

# Entrada e saída

Pedro Cruz

EEL770 – Sistemas Operacionais

# Ante-antepenúltima (???) aula



- Sistemas de arquivos
  - Tipos de arquivos
    - Regulares
    - Diretórios
    - Especiais de caracteres
    - Especiais de blocos
  - Atributos de arquivos
  - Operações com arquivos
  - Diretórios
  - Operações com diretórios
  - Fragmentação interna
  - Fragmentação externa

# Ante-antepenúltima (???) aula



- Implementação de sistemas de arquivos
  - Alocação contígua
  - Alocação por lista encadeada
    - Problema com tamanho do bloco
    - Problema do acesso aleatório
  - Alocação por lista encadeada usando tabela em memória
    - Problema da tabela
  - *I-nodes* (nós-índice)
    - Problema do número de índices por bloco
      - Solução?

- Processos precisam realizar entrada e saída de dados
  - Entrada
    - Variáveis de entrada
    - Eventos de interfaces com usuário
    - Eventos de interfaces com outras máquinas
  - Saída
    - Resultados de cálculos
    - Eventos para interfaces com usuário
    - Eventos para interfaces com outras máquinas

# Dispositivos de entrada e saída

- Orientados a bloco
  - Blocos são endereçáveis
    - Busca é possível
- Orientados a caractere
  - Caracteres são um fluxo
    - Busca é impossível



# Controladores de dispositivo



- Peça de hardware
  - Converte sinais entre processador e dispositivo
    - Padrões definidos
    - Sinais esperados pelos dispositivos
  - Pode ser interno ou externo
    - Placa mãe
    - Integrado ao dispositivo

# Comunicação com dispositivos



- Dispositivos possuem registradores e buffers
- Comunicação através da leitura e escrita
  - Espaço de portas E/S
  - Entrada e saída mapeadas na memória

- Instruções especiais para leitura e escrita de portas E/S
  - Proibidas do modo usuário

