



Entrada e Saída e Suporte do SO



Entrada e Saída

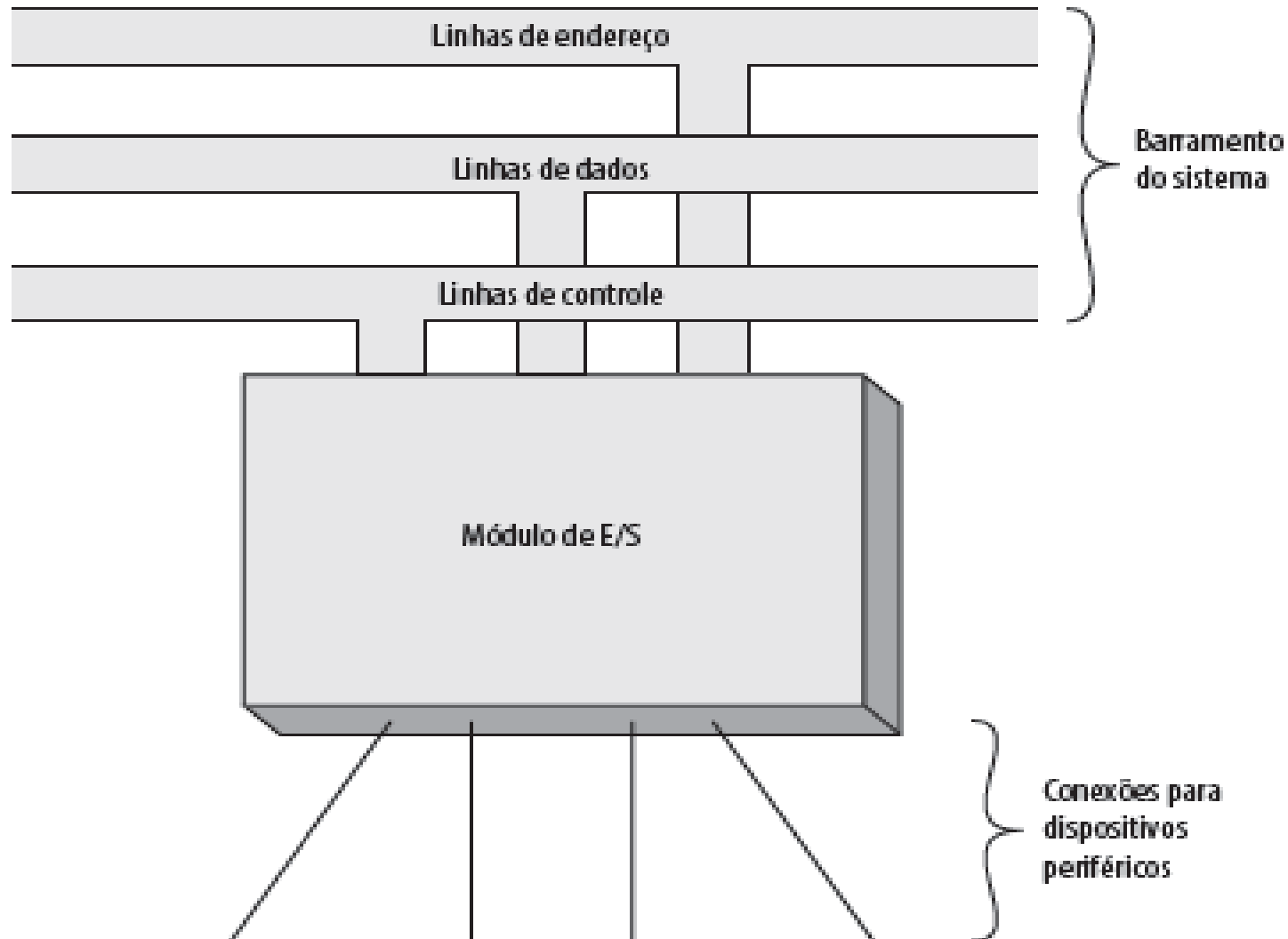


- Grande variedade de periféricos:
 - Entregando diferentes quantidades de dados.
 - Em velocidades diferentes.
 - Em formatos diferentes.
- Todos mais lentos que CPU e RAM.
- Precisa de módulos de E/S.



- Interface com CPU e memória.
- Interface com um ou mais periféricos.

