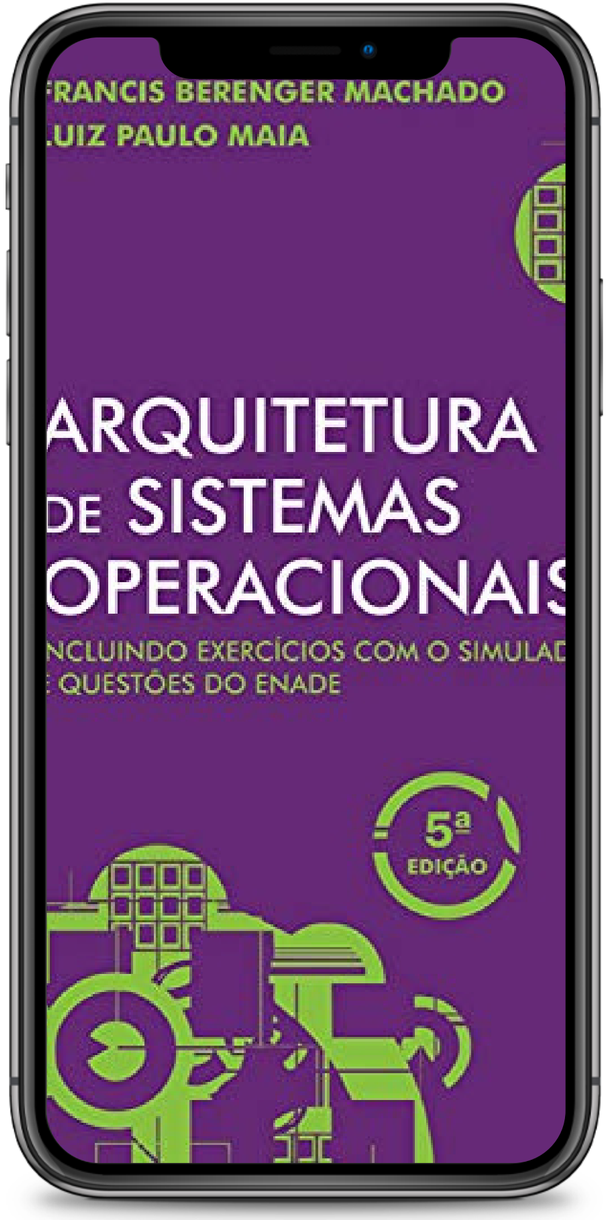

Arquitetura de Sistemas Operacionais

Capítulo 3 - Concorrência

Sumário

3.1 Introdução

O que é Concorrência em Arquitetura de Sistemas Operacionais?

A concorrência em arquitetura de sistemas operacionais refere-se à capacidade do sistema operacional de gerenciar e executar múltiplas tarefas ou processos de forma simultânea e eficiente. Isso é fundamental para garantir um uso eficaz dos recursos de hardware e oferecer uma experiência de computação multitarefa aos usuários.

Neste arquivo será apresentado primeiramente a diferença entre Sistemas Monoprogramáveis e Multiprogramáveis, depois serão apresentados conceitos fundamentais que possibilitam a implementação da concorrência como interrupções e exceções, buffering, spooling e reentrância.

3.2 Sistemas Operacionais Monoprogramáveis X Multiprogramáveis

O que são Sistemas Operacionais Monoprogramáveis?

Sistemas monoprogramáveis são sistemas operacionais que permitem a execução de apenas um programa por vez.

O que são Sistemas Operacionais Multiprogramáveis?

Sistemas multiprogramáveis são sistemas operacionais que permitem a execução simultânea de múltiplos programas em um computador, compartilhando recursos como CPU e memória.

Diferença entre Monoprogramáveis X Multiprogramáveis

A principal diferença entre Sistemas Monoprogramáveis e Multiprogramáveis, é que os multiprogramáveis têm um melhor desempenho em comparação com sistemas monoprogramáveis, pois permitem a execução simultânea de vários programas, aumentando a utilização de recursos de forma concorrente e eficiente.