IN REG, PORT

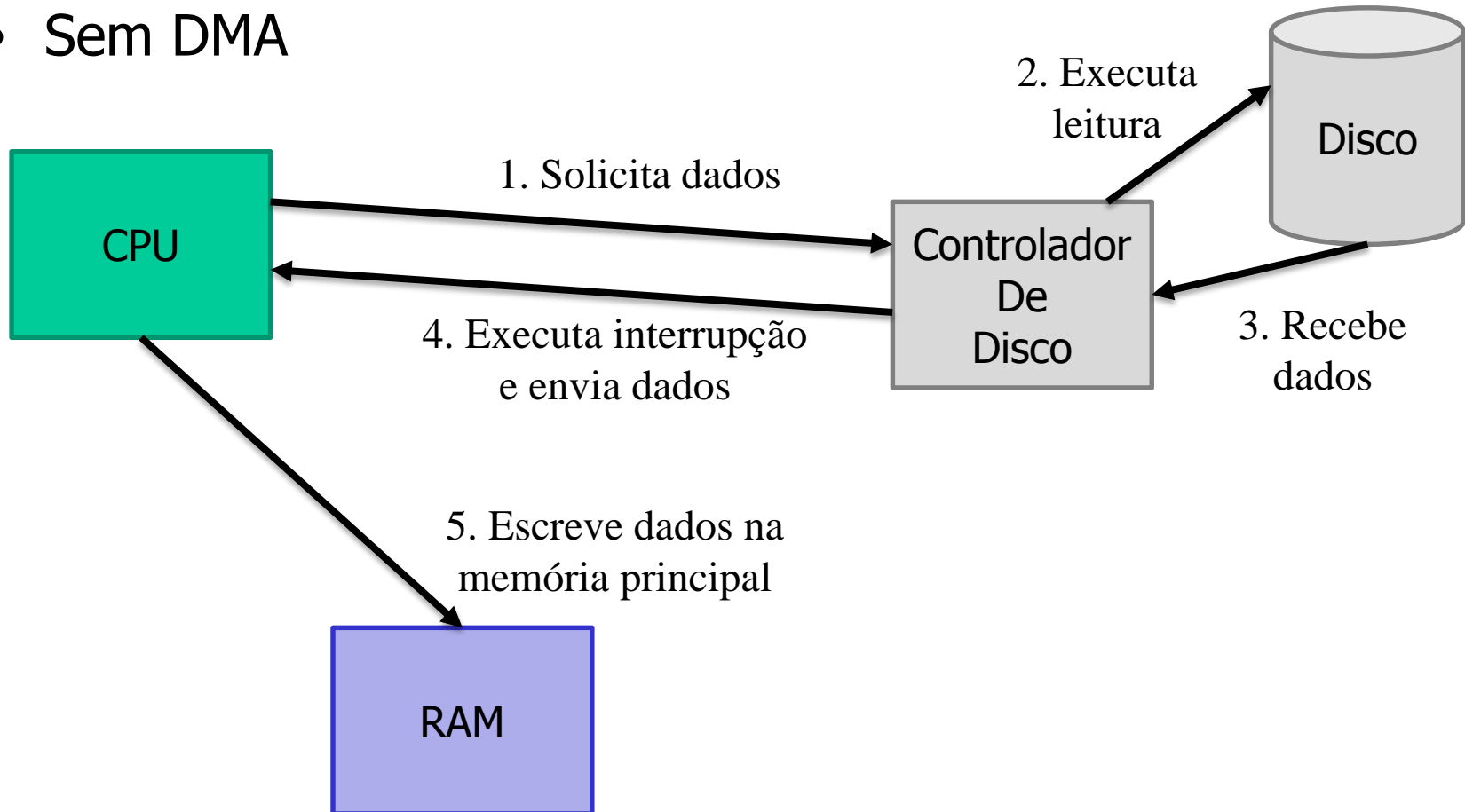
OUT PORT, REG



- Registradores de dispositivos são mapeados em memória
  - Endereços especiais correspondem a dispositivos
  - Busca por endereço de registrador encontra dispositivo
    - Barramento “entende” que deve buscar um dispositivo
  - Vantagens:
    - Instruções especiais não são usadas
    - Dispositivos são protegidos com mecanismos da memória
    - Referenciar a memória é referenciar dispositivos
- TEST PORT\_3

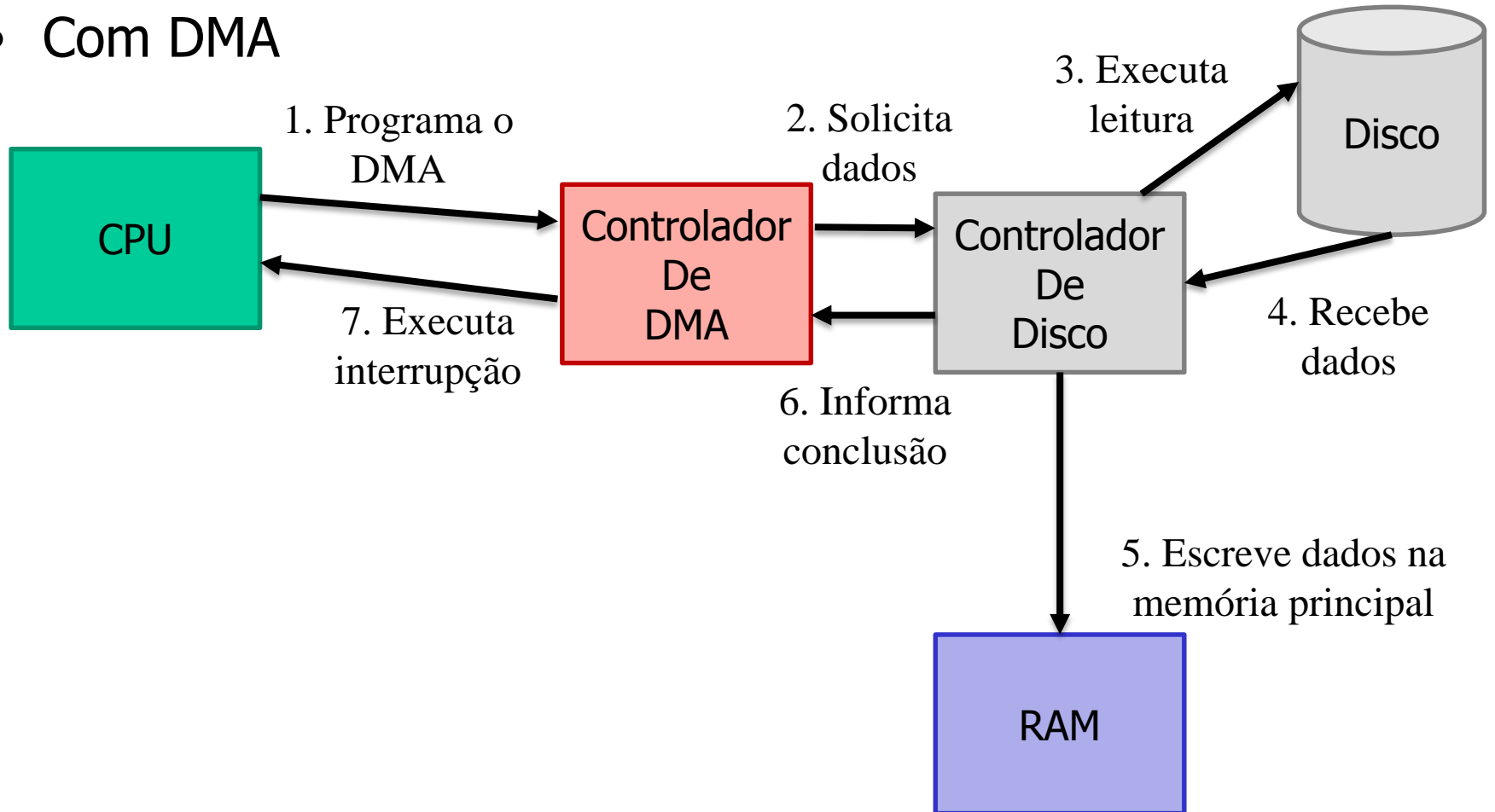
# Acesso Direto à Memória (*Direct Memory Access* - DMA)

