

Conceito de processos

Sumário

- 3.1 Introdução
 - 3.1.1 Definição de processo
- 3.2 Estados de processo: ciclo de vida de um processo
- 3.3 Gerenciamento de processo
 - 3.3.1 Estados de processo e estados de transição
 - 3.3.2 Blocos de controle de processo (PCBs)/Descritores de processo
 - 3.3.3 Operações de processo
 - 3.3.4 Suspende e retomar
 - 3.3.5 Chaveamento de contexto
- 3.4 Interrupções
 - 3.4.1 Processamento de interrupções
 - 3.4.2 Classes de interrupção
- 3.5 Comunicação interprocessos
 - 3.5.1 Sinais
 - 3.5.2 Troca de mensagens

Objetivos

■ Este capítulo apresenta:

O conceito de um processo.

O ciclo de vida de um processo.

Estados de processo e transições de estado.

Blocos de controle de processos (PCBs)/descritores de processos.

Como os processadores transitam entre processos via chaveamento de contexto.

Como interrupções habilitam o hardware a se comunicar com o software.

Como processos conversam uns com os outros via comunicação interprocessos (IPC).

3.1 Introdução

- **Os computadores executam várias operações ao mesmo tempo.**

Por exemplo, compilar um programa, enviar um arquivo para a impressora, exibir uma página Web, reproduzir músicas e receber mensagens de correio eletrônico.

Os processos permitem que os sistemas executem e monitorem atividades simultâneas.

Os processos transitam entre estados de processo.

Os sistemas operacionais executam operações por meio de processos, como criar, destruir, suspender, retomar e acordar.

3.1 Introdução

- Analogia entre um **Processo** e um **Cozinheiro**
- Imagine um engenheiro com dotes culinários fazendo um bolo:
 - Receita = programa
 - Engenheiro cozinheiro = processador (CPU)
 - Ingredientes = dados de entrada.
 - Processo é a atividade desempenhada pelo cozinheiro em ler a receita, buscar os ingredientes e assar o bolo.



3.1.1 Definição de processo

■ Um programa em execução

Um processo tem seu próprio espaço de endereço, que consiste em:

Região de texto

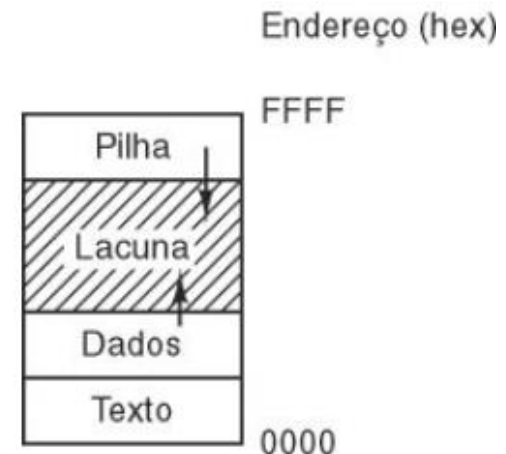
Armazena o código que o processador executa.

Região de dados

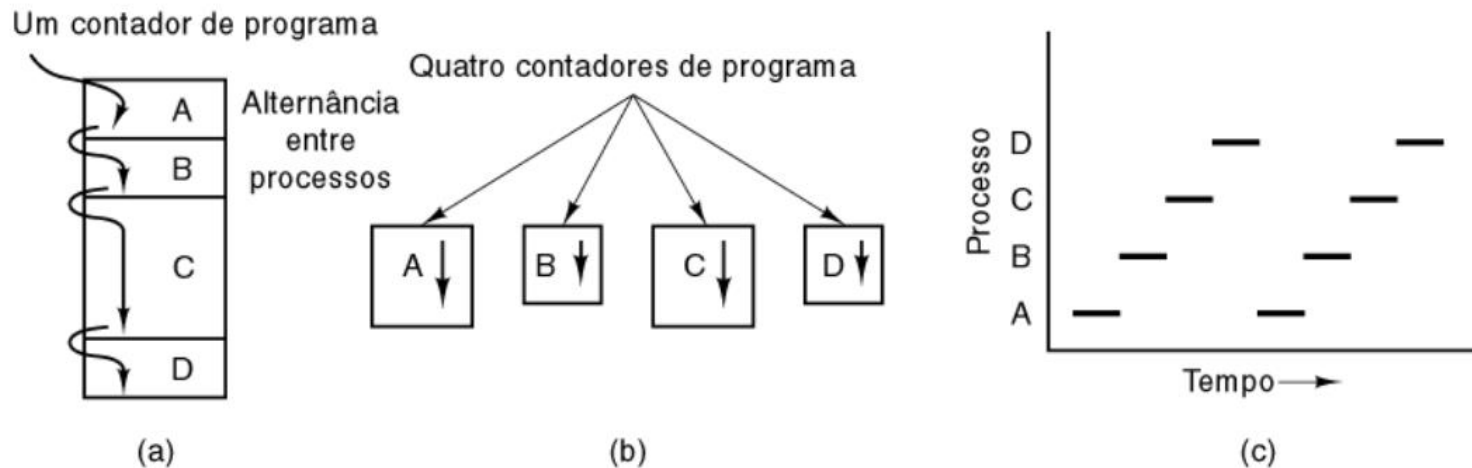
Armazena variáveis e memória alocada dinamicamente.