Modelo genérico de módulo de E/S



Fonte: Arquitetura e Organização de Computadores – William Stallings

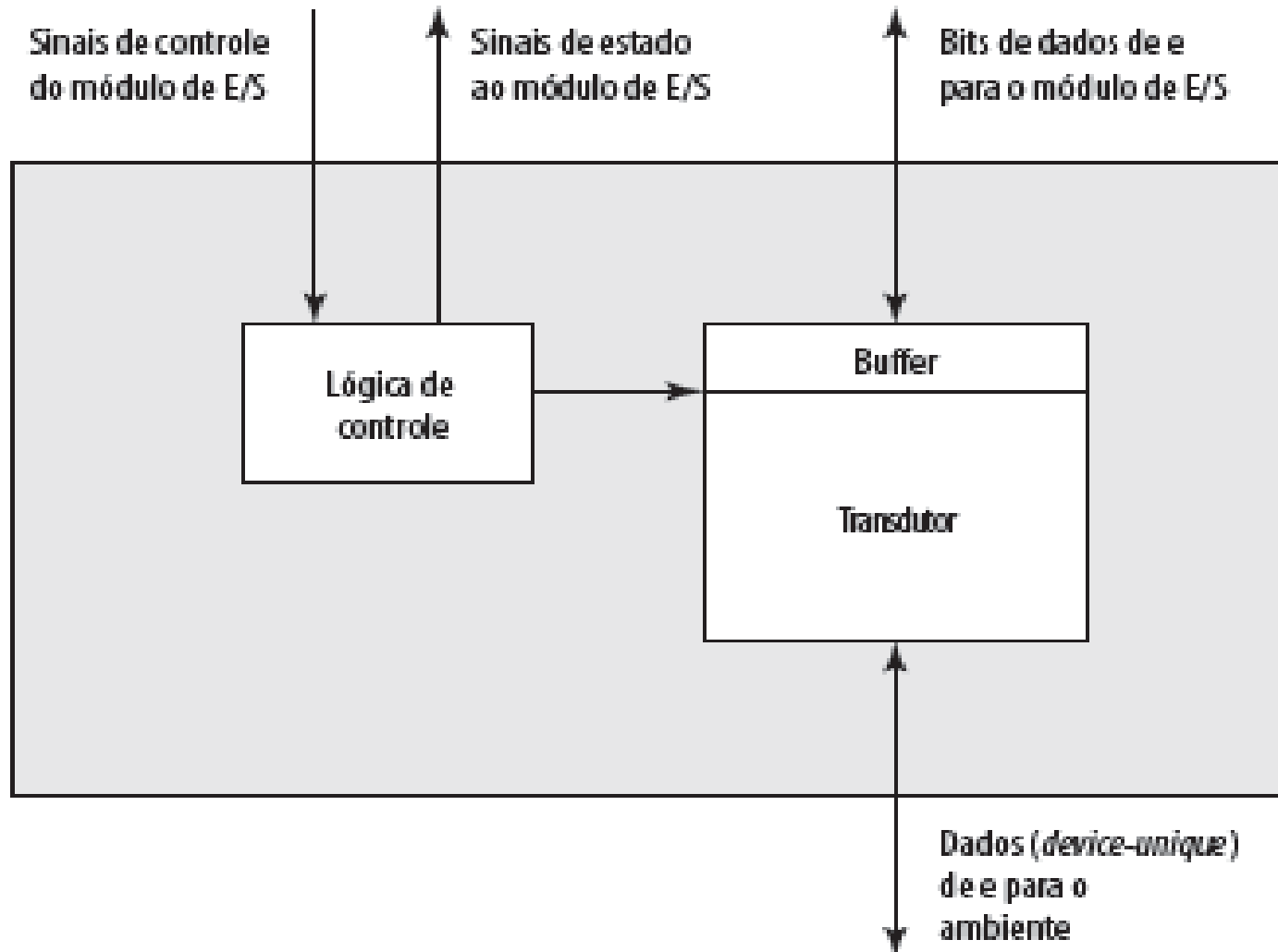
Dispositivos externos



Quem usa os módulos de E/S:

- Legíveis ao ser humano:
 - Monitor, impressora, teclado.
- Legíveis à máquina:
 - Monitoração e controle.
- Comunicação:
 - Modem.
 - Placa de interface de rede.

Diagrama em blocos de um dispositivo externo



Fonte: Arquitetura e Organização de Computadores – William Stallings

Funções do módulo de E/S



- Controle e temporização.
- Comunicação com CPU.
- Comunicação com dispositivo.
- Buffering de dados.
- Detecção de erro.

Etapas da E/S



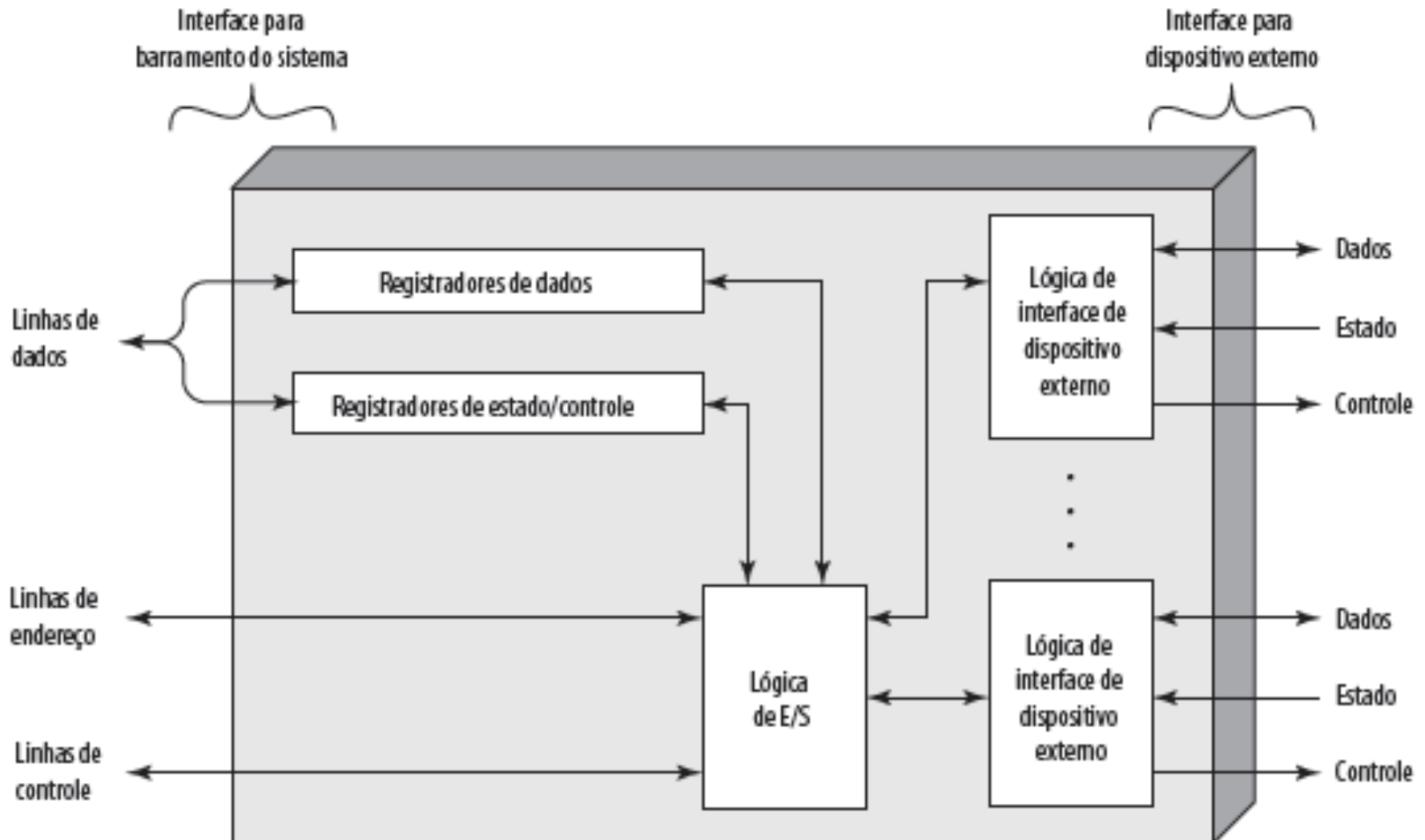
- CPU verifica estado do dispositivo do módulo de E/S.
- Módulo de E/S retorna o estado.
- Se estiver pronto, CPU solicita transferência de dados.

Etapas da E/S



- Módulo de E/S recebe dados do dispositivo.
- Módulo de E/S transfere dados à CPU.
- Variações para saída, DMA etc.

Diagrama do módulo de E/S



Fonte: Arquitetura e Organização de Computadores – William Stallings



- Ocultar ou revelar propriedades do dispositivo à CPU.
- Admitir dispositivo múltiplo ou único.
- Controlar funções do dispositivo ou sair para CPU.