- Sem DMA



# Acesso Direto à Memória (*Direct Memory Access - DMA*)

- Com DMA



# Problemas com DMA



- Compete com a CPU pelo barramento
  - DMA geralmente tem preferência
    - Dispositivos são lentos
- Precisa de endereços físicos da memória
  - CPU solicita já com endereço físico
  - DMA tem acesso à MMU
- Adiciona custo ao projeto
  - Se a CPU consegue realizar bem o trabalho, custo inútil
    - *No pain, no pain*

1. Controlador de interrupções recebe sinal de interrupção
  1. Se necessário, há fila de interrupções
2. Controlador repassa interrupção à CPU
3. CPU armazena seu estado atual
  1. Coloca registradores em uma pilha
4. CPU busca rotina no **vetor de interrupções**
5. CPU executa rotina de tratamento da interrupção
6. CPU retorna à execução que foi interrompida

# Interrupções (im)precisas



- CPUs podem ter paralelismo interno
  - Instruções não estão totalmente completas simultaneamente
- Interrupção precisa
  - *Program counter* é salvo em lugar conhecido
  - Instruções antes do PC estão completas
  - Instruções após PC não estão completas
  - Instrução apontada pelo PC tem estado conhecido
- Interrupção imprecisa
  - Não é precisa

# Interrupções imprecisas - solução



- “Estado” salvo é maior do que o das interrupções precisas
- Reiniciar o código requer iniciar diversos registradores
- Retrocompatibilidade
  - Interrupção aguarda término de instruções iniciadas

# Software para Entrada e saída (E/S)



- Objetivos
  - Independência de dispositivo
  - Tratamento de erros
- Arquitetura
  - Camadas



- E/S programada
- E/S orientada a interrupções
- E/S utilizando DMA (*Direct Memory Access*)

- Processo coloca memória à disposição do operacional
  - Cheia se for saída
  - Vazia se for entrada
- Processo de usuário realiza chamada de sistema
  - Bloqueante
- Operacional entrega/recebe dados de dispositivo
  1. Entrega uma palavra
  2. Espera dispositivo ficar pronto
  3. Volta a 1

# E/S orientada a interrupções



- Processo coloca memória à disposição do operacional
  - Cheia se for saída
  - Vazia se for entrada
- Processo de usuário realiza chamada de sistema
  - Bloqueante para o processo
- Operacional entrega/recebe dados de dispositivo
  1. Entrega uma palavra
  2. Executa outro processo até ser chamado por interrupção
  3. Volta a 1

# E/S utilizando DMA



- Processo coloca memória à disposição do operacional
- Processo de usuário realiza chamada de sistema
  - Bloqueante para o processo
- Operacional configura controlador de DMA
- Operacional escalona outros processos
- Controlador de DMA entrega dados ao dispositivo
  1. Entrega uma palavra
  2. Espera dispositivo ficar pronto
  3. Volta a 1
- Controlador faz interrupção e acorda processo solicitante

# Camadas de software de E/S



Processos de usuário

Solicita E/S, formata E/S e coloca na fila

Software independente de dispositivo

Nomeia, protege, bloqueia, define buffer e aloca

Drivers de dispositivos

Configura dispositivo e registradores de dispositivo

Tratadores de interrupção

Acorda driver quando E/S está completa

Hardware

Executa operação de E/S

# Tratadores de interrupção



- Salvar registradores
- Estabelecer contexto para a rotina da interrupção
- Sinalizar o controlador de interrupções
- Executar a rotina de tratamento de interrupção
- Escolher o próximo processo
- Estabelecer o contexto para o próximo processo
- Iniciar a execução do próximo processo

