Fatec
Araras

Servidores e seus sistemas operacionais

Prof. Orlando Saraiva Júnior
orlando.saraiva@unesp.br

Processos

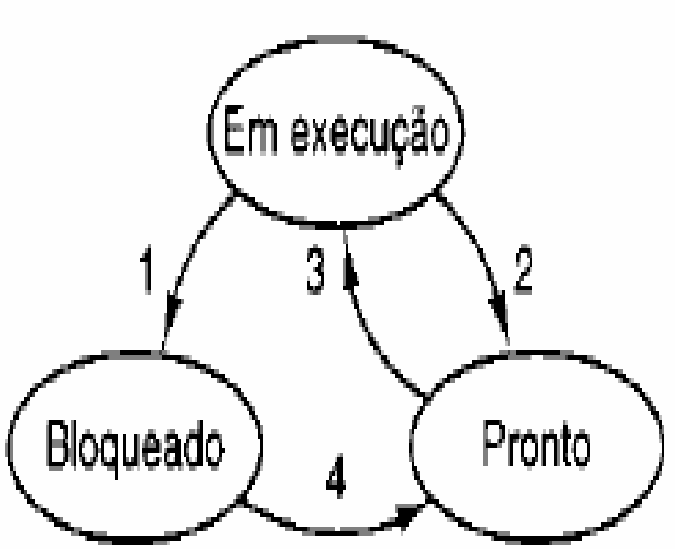
Prof. Me. Orlando Saraiva Júnior

Processos

É uma abstração para um programa em execução.

Estados:
- Ativos (em Execução)
- Suspensos (Bloqueados)
- Prontos

Modos:
- Usuário
- Núcleo



Programa 01

#include <baralho.h>

Pegar o baralho;

Na mão, ordenar por naipes;

Ordenar as cartas em ordem crescente sem tirar a ordem do naipe;

Programa 02

```
#include <baralho.h>
#include <mesa.h>
      Pegar o baralho;
      Separar por naipe na mesa;
      Para x=0;x<3;x++  {
            Ordenar naipe;
     }
```

Programa 03

```
#include <baralho.h>
#include <voluntarios.h>
      Pegar o baralho;
      Para x=0;x<3;x++  {
            Passar a carta para o voluntário com um tipo de naipe;
            trabalha_voluntario()
    }
       devolve_voluntario()

 trabalha_voluntario() {
        Ordenar as cartas em ordem crescente;
  }

  devolve_voluntario() {
        Retornar cartas ordenadas;
  }
```

Programas

Fatec Araras

Qual foi o tempo de execução do:

Programa 1:

Programa 2:

Programa 3:

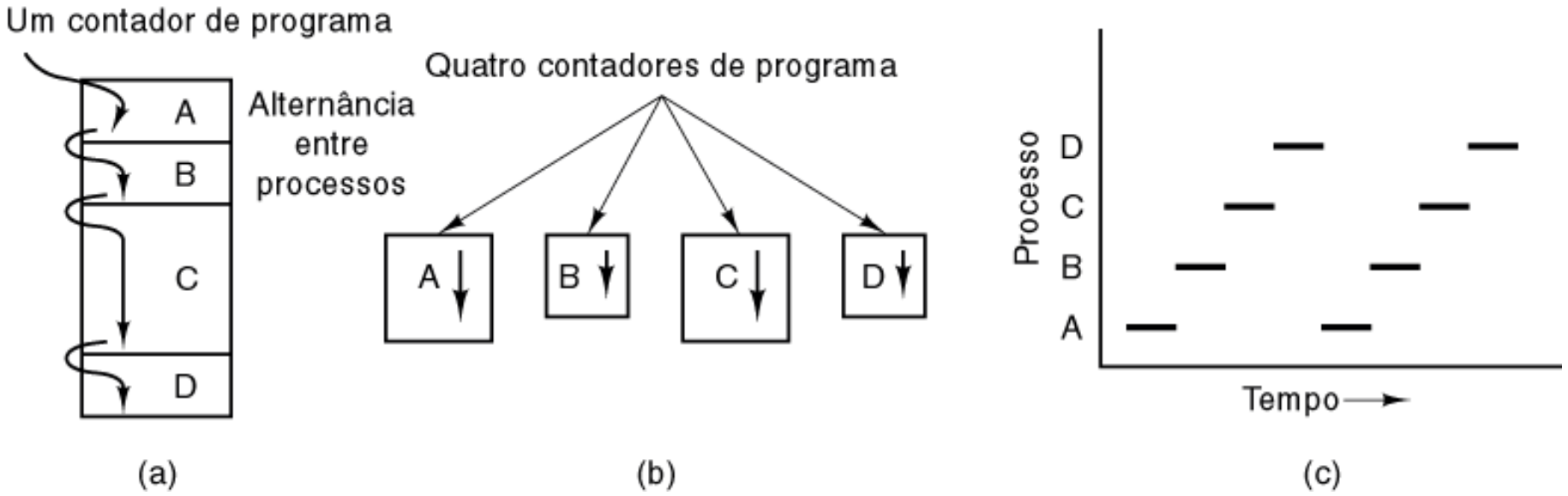
Processo

Um processo é um programa em execução acompanhado dos valores atuais do contador de programas, dos registradores e das variáveis.