3.3 Interrupções e Exceções

Interrupções

As interrupções desempenham um papel fundamental na implementação da concorrência em computadores e são essenciais para sistemas multiprogramáveis. Os sistemas operacionais utilizam interrupções para coordenar a execução de suas rotinas e dos programas dos usuários, além de controlar dispositivos.

Uma interrupção é sempre causada por um evento externo ao programa em execução, independente da instrução em andamento. Por exemplo, uma interrupção pode ocorrer quando um dispositivo informa ao processador que uma operação de entrada/saída foi concluída, exigindo que o programa seja interrompido para lidar com essa situação.

Após a execução de cada instrução, a unidade de controle verifica se ocorreu alguma interrupção. Em caso afirmativo, o programa é interrompido e o controle é redirecionado para uma rotina específica chamada de "rotina de tratamento de interrupção". Para que o programa possa continuar a ser executado após a interrupção, é necessário preservar informações cruciais, como o conteúdo dos registradores, que devem ser restaurados durante o processo de tratamento da interrupção.

Fig. 3.2 **Mecanismos de interrupção e exceção.**

Exceções

Exceções são eventos semelhantes a interrupções, mas a diferença crucial é a causa subjacente. Elas ocorrem devido à execução de instruções no próprio programa, como divisões por zero ou estouros em operações aritméticas. A principal distinção é que exceções são síncronas, resultando diretamente da execução do programa atual. Isso as torna previsíveis e limitadas a uma ocorrência por vez. Quando uma exceção é acionada, o programa é interrompido e o controle é transferido para uma rotina de tratamento correspondente, assim como nas interrupções. Cada tipo de exceção possui sua própria rotina de tratamento específica.

3.4 Operações de Entrada/Saída

- O loader, também conhecido como carregador, é uma ferramenta que carrega programas na memória principal para execução. Existem dois tipos de loaders: absoluto e relocável. No absoluto, apenas o endereço inicial e o tamanho do programa são necessários, enquanto no relocável, o programa pode ser carregado em qualquer posição de memória, com o loader ajustando os endereços para garantir o funcionamento correto.

- Antes dos controladores, o processador gerenciava operações de entrada/saída (E/S) de duas maneiras principais: E/S controlada por programa, onde o processador permanecia ocupado sincronizando-se com o periférico, e Polling, onde o sistema operacional verificava periodicamente o estado dos dispositivos, resultando em interrupções frequentes.

- O mecanismo de interrupção melhorou a eficiência das operações de E/S ao permitir que o controlador interrompesse o processador para informar o término da operação, introduzindo a E/S controlada por interrupção. Isso liberava o processador para outras tarefas e otimizava o uso da CPU.

Operações de Entrada/Saída

- O DMA (Direct Memory Access) aprimorou ainda mais a eficiência, permitindo a transferência de dados entre a memória principal e dispositivos de E/S sem intervenção constante do processador, melhorando a eficiência do sistema.

- O canal de entrada e saída, uma extensão do conceito DMA, foi introduzido pela IBM no Sistema 7094. É um processador especializado que executa programas de E/S, permitindo um controle completo sobre essas operações. Isso elimina a necessidade de o processador principal executar operações de E/S diretamente.

- A evolução do canal de E/S permitiu que ele tivesse sua própria memória, o que reduz a intervenção da unidade central de processamento (UCP) e é chamado de processador de entrada e saída. Ambos os termos são usados de forma intercambiável.