Região de pilha

Armazena instruções e variáveis locais para chamadas ativas ao procedimento.



3.1.2 O modelo de processo



- Multiprogramação de quatro programas
- Modelo conceitual de 4 processos sequenciais, independentes
- Somente um programa está ativo a cada momento

3.2 Estados de processo: ciclo de vida de um processo

- **Criados**

- Início do sistema
- Chamada de Sistema de criação de processo por um processo em execução
- Requisição do usuário para criar um novo processo
- Início de uma tarefa em lote (computadores de grande porte)

- **Destruídos**

- Saída normal (voluntária)
- Saída por erro (voluntária)
- Erro fatal (involuntário)
- Cancelamento por outros processos (involuntária)

- **Processos CPU-bound**

- Ciclo de processador > Ciclo de E/S

- **Processos I/O bound**

- Ciclo de E/S > Ciclo de processador

3.3 Estados de processo: ciclo de vida de um processo

- Um processo passa por uma série de estados de processo distinto:

Estado de *execução*

O processo está sendo executado em um processador.

Estado “*de pronto*”

O processo poderia ser executado em um processador se houvesse algum disponível.

Estado *bloqueado*

O processo está aguardando a ocorrência de algum evento para prosseguir.

- O sistema operacional mantém uma lista de *prontos* e uma lista de *bloqueados* para armazenar referências a processos que não estão sendo executados.

3.3 Gerenciamento de processo

- **Os sistemas operacionais prestam certos serviços fundamentais aos processos. Por exemplo:**
 - Criam processos.
 - Destroem processos.
 - Suspendem processos.
 - Retomam processos.
 - Mudam a prioridade de um processo.
 - Bloqueiam processos.
 - Acordam (ativam) processos.
 - Despacham processos.
 - Capacitam os processos à comunicação interprocessos (IPC).

3.3.1 Estados de processo e estados de transição

■ Estados de processo

O ato de designar um processador ao primeiro processo da lista de prontos é denominado despacho.

O sistema operacional usam um temporizador de intervalo para permitir que um processo seja executado durante um intervalo de tempo específico ou quantum.

A multitarefa cooperativa permite que todo processo seja executado até o fim.

■ Estados de transição: Até agora, existem quatro estados de transição possíveis:

Quando um processo é despachado, ele transita de *pronto* para *em execução*.