Com a alternância da CPU entre os processos, a taxa que o processo realiza sua computação não é uniforme e provavelmente **não reproduzível**.

Processos não podem ser programados com hipóteses predefinidas de temporização.

Processo



(a) Multiprogramação de quatro programas
(b) Modelo conceitual de 4 processos sequenciais, independentes
(c) Somente um programa está ativo a cada momento

Processos

Principais eventos que levam à criação de processos

1) Início do sistema;
2) Execução de chamada ao sistema de criação de processos;
3) Solicitação do usuário para criar um novo processo;
4) Início de um job em lote;

Processos

Processos em primeiro plano:

Interage com o usuário;

Processos em segundo plano:

Não associados a um usuário, mas a uma função específica;

Daemons (lê-se dímons);

Processos

Fatec Araras

Principais eventos que levam ao término de processos

1) Saída normal (voluntária)
Processo terminou a tarefa;a ao sistema de criação de processos;

2) Saída por erro (voluntária)
Chamada ao sistema exit (Unix) ou ExitProcess (Windows)

3) Erro fatal (involuntário)
Erro de programação, instrução ilegal, acesso à memória inexistente, etc...

4) Cancelamento por um outro processo (involuntário)
Um processo solicita ao sistema operacional a finalização de outro processo.
Chamada kill (Unix) ou TerminateProcess (Windows)

Processos

Pai cria um processo filho, processo filho pode criar seu próprio processo

Formam uma hierarquia
UNIX chama isso de “grupo de processos”

Windows não possui o conceito de hierarquia de processos
Todos os processos são criados iguais.
Ao pai, é dado um identificador especial (*handle*)

DICA: comando pstree

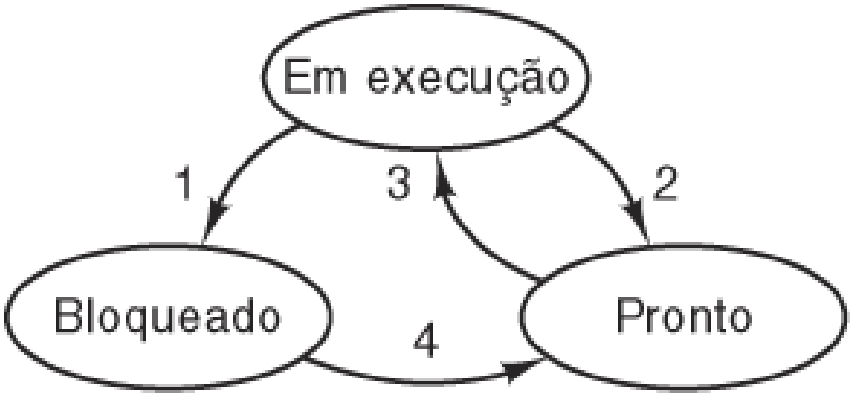
Processos
Estados de um processo

Em execução: realmente usando a CPU naquele instante

Pronto: Executável, temporariamente parado para dar lugar a outro processo

Bloqueado: incapaz de executar enquanto um evento externo não acontecer

Processos
Estados de um processo



1. O processo bloqueia aguardando uma entrada
2. O escalonador seleciona outro processo
3. O escalonador seleciona esse processo
4. A entrada torna-se disponível

O sistema operacional mantem uma tabela (vetor de estruturas) chamada tabela de processos.

Processos

Campos de um processo

Fatec Araras

Gerenciamento de processos	Gerenciamento de memória	Gerenciamento de arquivos
Registradores	Ponteiro para o segmento de código	Diretório-raiz
Contador de programa	Ponteiro para o segmento de dados	Diretório de trabalho
Palavra de estado do programa	Ponteiro para o segmento de pilha	Descritores de arquivos
Ponteiro de pilha		Identificador (ID) do usuário
Estado do processo		Identificador (ID) do grupo
Prioridade		
Parâmetros de escalonamento		
Identificador (ID) do processo		
Processo pai		
Grupo do processo		
Sinais		
Momento em que o processo iniciou		
Tempo usado da CPU		
Tempo de CPU do filho		
Momento do próximo alarme		