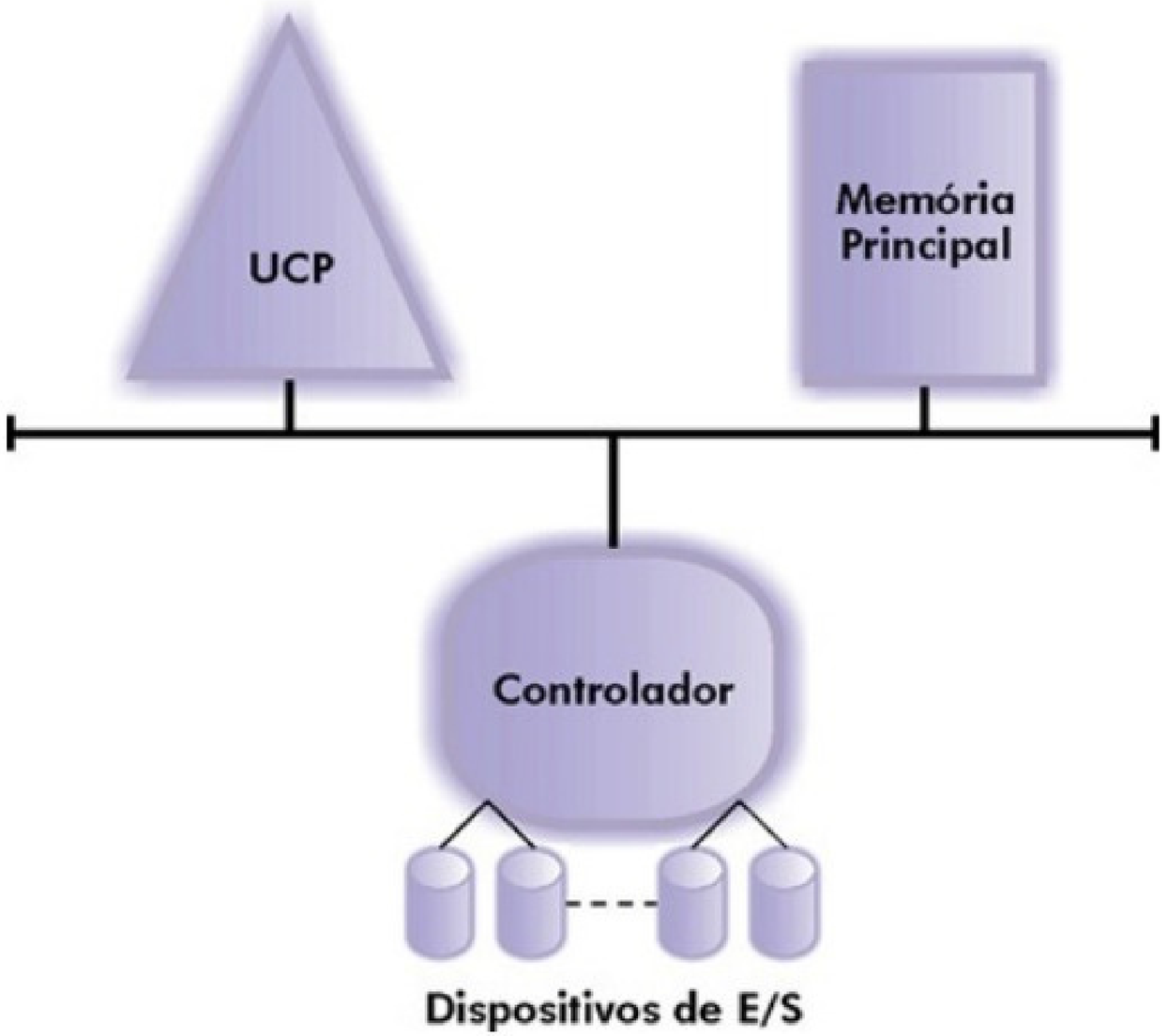

Fig. 3.3 **Controlador.**

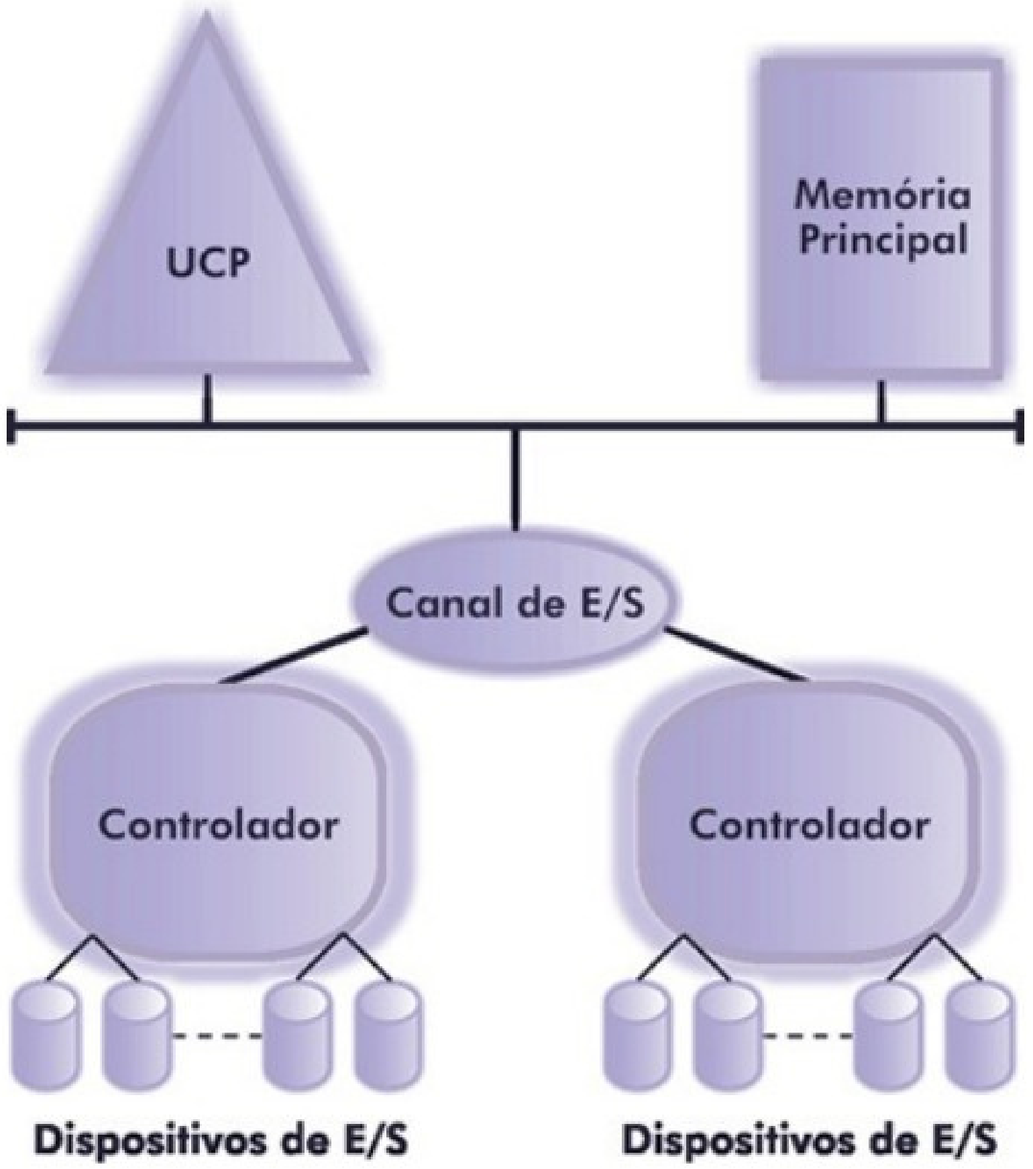

Fig. 3.4 Canal de E/S.

3.5 Buffering

- A técnica de buffering usa uma área de memória principal chamada buffer para transferir eficientemente dados entre dispositivos de E/S e a memória. Em operações de leitura, os dados são inicialmente transferidos para o buffer, permitindo que o dispositivo de entrada realize outra leitura imediatamente. Isso permite que o processador processe dados no buffer enquanto o dispositivo faz outra leitura. A técnica também se aplica a operações de gravação, otimizando as operações de E/S, permitindo operações paralelas e reduzindo o tempo de espera do processador.

- O buffering visa manter o processador e os dispositivos de E/S ocupados na maior parte do tempo.

Buffering

- O mecanismo de buffering usa registros como unidade de transferência. O tamanho do registro pode ser ajustado com base na natureza do dispositivo, como uma linha de impressão ou um caractere de teclado, ou de acordo com as necessidades da aplicação, como um registro lógico em um arquivo.