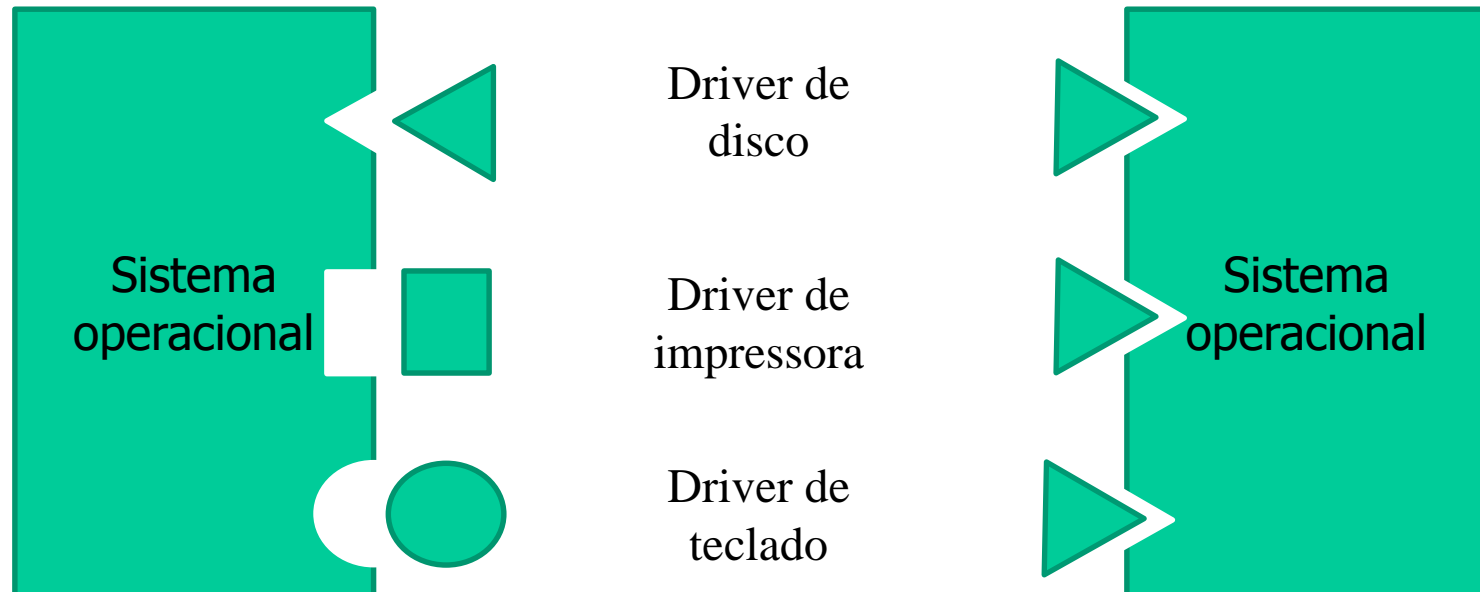
# Drivers de dispositivos



- Lidam com as peculiaridades de cada dispositivo
  - Ou de classes de dispositivos
- Podem ser empilhados
  - Dispositivos que possuem tecnologias diferentes encadeadas
    - Exemplo: dispositivos USB
- Ficam no espaço do núcleo
- Falam com o *hardware*
  - Controladores de dispositivos
- Devem lidar com reentrância
  - E/S chega antes do processamento da E/S anterior
- Devem lidar com desligamento súbito

# Software de E/S independente de dispositivo

- Código deve ser sempre geral
  - Manutenibilidade
  - Curva de aprendizado dos desenvolvedores
- Interface uniformizada para classes de dispositivos
  - Operacional define operações que driver deve oferecer





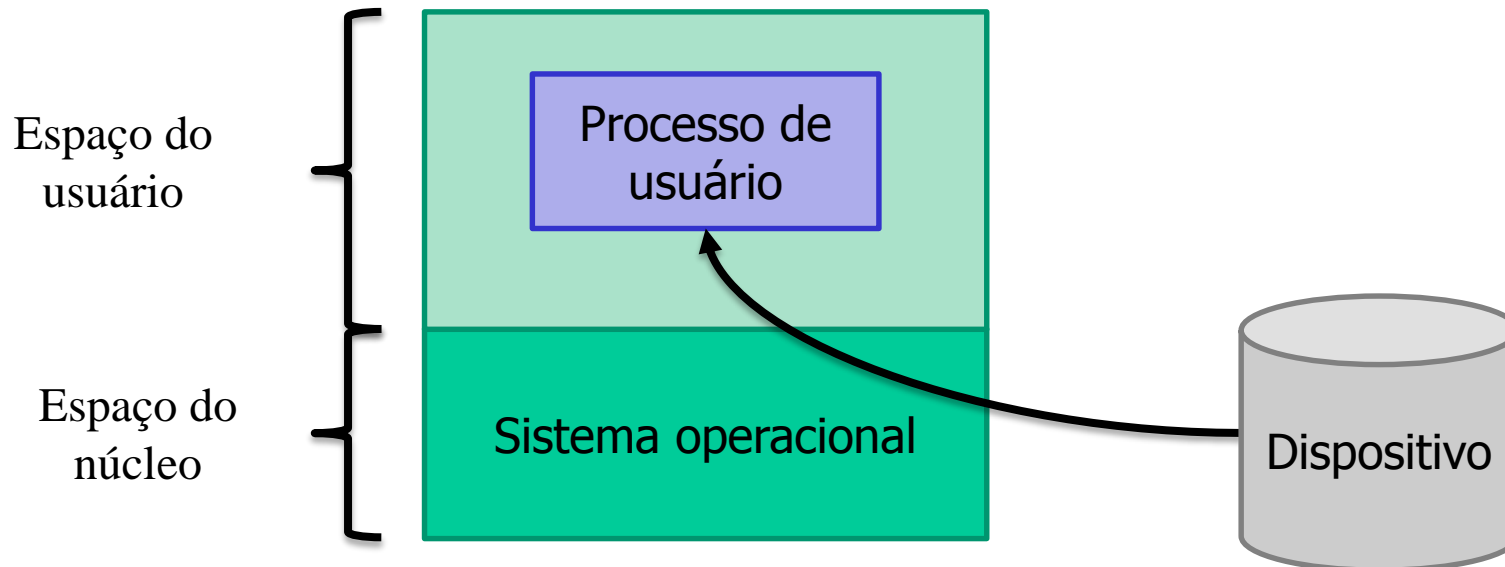
# Buffers de dados



- Troca de dados
  - Sem buffer
  - Com buffer em espaço de usuário
  - Com buffer no núcleo
  - Com buffer duplo no núcleo