- Programada.
- Por interrupção.
- Acesso direto à memória (DMA).



- CPU tem controle direto sobre E/S:
 - Conhecendo o estado.
 - Comandos de leitura/escrita.
 - Transferindo dados.
- CPU **espera** que módulo de E/S termine a operação.
- **Desperdiça** tempo de CPU.

E/S programada – detalhamento



- CPU solicita operação de E/S.
- Módulo de E/S realiza operação.
- Módulo de E/S define bits de estado.

E/S programada – detalhamento



- CPU verifica bits de estado periodicamente.
- Módulo de E/S não informa à CPU diretamente.
- Módulo de E/S não interrompe CPU.
- CPU deve esperar e acessar posteriormente.

E/S controlada por interrupção



- **Contorna problema de espera** da CPU.
- Sem verificação de dispositivo repetida da CPU.
- Módulo de E/S interrompe quando estiver pronto.

E/S controlada por interrupção – Operação básica



- CPU emite comando de leitura.
- Módulo de E/S recebe dados do periférico enquanto CPU faz outro trabalho.
- Módulo de E/S interrompe CPU.
- CPU solicita dados.
- Módulo de E/S transfere dados.

Ótica da CPU para a Interrupção



- Emite comando de leitura.
- Realiza outro trabalho.
- Verifica interrupção ao final de cada ciclo de instrução.

Ótica da CPU para a Interrupção



- Se interrompida:
 - Salva contexto (registradores).
 - Processa interrupção.
 - Busca dados e armazena.

Aspectos de projeto da Interrupção



- Como identificar o módulo que emite a interrupção?
- Como lidar com interrupções múltiplas?
 - Ou seja, um tratador de interrupção sendo interrompido.

Formas de Identificar o módulo que interrompe



- Linha diferente para cada módulo;
- Verificação por software:
 - CPU verifica cada módulo por vez.
 - Lento.
- Verificação por hardware;
- Arbitração de barramento (Mais utilizado):
 - Módulo deve reivindicar o barramento antes que possa causar uma interrupção.

Múltiplas interrupções – Pode haver



- Cada linha de interrupção tem uma **prioridade**.
- Linhas com prioridade mais alta podem interromper linhas com prioridade mais baixa.
- Com *bus mastering*, só o mestre atual pode interromper.

Acesso direto à memória (DMA)



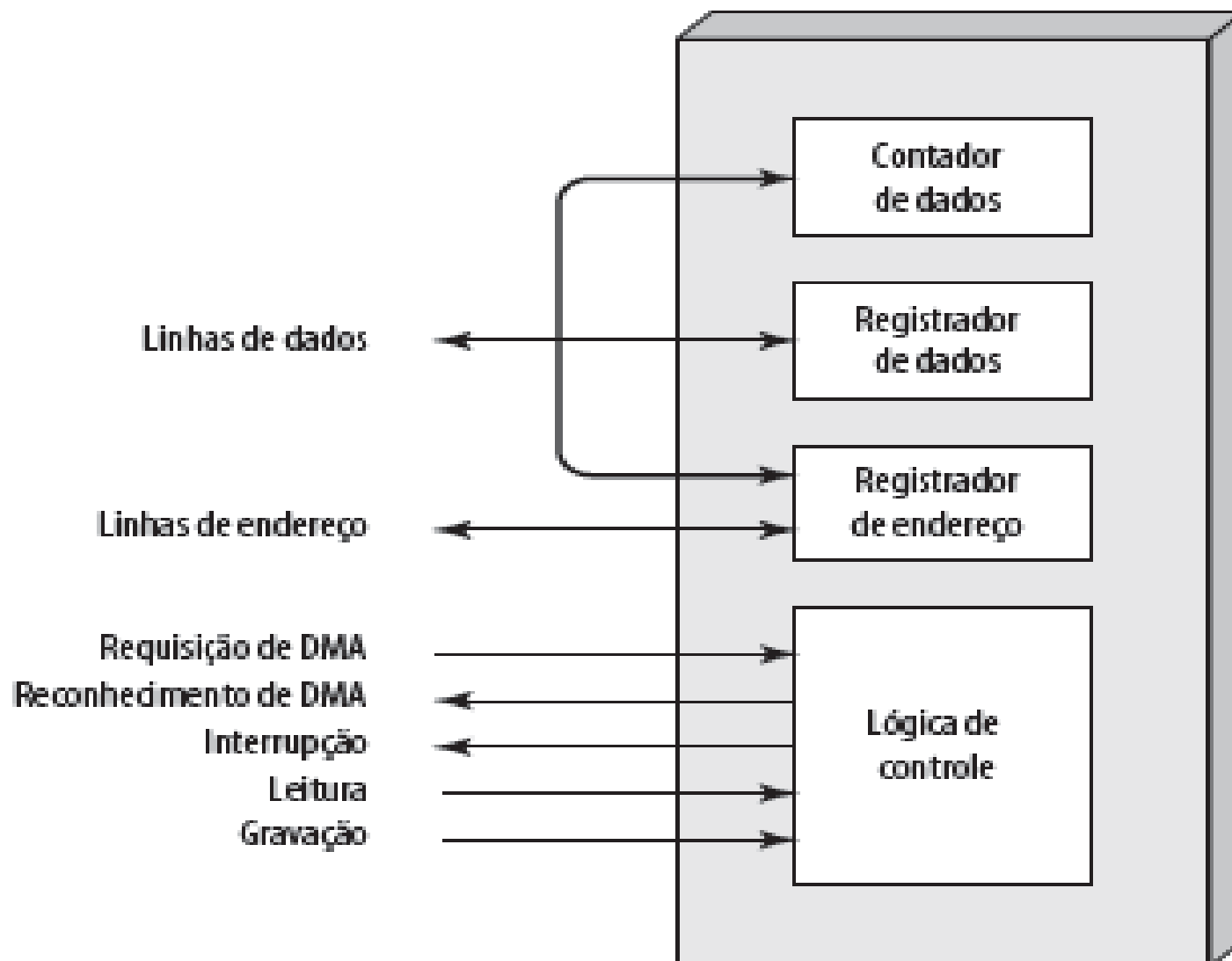
- E/S controlada por interrupção e programada exige intervenção ativa da CPU.
 - Taxa de transferência é limitada.
 - CPU fica restrita.
- DMA é a **solução**.

Função do DMA



- Módulo adicional (hardware) no barramento.
- Controlador de DMA toma o comando da CPU para E/S.

