  - quando o recurso solicitado pelo processo se torna disponível, ele pode então voltar ao estado de "pronto".

Processos
(-- transições --)



Processo
(-- *process control block* --)

Representação do processo para o sistema operacional



Silberschatz, capítulo 2

Processo
(-- *process control block* --)

- Informações associadas a cada processo:
  - Estado do processo
  - Valor do Contador de Programa
    - indica a próxima instrução a ser executada
  - Área para guardar valor dos registradores (dados)
  - Dados para gerenciamento da CPU (escalonamento)
  - Dados para gerenciamento de memória
  - Número do processos
  - Informações sobre E/S

Processos
(-- criação --)

- O que motiva a criação de um processo?
  - Inicialização do sistema operacional;
  - Inicialização de um programa;
  - Chamada de sistema;

Processos
(-- execução --)

- Os processos podem executar de duas formas:
  - Em *FOREGROUND*
    - Processos que interagem com os usuários
  - EM *BACKGROUND*
    - Não associados a usuários
    - Possuem funções específicas

Processos
(-- finalização --)

- O que motiva a finalização de um processo?
  - Saída normal;
  - Saída com erro;
  - *Fatal Error* (involuntário);
  - Outro processo (involuntário)

Para a próxima aula

1. O que é um processo?
2. Quais os estados de um processo?
3. Quais os principais atributos de processos?
4. Qual a diferença entre programa e processo?
5. O que é o escalonador? Pelo que ele é responsável?
6. O que motiva a inicialização de um processo?
7. O que pode motivar a finalização de um processo?