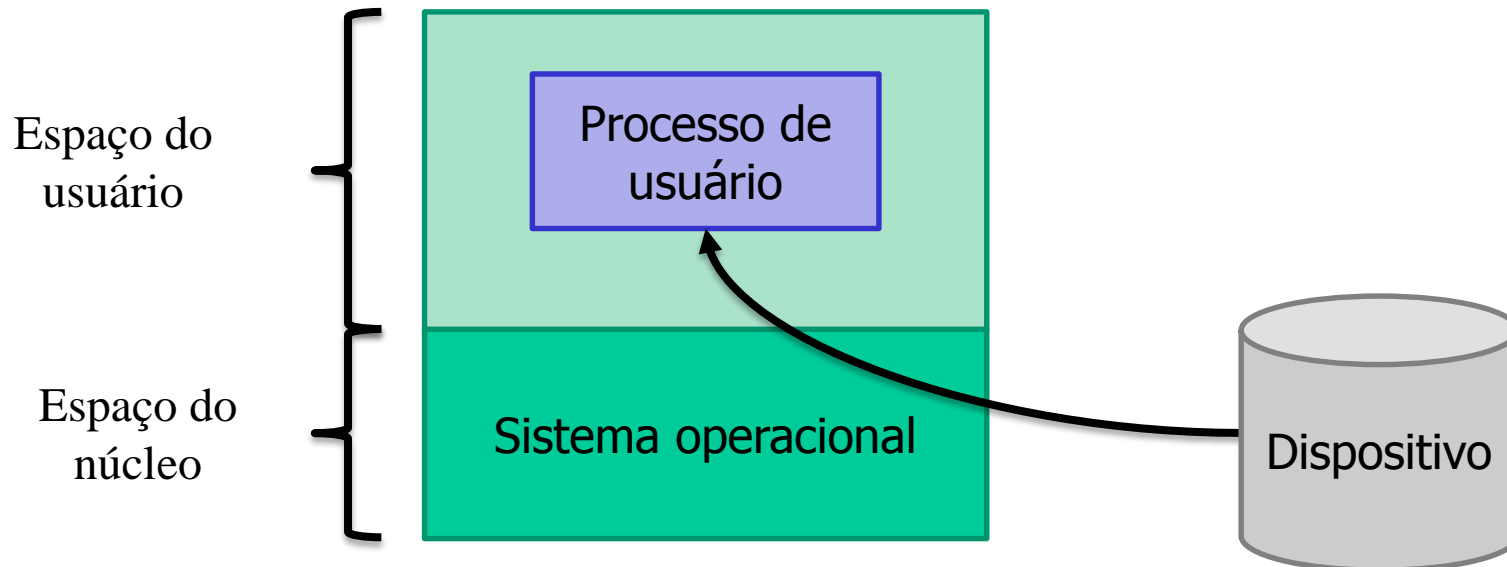
# Troca sem buffer

- Processo faz chamada de sistema e aguarda interrupção
  - Lê/escreve um caractere/bloco