Diagrama típico do módulo de DMA



Fonte: Arquitetura e Organização de Computadores – William Stallings

Operação do DMA



- CPU diz ao controlador de DMA:
 - Leitura/escrita.
 - Endereço do dispositivo.
 - Endereço inicial do bloco de memória para dados.
 - Quantidade de dados a serem transferidos.

Operação do DMA



- CPU prossegue com outro trabalho.
- Controlador de DMA lida com transferência.
- Controlador de DMA envia interrupção quando terminar.

Transferência de DMA – Roubo de ciclo (*Cycle Stealing*)



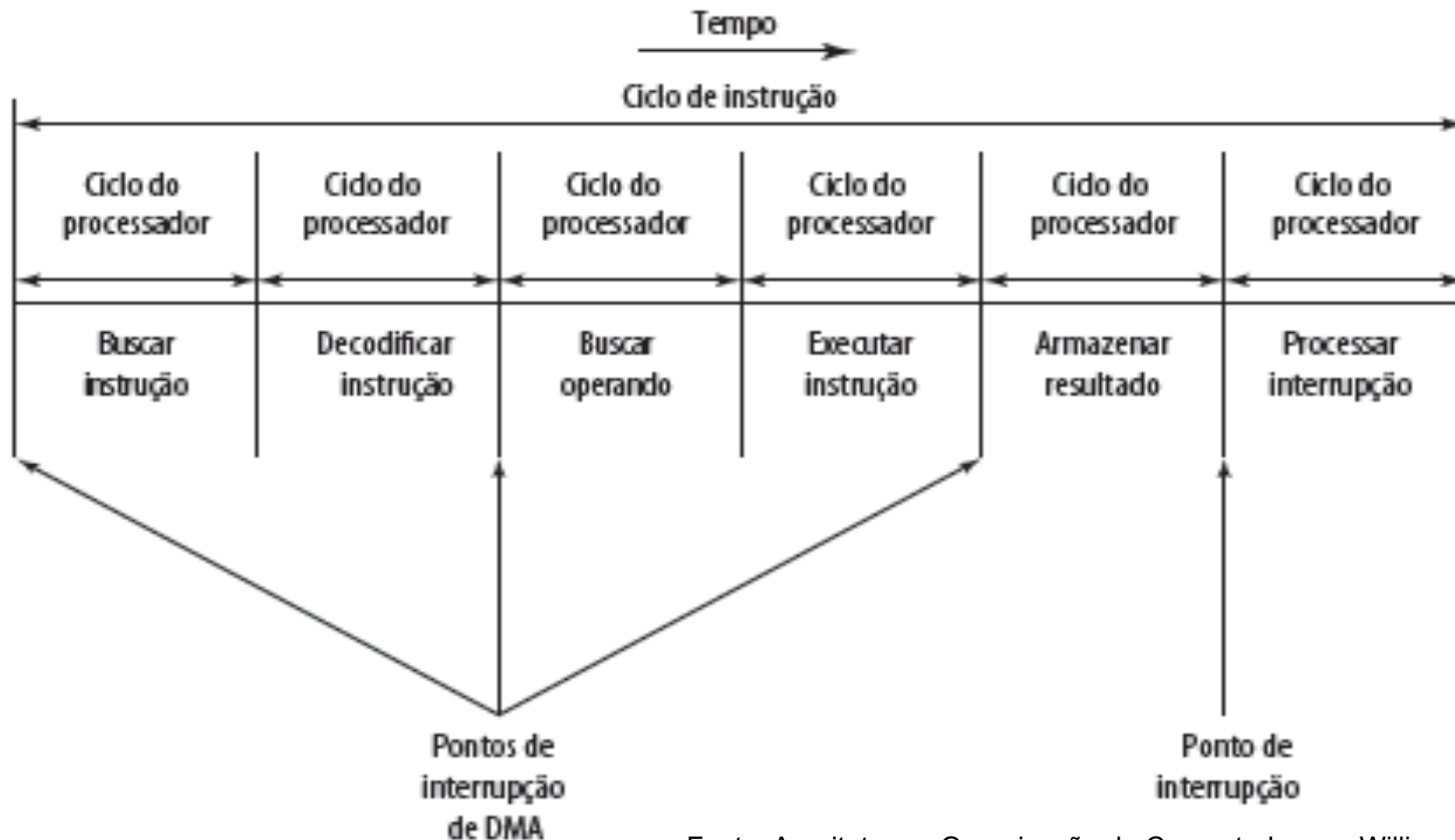
- Controlador de DMA assume o barramento por um ciclo.
- Transferência de uma palavra de dados.
- Não uma interrupção.
 - CPU não troca de contexto.

Transferência de DMA – Roubo de ciclo (*Cycle Stealing*)



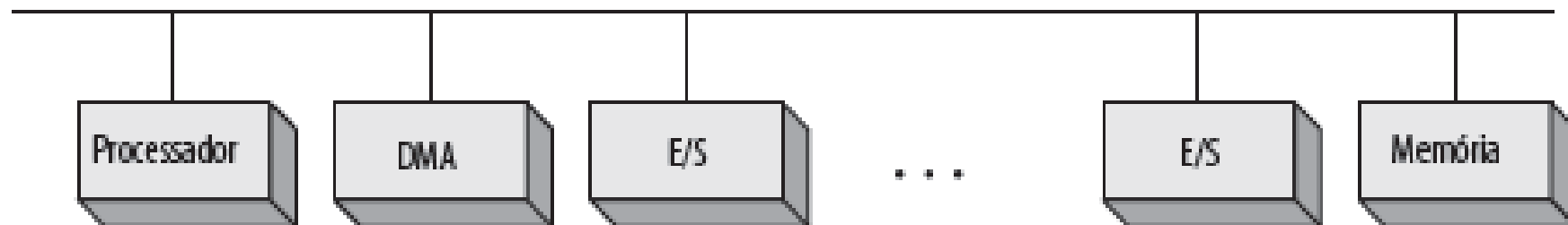
- CPU suspensa logo antes de acessar o barramento.
 - Ou seja, antes de uma busca de operando ou dados ou uma escrita de dados.
- Atrasa a CPU, mas não tanto quanto a CPU fazendo transferência.