- O buffer deve ser capaz de armazenar vários registros, permitindo dados não processados (para operações de leitura) ou dados processados, mas não gravados (para operações de gravação). Isso permite que o dispositivo de entrada leia vários registros antes do processamento pelo processador ou que o processador processe vários registros antes da gravação pelo dispositivo de saída. Essa abordagem eficiente ajuda a lidar com a diferença de tempo entre a execução de instruções pelo processador e as operações de leitura e gravação realizadas pelo dispositivo de E/S.

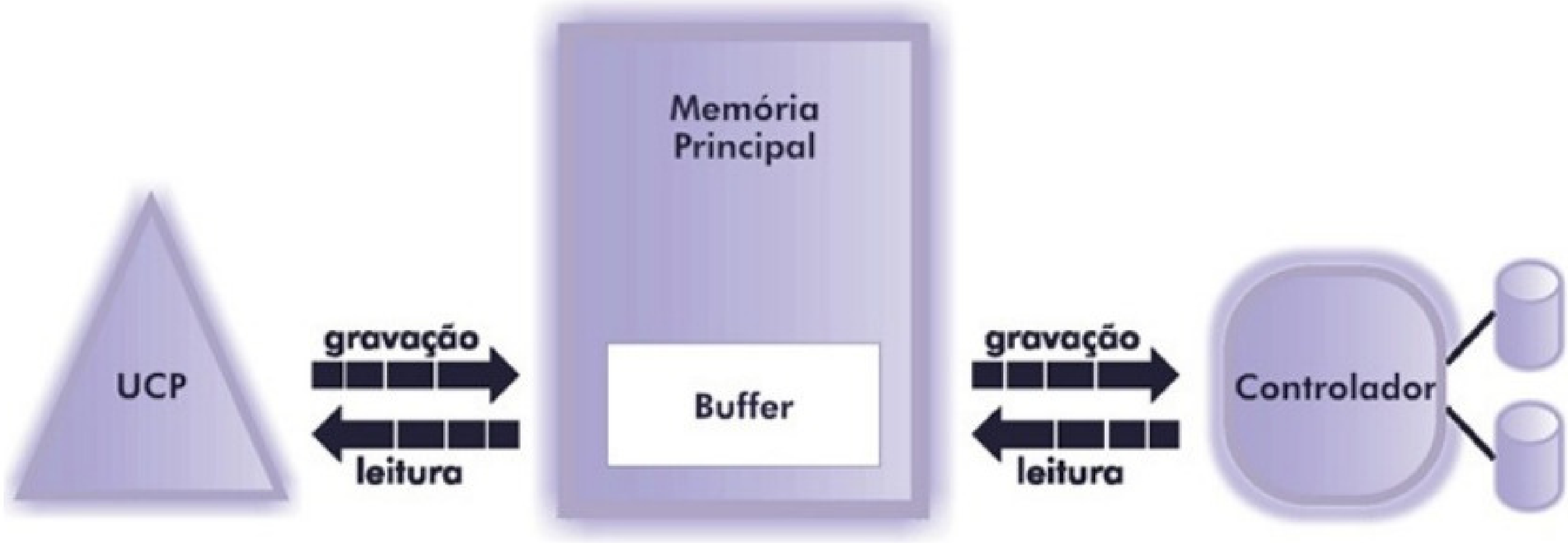

Fig. 3.5 Operações de E/S utilizando buffer.

3.6 Spooling

- No final da década de 50, foi introduzida a técnica Spooling (Simultaneous Peripheral Operation On-Line), com o objetivo de permitir a gestão eficiente de tarefas de entrada/saída, como impressão ou transferência de arquivos, armazenando temporariamente os dados em uma fita magnética antes de serem processados ou enviados para o dispositivo de destino. Isso ajuda a evitar a espera ociosa do processador enquanto aguarda a conclusão de operações de E/S lentas, permitindo que o sistema continue executando outras tarefas.

- Com o surgimento de dispositivos de acesso direto, como discos, foi possível tornar o spooling mais eficiente e, principalmente, possibilitar o processamento não sequencial dos jobs (programas e seus dados).

Spooling

- Atualmente, essa técnica está presente na maioria dos sistemas operacionais, sendo utilizada no gerenciamento de impressão.