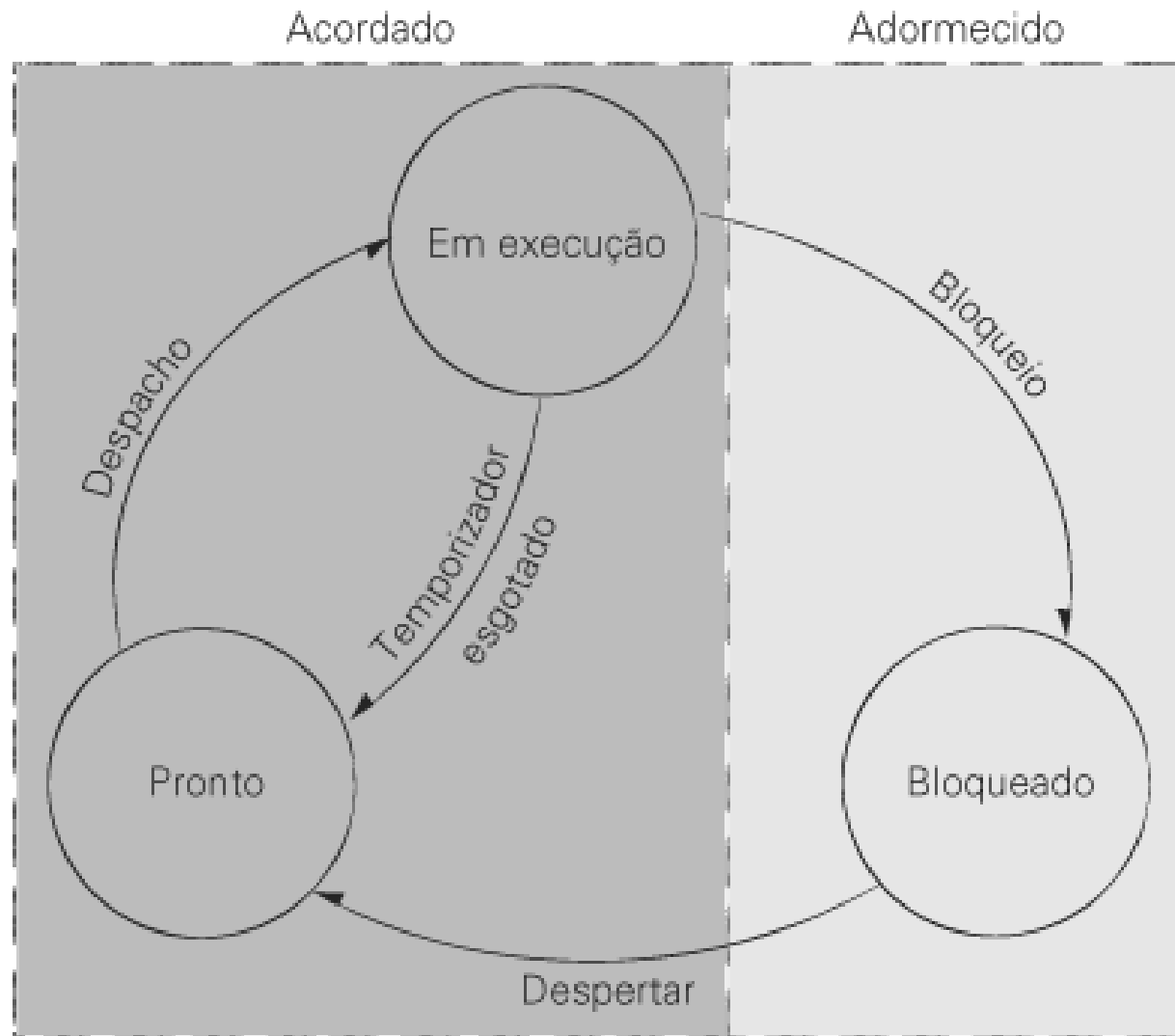
Quando o quantum de um processo expira, ele transita de *em execução* para *pronto*.

Quando um processo é bloqueado, ele transita de *em execução* para *bloqueado*.

Quando um evento ocorre, ele transita de *bloqueado* para *pronto*.

3.3.1 Estados de processo e estados de transição

Figura 3.1 Transições de estado de processo.



3.3.2 Blocos de controle de processo (PCBs) /Descritores de processo

- **Os PCBs mantêm as informações que o sistema operacional precisa para gerenciar o processo.**

Normalmente, chamado de Escalonador ele incluem as seguintes informações:

- Número de identificação de processo (PID)