DMA e pontos de interrupção durante um ciclo de instrução



Fonte: Arquitetura e Organização de Computadores – William Stallings

Configurações de DMA

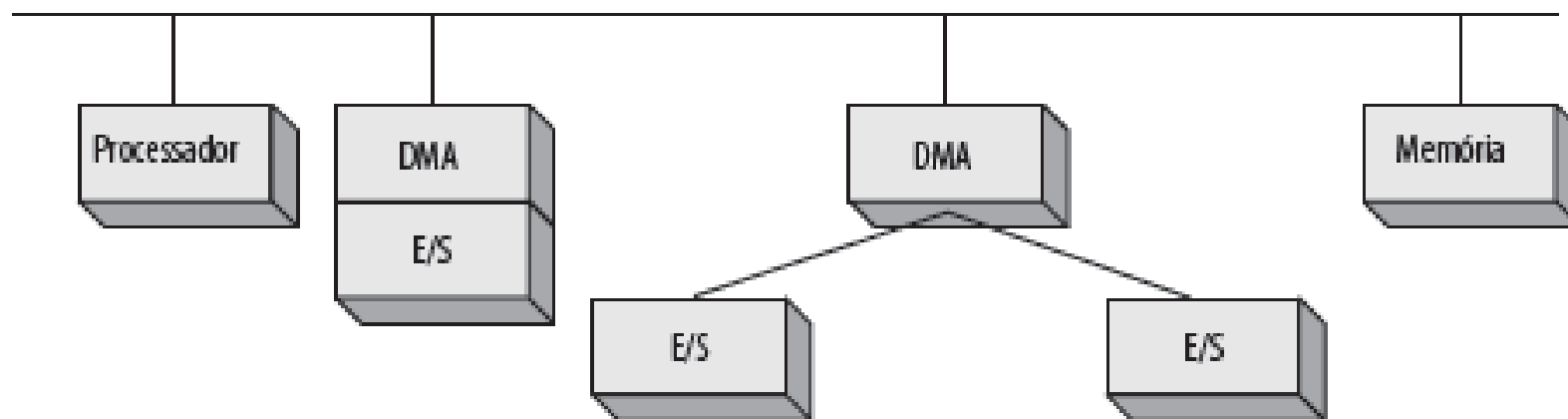


(a) Único barramento, DMA separado

Fonte: Arquitetura e Organização de Computadores – William Stallings

- Único barramento, controle de DMA separado.
- Cada transferência usa barramento duas vezes.
 - E/S para DMA, depois DMA para memória.
- CPU é suspensa duas vezes.

Configurações de DMA

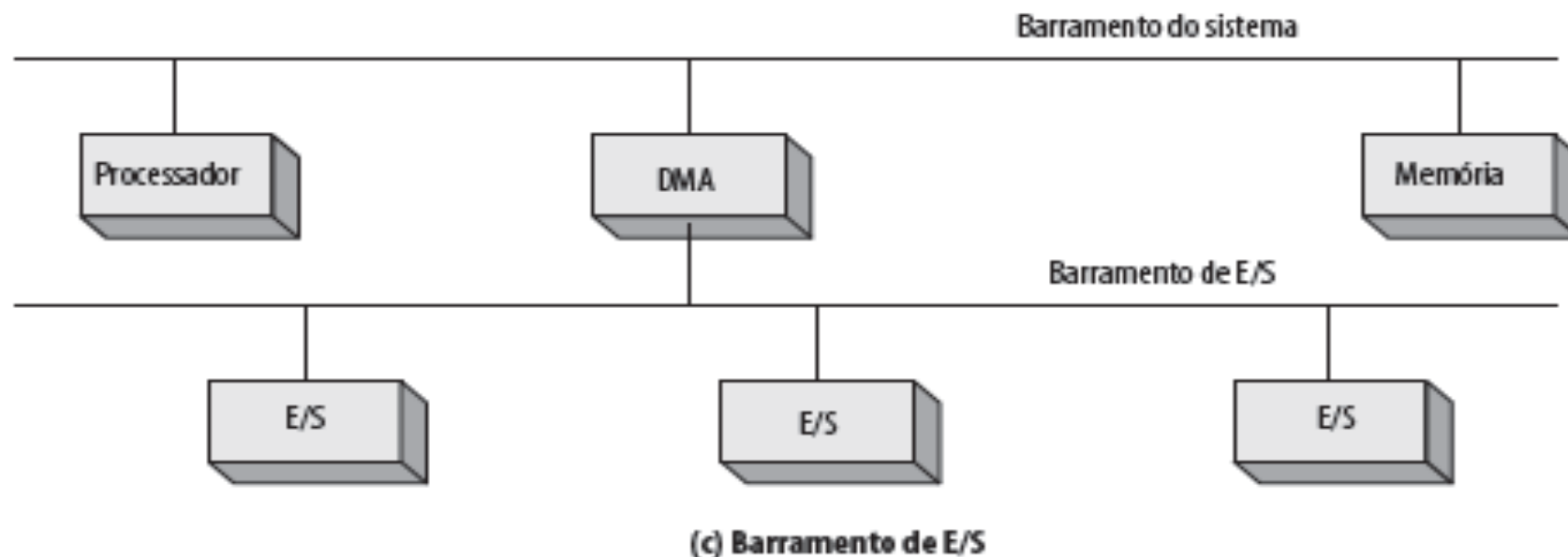


(b) Único barramento, DMA-E/S integrados

Fonte: Arquitetura e Organização de Computadores – William Stallings

- Único barramento, controlador de DMA integrado.
- Controlador pode aceitar mais de um dispositivo.
- Cada transferência usa barramento uma vez.
 - DMA para memória.
- CPU é suspensa uma vez.

Configurações de DMA



Fonte: Arquitetura e Organização de Computadores – William Stallings

- Barramento de E/S separado.
- Barramento aceita todos dispositivos habilitados para DMA.
- Cada transferência usa barramento uma vez.
 - DMA para memória.
- CPU é suspensa uma vez.

Processador fica desocupado



- Enquanto DMA usa barramentos, processador fica ocioso.
- Processador usando barramento, DMA ocioso:
 - Conhecido como **controlador de DMA flutuante**.
- Dados não passam e são armazenados no chip de DMA.
 - DMA apenas entre porta de E/S e memória.
 - Não entre duas portas de E/S ou dois locais de memória.