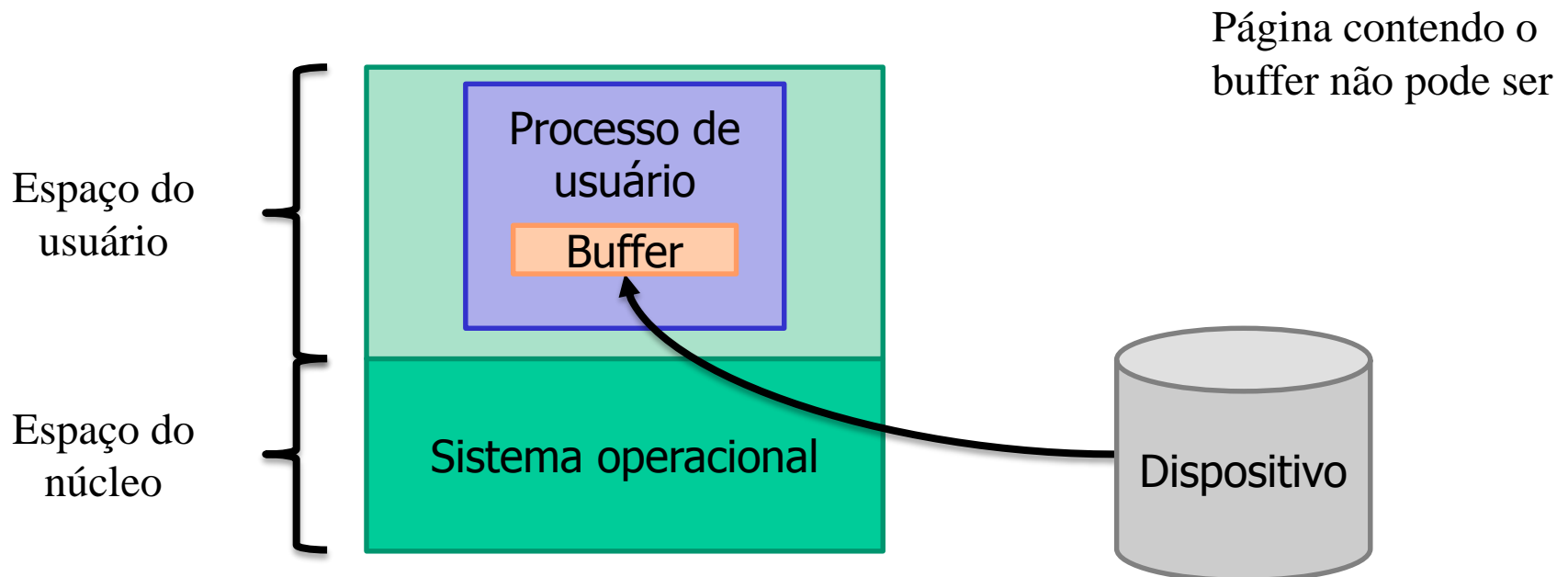
# Troca sem buffer - problemas

- Uma interrupção por caractere/bloco



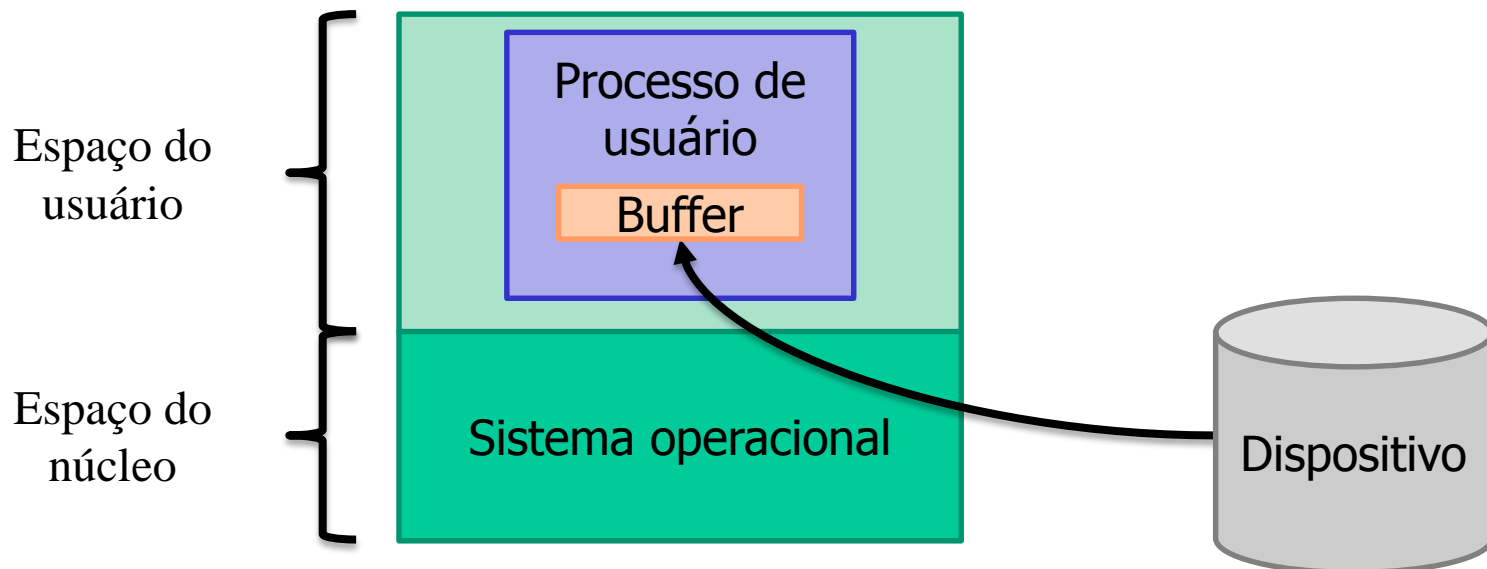
# Troca com buffer em espaço de usuário

- Processo faz chamada de sistema e aguarda interrupção
  - Oferece um buffer
  - Rotina de interrupção Lê/escreve um caractere/bloco
    - Desbloqueia quando completo