- Estado do processo

- Contador de programa

- Prioridade de escalonamento

- Credenciais

- Um ponteiro para o processo-pai

- Ponteiros para os processos-filho

- Ponteiros para localizar os dados e instruções do processo na memória

- Ponteiros para recursos alocados

3.3.2 Blocos de controle de processo (PCBs) / Descritores de processo

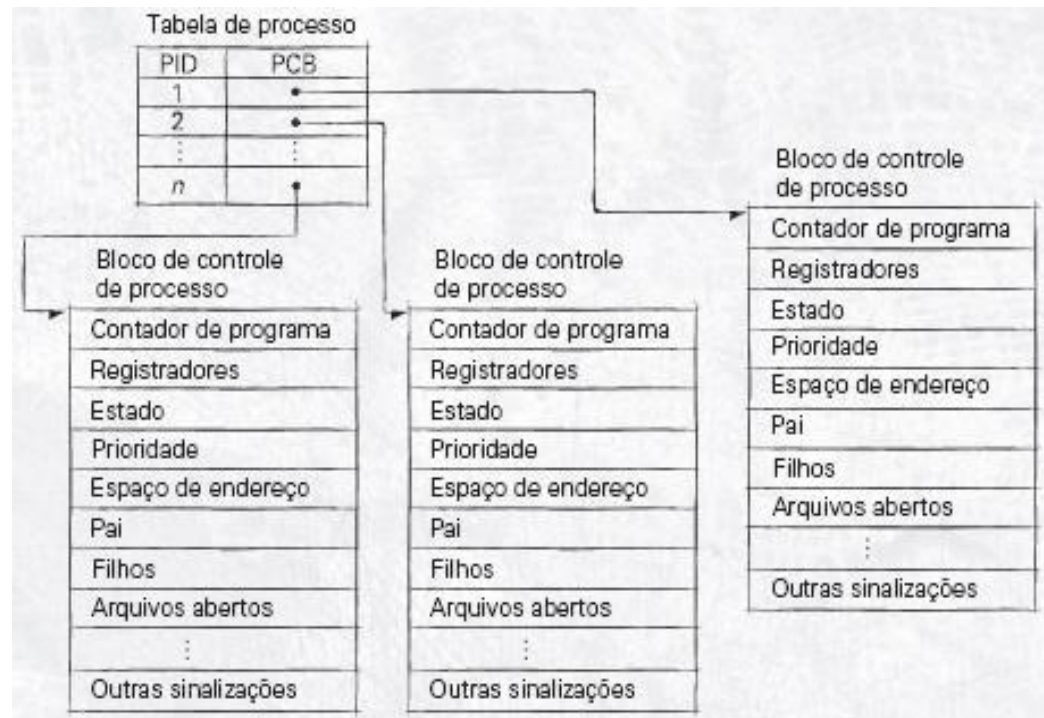
■ Tabela de processos

O sistema operacional mantém ponteiros para cada PCB do processo em uma tabela de processos no âmbito total do sistema ou por usuário.

Permite acesso rápido aos PCBs.

Quando um processo é encerrado, a tabela de processos retira o processo da tabela e disponibiliza todos os seus recursos.

Figura 3.2 Tabela de processos e blocos de controle de processo.



3.3.3 Operações de processo

Um processo pode gerar um novo processo.

O processo de criação é chamado de processo-pai.

O processo criado é chamado de processo-filho.

Cada processo-filho é criado por exatamente um processo-pai.

Quando um processo-pai é desfeito, os sistemas operacionais em geral respondem de uma destas duas formas:

Destroem todos os processos-filho desse processo-pai.

Permitem que os processos-filho prossigam independentemente dos processos-pai.

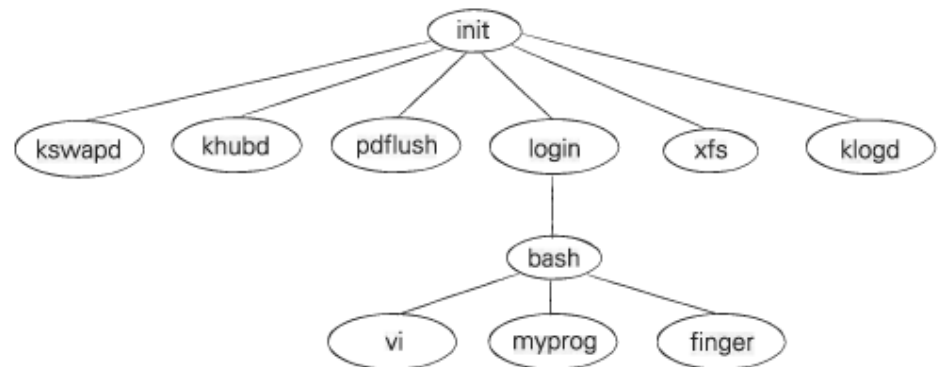
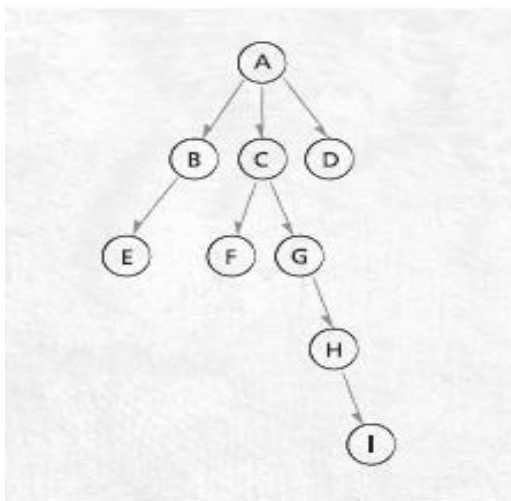


Figura 3.4 Hierarquia de processo no Linux.

Figura 3.3 Hierarquia de criação de processo.

3.3.4 Suspende e retomar

■ **Suspensão de um processo**

O processo é retirado indefinidamente da disputa por tempo em um processador, sem ser destruído.

É útil para detectar ameaças à segurança e para depuração de software.

A suspensão pode ser iniciada pelo processo que está sendo suspenso ou por outro processo.

O processo suspenso precisa ser retomado por outro processo.