Suporte do SO

Considerações Iniciais



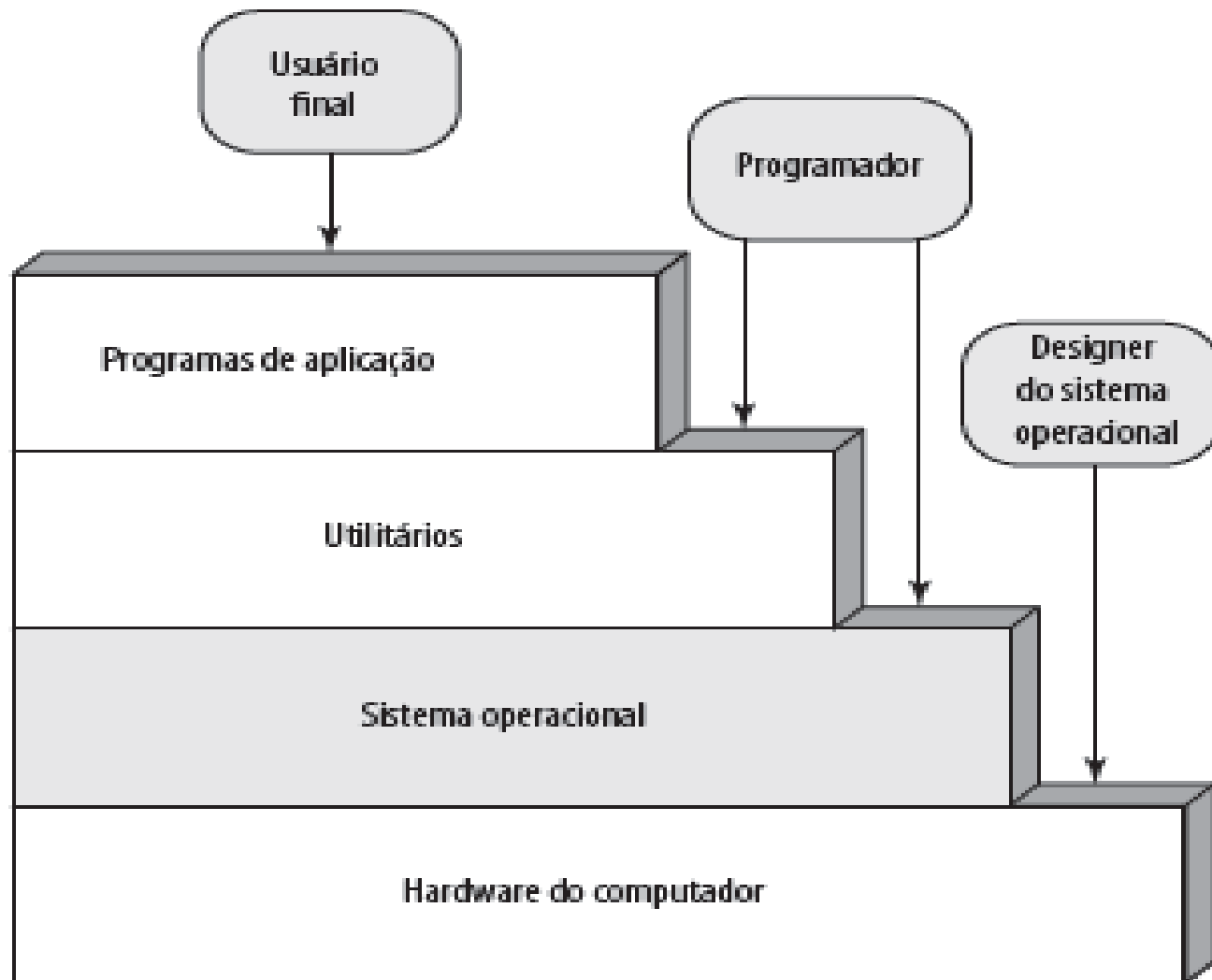
- Verificou-se todo processo de construção de uma sistema computacional;
- Este sistema precisa de uma suporte de software;
- Gestão do conjunto de instruções;
- Assim, o **Sistema Operacional** é fundamental.

Objetivos e funções



- Conveniência:
 - Tornar o sistema computacional mais fácil de usar.
- Eficiência:
 - Permitir o melhor uso dos **recursos** do computador.

Camadas e visões de um sistema de computação



Serviços do sistema operacional



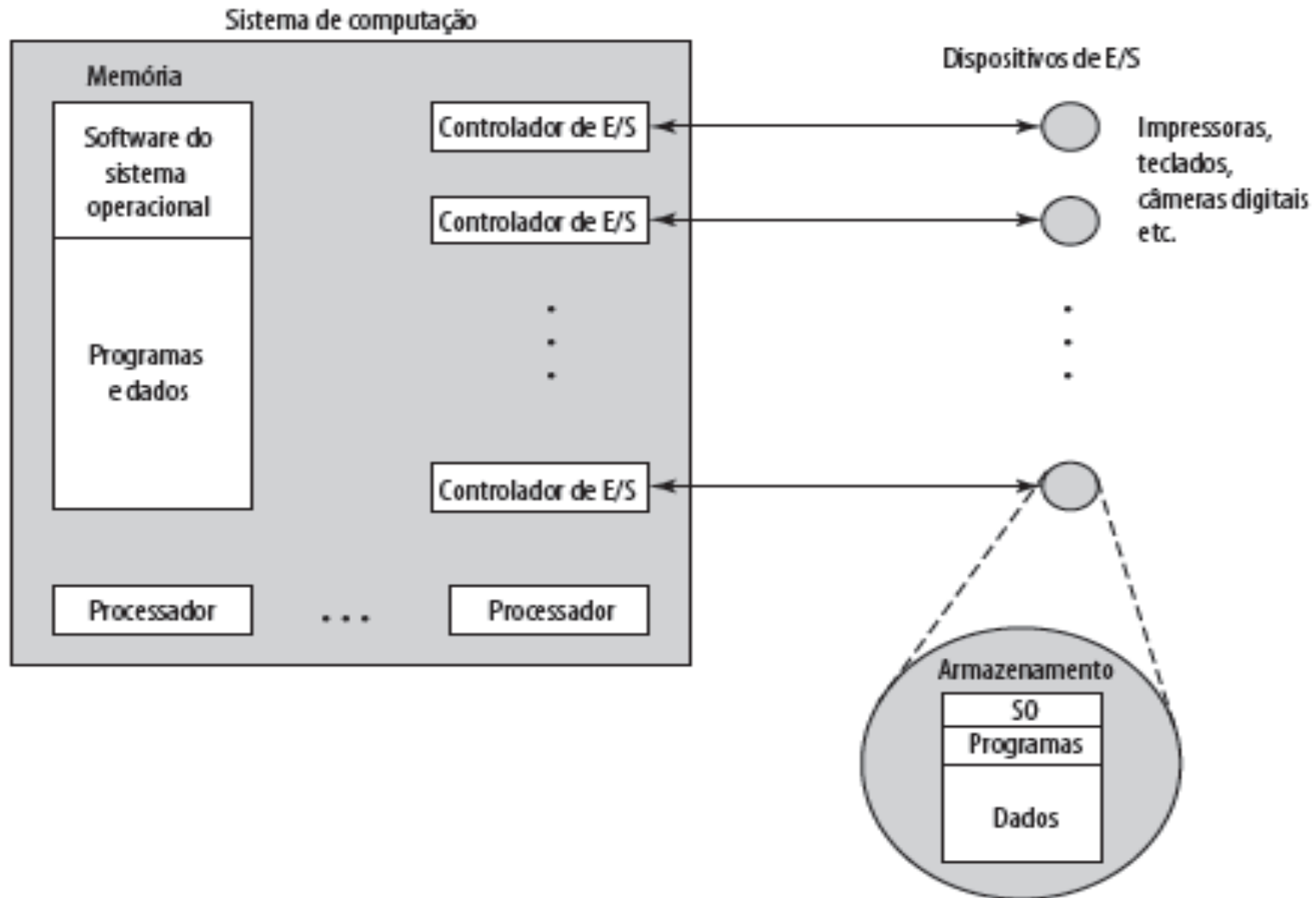
- Criação de programas.
- Execução de programas.
- Acesso aos dispositivos de E/S.