Fatec
Araras

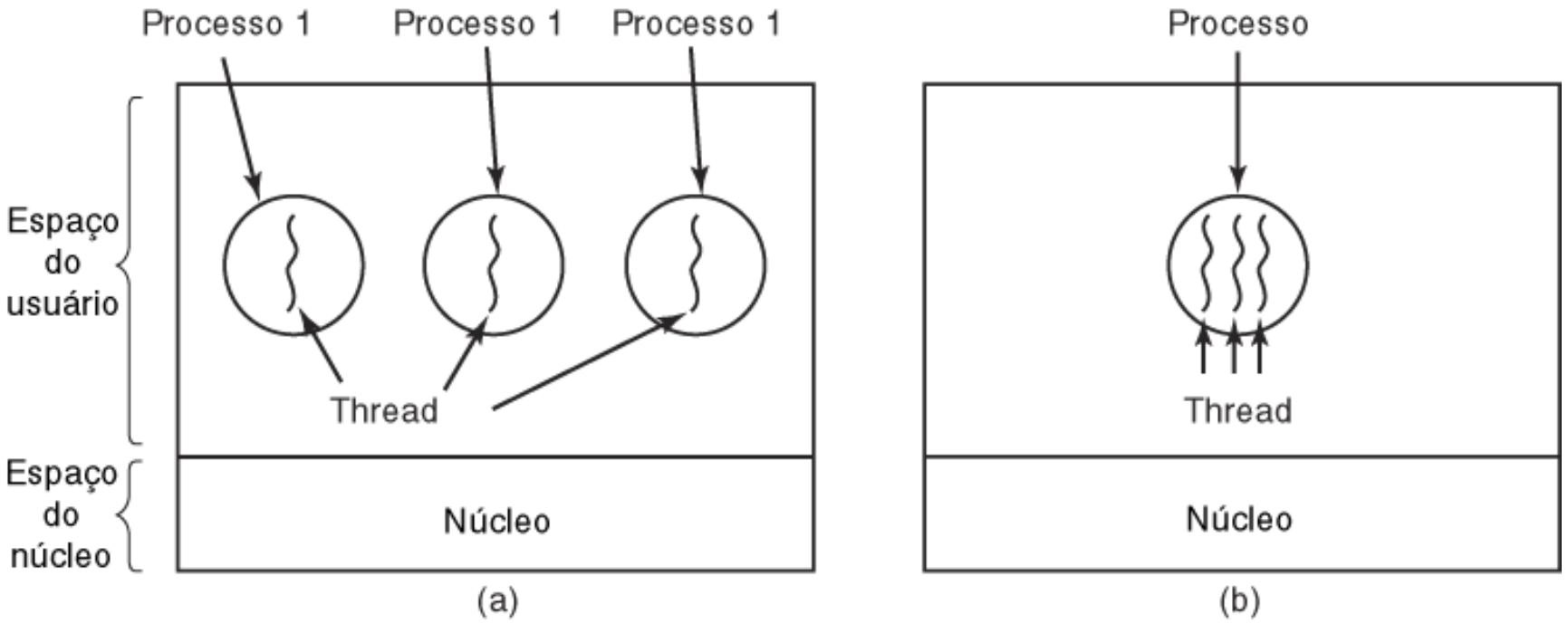
Threads

Prof. Me. Orlando Saraiva Júnior

Threads

Thread (linha de execução), é uma forma de um processo dividir a si mesmo em duas ou mais tarefas que podem ser executadas simultaneamente.
Threads mantêm algumas das propriedades do processo, por isso, são chamados também de processos leves (*lightweight process*).
O termo *multithread* é usado para descrever a situação que permite a existência de múltiplos *threads*.

Threads



(a) Três processos, cada um com uma *thread*
(b) Um processo com três *threads*

Threads
Estados de uma thread

- Em execução
- Pronto
- Bloqueado
- Finalizado

Threads
Estados de uma thread

Itens por processo	Itens por thread
Espaço de endereçamento	Contador de programa
Variáveis globais	Registradores
Arquivos abertos	Pilha
Processos filhos	Estado
Alarmes pendentes	
Sinais e tratadores de sinais	
Informação de contabilidade	

1) Itens compartilhados em um processo
2) Alguns campos restritos de cada thread.

Threads
Motivos para usar thread

Fatec Araras

- Compartilhar dados do processo, sem criar processos filhos;

- Mais fáceis de criar e destruir;

- Quando aplicativo exige grande quantidade de processamento ou E/S, *threads* podem acelerar a aplicação;

Fatec
Araras

Hora da prática

Prof. Orlando Saraiva Júnior
orlando.saraiva@unesp.br