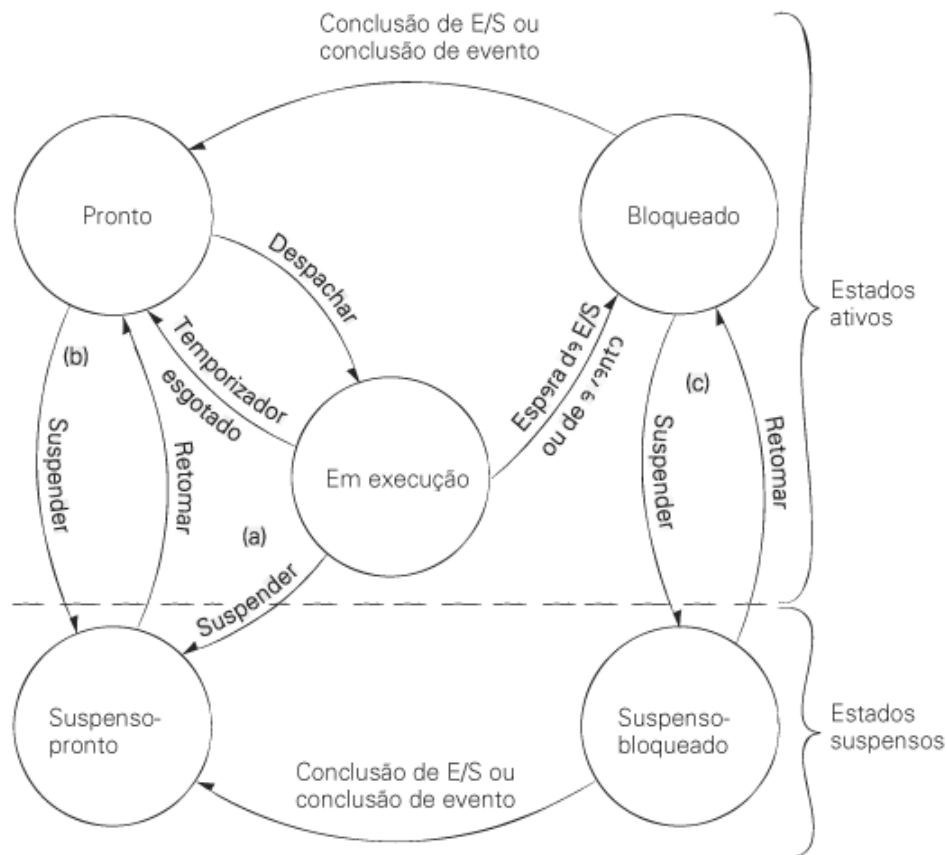
Existem dois estados de suspenso:

Suspenso-pronto

Suspenso-bloqueado

3.3.4 Suspender e retomar

Figura 3.5 Transições de estado de processo com suspensão e retomada.



3.3.5 Chaveamento de contexto

■ Chaveamento de contexto

O sistema operacional realiza um chaveamento de contexto para interromper um processo *em execução* e começar a executar um processo previamente *pronto*.

Salva o contexto de execução do processo *em execução* no PCP(Process Control Blocks) desse processo.

Carrega o contexto de execução anterior do processo *pronto* no PCP desse último.

Deve ser transparente aos processos.

O processador não pode realizar nenhuma computação “útil”.
O sistema operacional precisa, portanto, diminuir o tempo de chaveamento de contexto.

É executado no hardware por algumas arquiteturas.