Serviços do sistema operacional



- Acesso controlado aos arquivos.
- Acesso ao sistema.
- Detecção e resposta a erros.

SO como gerenciador de recursos



Fonte: Arquitetura e Organização de Computadores – William Stallings

Tipos de sistemas operacionais



- Em lote (*batch*).
- Único programa (uniprogramação).
- Multiprogramação (multitarefa).

Primeiros sistemas



- Final da década de 1940 a meados da década de 1950.
- Sem sistema operacional.
- Programas interagem diretamente com o hardware.
- Dois problemas principais:
 - **Escalonamento.**
 - **Tempo de preparação.**

Sistemas de tempo compartilhado



- Modelo utilizado atualmente;
- Multiprogramação permite que uma série de usuários interajam com o computador.

O segredo da multiprogramação: Escalonamento



- Chave para multiprogramação.
- Por fases:
 - Longo prazo.
 - Médio prazo.
 - Curto prazo.

Escalonamento a longo prazo



- Determina quais programas são submetidos para processamento (tamanho e prioridade).
- Ou seja, controla o **grau de eficiência** de multiprogramação.
- Uma vez submetido, um *job* (**tarefa**) torna-se um processo para o escalonador a curto prazo;

Escalonamento a médio prazo



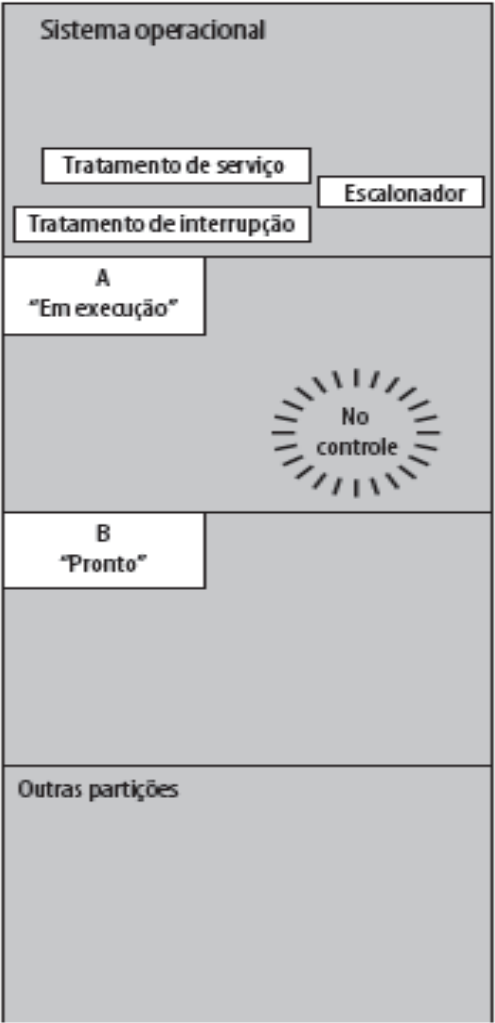
- Parte da função de troca de processo;
- Normalmente baseado na necessidade de gerenciar a multiprogramação.
- Se não há memória virtual, o gerenciamento de memória também é um ponto (cuida da memória).

Escalonamento a curto prazo

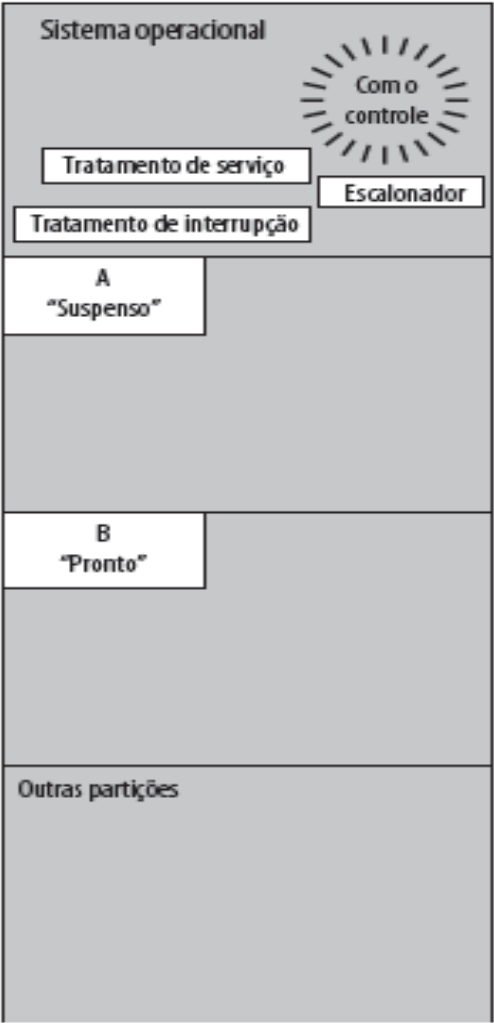


- Despachante;
- Decisões de nível mais baixo de qual tarefa executar em seguida.
- Ou seja, qual tarefa **realmente usa** o processador no próximo intervalo de tempo.