- Quando um comando de impressão é executado, as informações a serem impressas são primeiro gravadas em um arquivo de spool (um arquivo em disco), permitindo que o programa continue executando outras tarefas. Posteriormente, o arquivo spool envia o conteúdo armazenado para a impressora.

- O uso do spooling tem a vantagem de evitar que um programa monopolize a impressora para seu uso exclusivo. O sistema operacional assume a responsabilidade de gerenciar a ordem das solicitações de impressão feitas pelos programas, seguindo critérios que garantem a segurança e a utilização eficiente das impressoras.

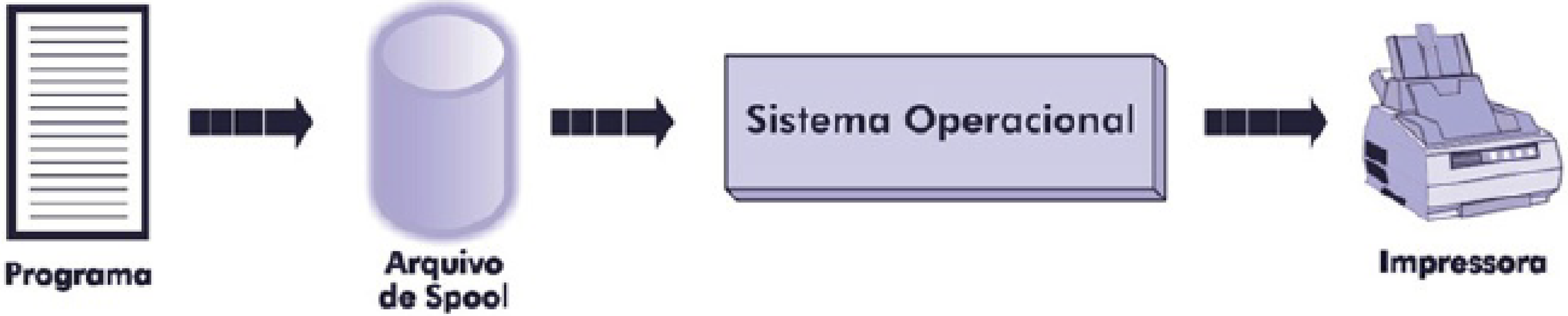

Fig. 3.6 Técnica de spooling.

3.7 Reentrância

- Em sistemas multiprogramáveis, é comum que vários usuários utilizem simultaneamente os mesmos aplicativos. Se cada usuário carregasse o código executável na memória, ocorreria várias cópias do mesmo programa na memória principal, resultando em um desperdício de espaço.

- A Reentrância é a capacidade de um código ser executado simultaneamente por diversos usuários sem causar conflitos ou problemas de concorrência, e exigindo que apenas uma cópia do programa esteja na memória. Isso é importante para garantir a integridade dos dados compartilhados em sistemas multiprogramáveis. Uma rotina reentrante é projetada de forma que possa ser chamada por várias partes do programa ao mesmo tempo, sem que uma chamada interfira nas outras.

- Editores de texto, compiladores e linkers, são exemplos de código reentrante que proporcionam uma utilização mais eficiente da memória principal e aumento no desempenho do sistema.