3.3.5 Chaveamento de contexto

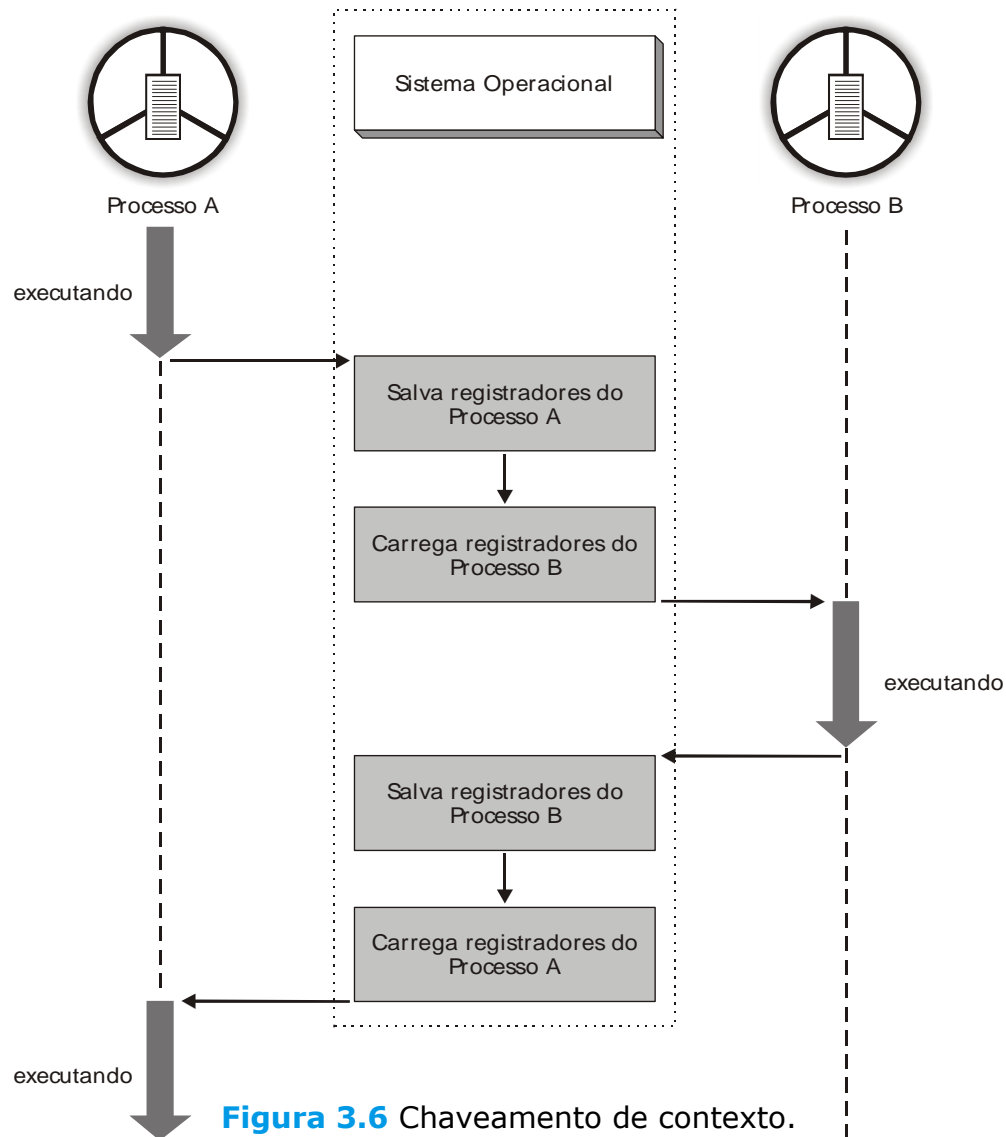


Figura 3.6 Chaveamento de contexto.

3.4 Interrupções

- **As interrupções habilitam o software a responder a sinais do hardware.**

Podem ser iniciadas por um processo em execução.

A interrupção é denominada desvio (trap).

É considerada síncrona em relação à operação do processo.

Por exemplo, quando um processo tenta realizar uma ação ilegal, como dividir por zero ou referir-se a uma memória protegida.

Podem ser iniciadas por algum evento que pode ou não estar relacionado ao processo em execução.

Nesse caso, é considerada assíncrona em relação à operação do processo.

Por exemplo, quando uma tecla é pressionada no teclado ou o mouse é movido.

Pouco esforço

- **A sondagem (polling) é uma abordagem alternativa.**

O processador solicita repetidamente o status de cada dispositivo.

Visto que a complexidade do sistema aumenta, o esforço também aumenta.

3.4.1 Processamento de interrupções

■ Tratamento de interrupções

Assim que recebe uma interrupção, o processador conclui a execução da instrução corrente e faz uma pausa no processo corrente.

Em seguida, o processador executa uma das funções de tratamento de interrupção do núcleo.