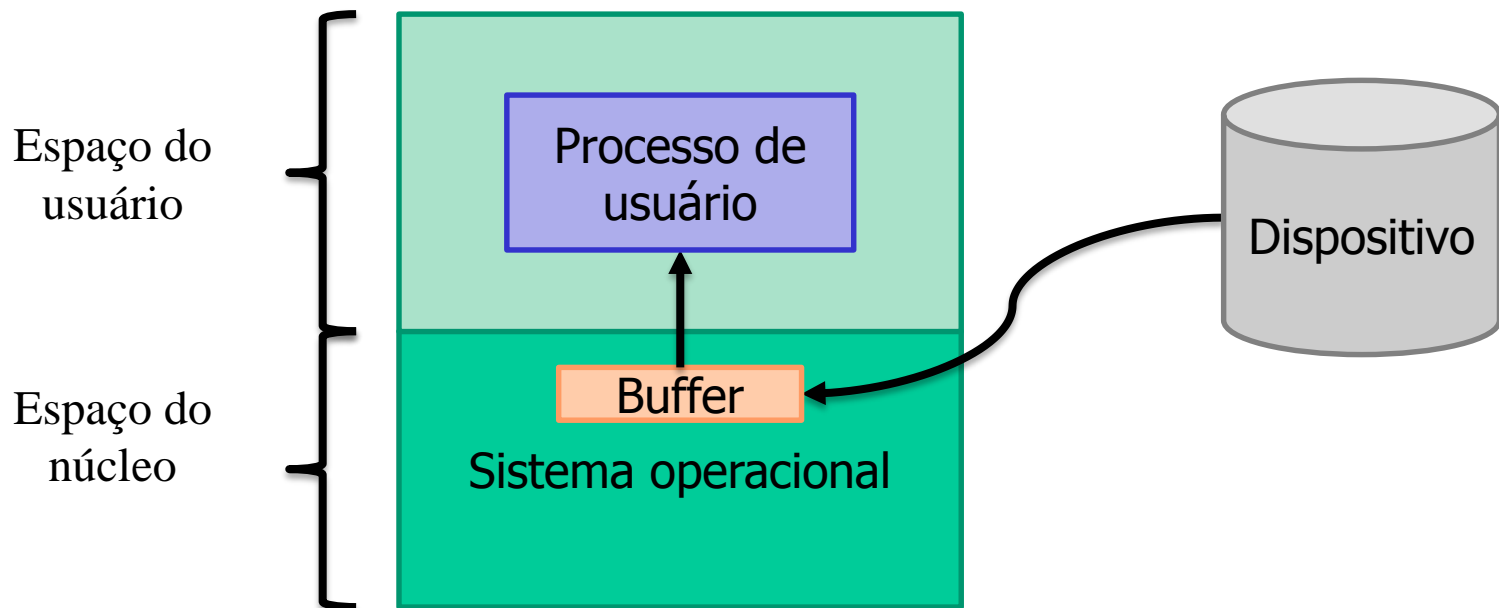
# Troca com buffer em espaço de usuário - problemas

- Página com o buffer deve ficar na memória primária
  - Se muitos processos travarem páginas



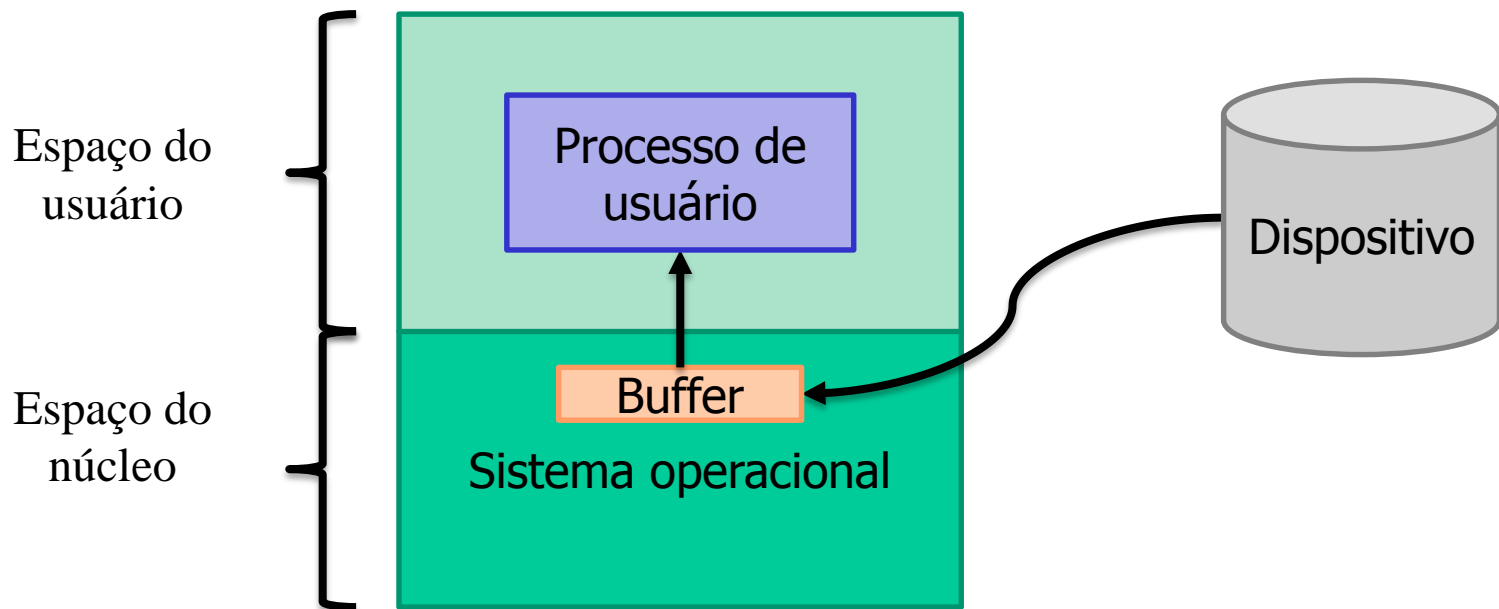
# Troca com buffer no núcleo

- Processo faz chamada de sistema e aguarda interrupção
  - Operacional oferece um buffer
  - Rotina de interrupção enche o buffer
  - Operacional copia o buffer para processo e o desbloqueia