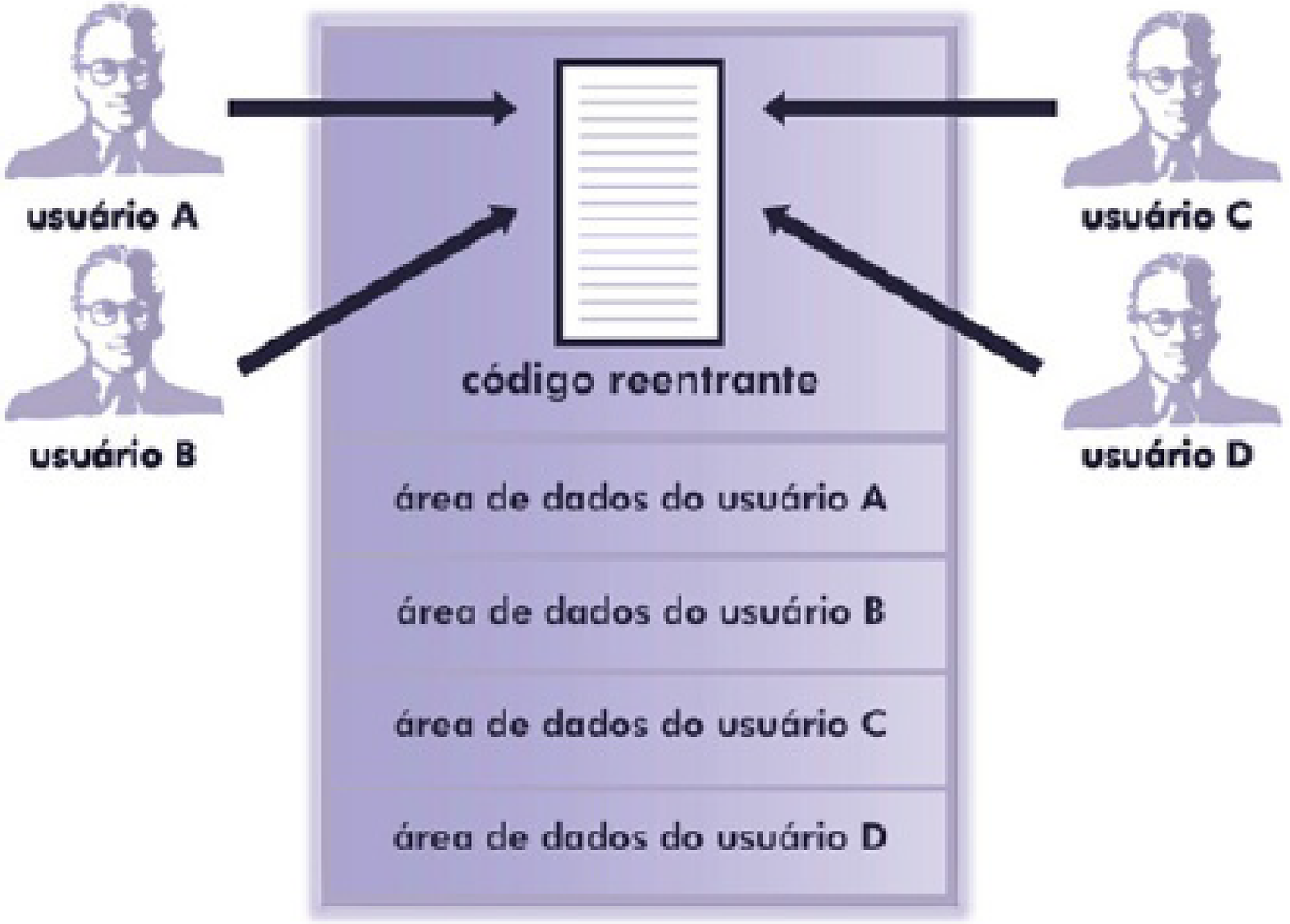

Fig. 3.7 Reentrância.

Conclusão

A concorrência é essencial para maximizar a utilização de recursos em sistemas de computação. Os sistemas monoprogramáveis são simples e adequados para tarefas únicas, enquanto os multiprogramáveis permitem a execução simultânea de várias tarefas. Interrupções e exceções são mecanismos de gerenciamento de eventos, com as exceções sendo acionadas de forma síncrona e as interrupções, de forma assíncrona. Operações de entrada/saída, buffering e spooling são técnicas para otimizar a eficiência da transferência de dados entre dispositivos e programas. A reentrância permite que várias instâncias de um programa compartilhem recursos sem conflitos. No geral, esses conceitos desempenham papéis cruciais na concepção e operação de sistemas operacionais eficientes e multitarefa.

Referência

Livro: Arquitetura de Sistemas Operacionas

Autores: Francis Berenger Machado e Luiz Paulo Maia

Alunos

Emelly Yasmin Andrade Formigário

Gabriélly Custódio Ferreira

Henrique Siviero

Marco Antonio Alcindo Gitti

Fatec

Jales

Prof. José Camargo