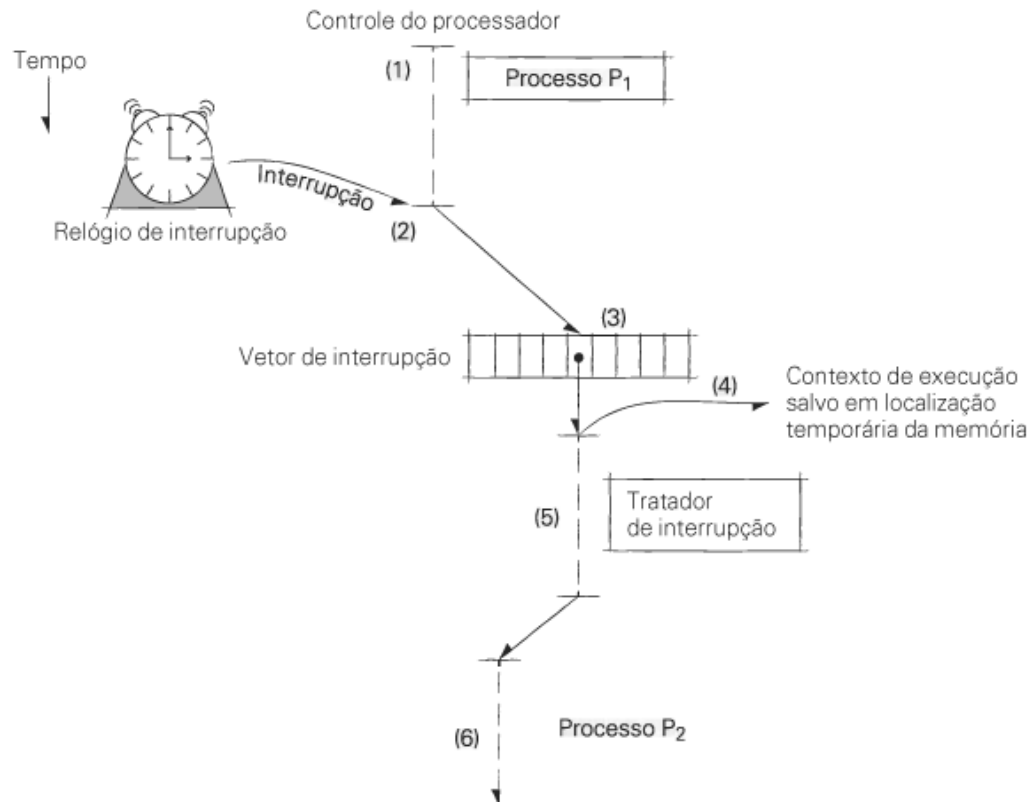
O tratador de interrupção determina como o sistema deve responder.

Os tratadores de interrupção são armazenados em um conjunto de ponteiros denominado vetor de interrupção.

Após a conclusão do tratador de interrupção, o processo interrompido é restaurado e executado ou o processo seguinte é executado.

3.4.1 Processamento de interrupções

Figura 3.7 Tratamento de interrupções.



3.4.2 Classes de interrupção

■ As interrupções suportadas dependem da arquitetura do sistema.

A especificação IA-32 distingue os dois tipos de sinal que um processador pode receber:

Interrupções

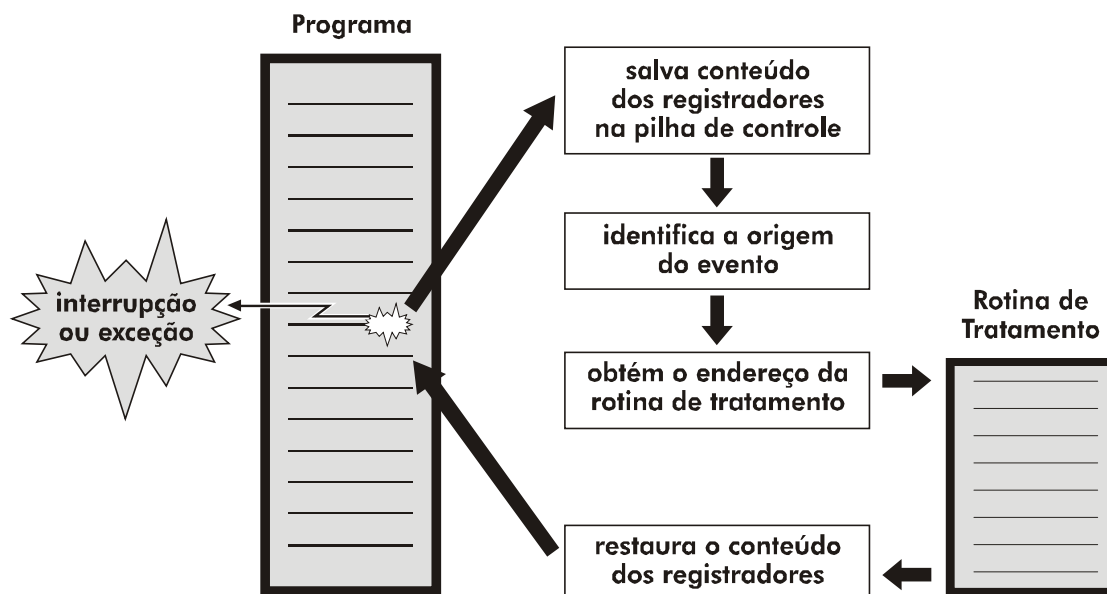
Notificam o processador de que ocorreu um evento ou que o status de um dispositivo externo mudou.