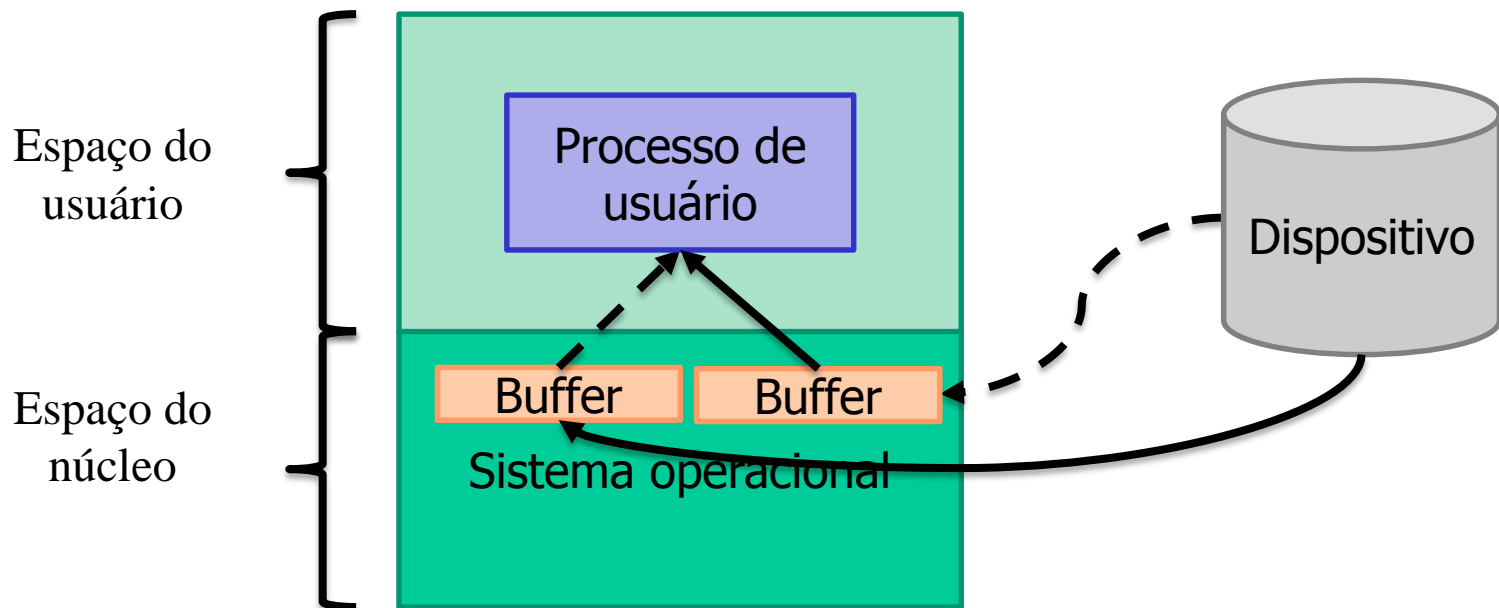
# Troca com buffer no núcleo - problemas

- Buffer cheio
  - Caractere chega antes da cópia estar completa
    - O que fazer?



# Troca com buffer duplo no núcleo

- Processo faz chamada de sistema e aguarda interrupção
  - Operacional oferece um buffer
  - Rotina de interrupção enche o buffer
  - Operacional copia o buffer para processo e o desbloqueia





- Processos não têm acesso direto ao dispositivo de E/S
- *Daemon* controla o acesso ao dispositivo
  - Exemplo: impressora
    - Processos escrevem arquivos em diretório especial
    - *Daemon* de impressão lê arquivos e envia à impressora

# *Redundant Array of Inexpensive Disks - RAID*



- Forma de organizar os arquivos com redundância
  - Sete maneiras diferentes
- Forma de fazer vários discos físicos “parecerem” um só

# RAID Nível 0

- Endereçamento virtual linear
  - Setores distribuídos ao longo dos discos em *round-Robin*
- Operações de disco podem ser paralelizadas

| Disco            | Disco 1 | Disco 2 | Disco 3 |
|------------------|---------|---------|---------|
| Endereço virtual | 0       | 1       | 2       |
|                  | 3       | 4       | 5       |
|                  | 6       | 7       | 8       |
|                  | 9       | 10      | 11      |

# Entrada e saída

Pedro Cruz

EEL770 – Sistemas Operacionais