São geradas por dispositivos externos a um processador.

Exceções

Indicam que ocorreu um erro, seja no hardware, seja em decorrência de uma instrução de software.

São classificadas como falhas, desvios ou abortos.



3.4.2 Classes de interrupção

Figura 3.8 Tipos comuns de interrupção reconhecidos pela arquitetura Intel IA-32.

<i>Tipo de interrupção</i>	<i>Descrição das interrupções de cada tipo</i>
E/S	São iniciadas pelo hardware de entrada/saída. Notificam ao processador que o status de um canal ou dispositivo mudou. Interrupções de E/S são causadas quando uma operação de E/S é concluída, por exemplo.
Temporizador	Um sistema pode conter dispositivos que geram interrupções periodicamente. Essas interrupções podem ser usadas para tarefas como controle de tempo e monitoração de desempenho. Temporizadores também habilitam o sistema operacional a determinar se o quantum de um processo expirou.
Interrupções interprocessadores	Essas interrupções permitem que um processador envie uma mensagem a outro em um sistema multiprocessador.

3.4.3 Classes de exceções

Figura 3.9 Classes de exceções do Intel IA-32.

Classes de exceções

Descrição das exceções de cada classe

Falha

Causadas por uma vasta gama de problemas que podem ocorrer, enquanto as instruções em linguagem de máquina de um programa são executadas. Entre esses problemas, estão a divisão por zero, dados (que estão sendo utilizados) no formato errado, tentativa de executar um código de operação inválido, tentativa de referência a uma localização de memória que está fora dos limites da memória real, tentativa de um processo de usuário de executar uma instrução privilegiada, e tentativa de referir-se a um recurso protegido.

Desvio

Gerados por exceções como transbordamento (quando o valor armazenado por um registrador excede à capacidade do registrador) e quando o controle do programa chega a um ponto de parada no código.

Aborto

Ocorrem quando o processador detecta um erro do qual o processo não pode se recuperar. Por exemplo, quando a própria rotina de tratamento de exceções causa uma exceção, o processador pode não conseguir tratar ambos os erros seqüencialmente, o que é denominado exceção de dupla falha, que extingue o processo que lhe deu início.

3.5 Comunicação interprocessos

Muitos sistemas operacionais fornecem mecanismos para comunicações interprocessos (IPC).

Os processos precisam se comunicar uns com os outros em ambientes de multiprogramação ou de rede.

Por exemplo, um navegador Web pode recuperar dados de um servidor remoto.

É essencial para processos que precisam coordenar (sincronizar) atividades para alcançar uma meta comum.

1. Sinais

2. Troca de mensagens

3.5.1 Sinais

São interrupções de software que notificam o processo de que um evento ocorreu.

Não permitem que os processos especifiquem dados para trocar com outros processos.

Os processos podem capturar, ignorar ou mascarar um sinal.

Um processo captura um sinal especificando uma rotina que o sistema operacional chama quando libera o sinal.

Um processo ignora um sinal dependendo da ação-padrão do sistema operacional para tratá-lo.

Um processo mascara um sinal instruindo o sistema operacional a não liberar sinais desse tipo até que o processo bloqueie a máscara do sinal.

3.5.2 Troca de mensagens

Comunicação interprocessos baseada em mensagens

As mensagens podem ser passadas em uma direção por vez.

Um processo é o emissor e o outro é receptor.

A troca de mensagens pode ser bidirecional.

Cada processo pode agir como emissor ou como receptor.

As mensagens podem ser bloqueantes ou não bloqueantes.

As bloqueantes requerem que o receptor notifique o emissor ao receber a mensagem.

As não bloqueantes permitem que o emissor prossiga com outro processamento.

Uma implementação popular na troca de mensagens é um pipe:

Uma região da memória protegida pelo sistema operacional que serve como buffer, permitindo que dois ou mais processos troquem dados.

3.5.2 Troca de mensagens

IPC em sistemas distribuídos

As mensagens, ao ser transmitidas, podem apresentar falhas ou perderem-se.

Os protocolos de confirmação confirmam que as transmissões foram devidamente recebidas.

Mecanismos de tempo de espera retransmitem as mensagens se a confirmação não for recebida.

Processos denominados de modo ambíguo fazem com que as mensagens sejam referenciadas incorretamente.

As mensagens são transmitidas entre computadores usando portas numeradas que atendem aos processos, evitando esse problema.

A segurança é um problema sério.

Garantia de autenticação.