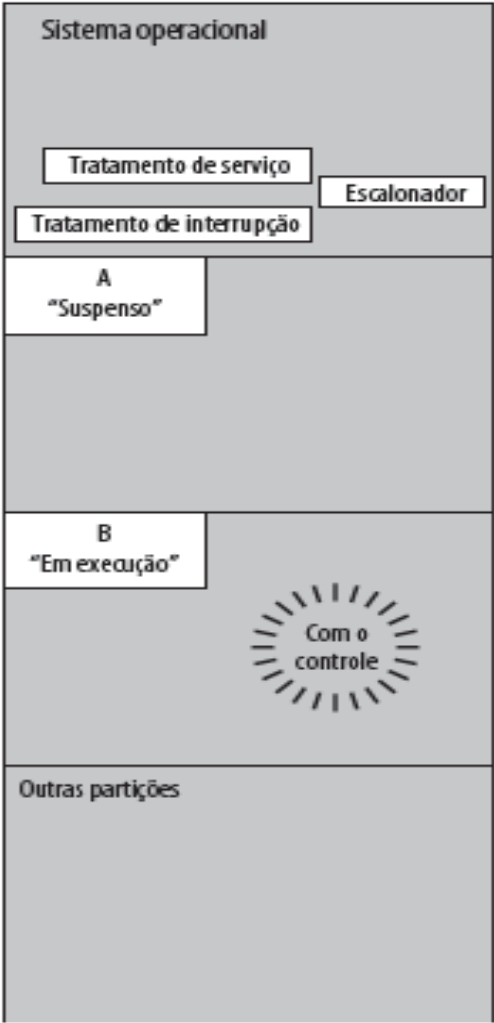
Exemplo de escalonamento



(a)



(b)



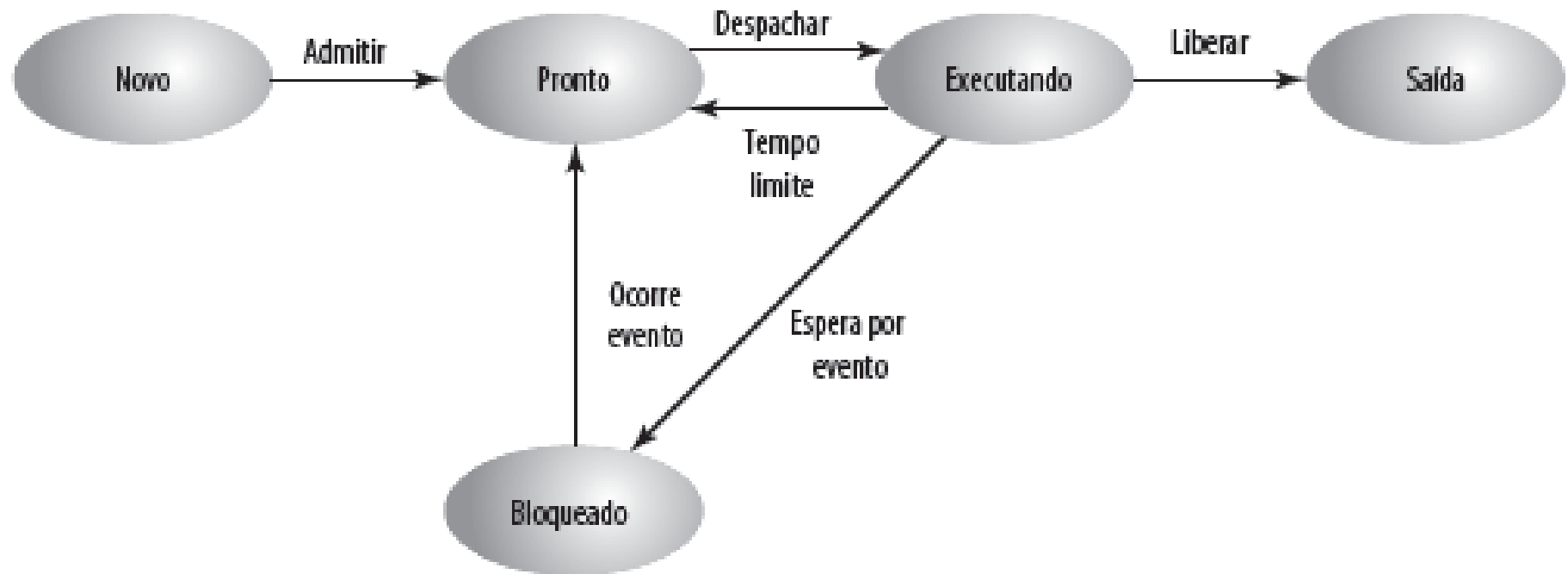
(c)

Fonte: Arquitetura e Organização de Computadores – William Stallings

Modelo de processo com cinco estados



Um processo tem alguns estados que ele pode assumir:



Fonte: Arquitetura e Organização de Computadores – William Stallings

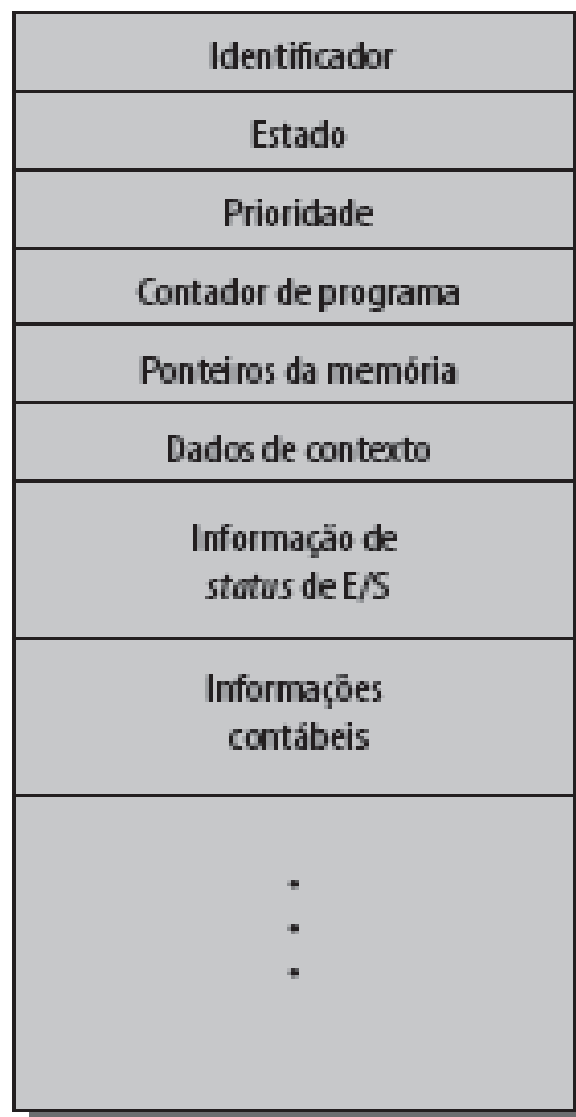
Bloco de controle de processo (BCP)



Elementos que compõem o controle de um processo:

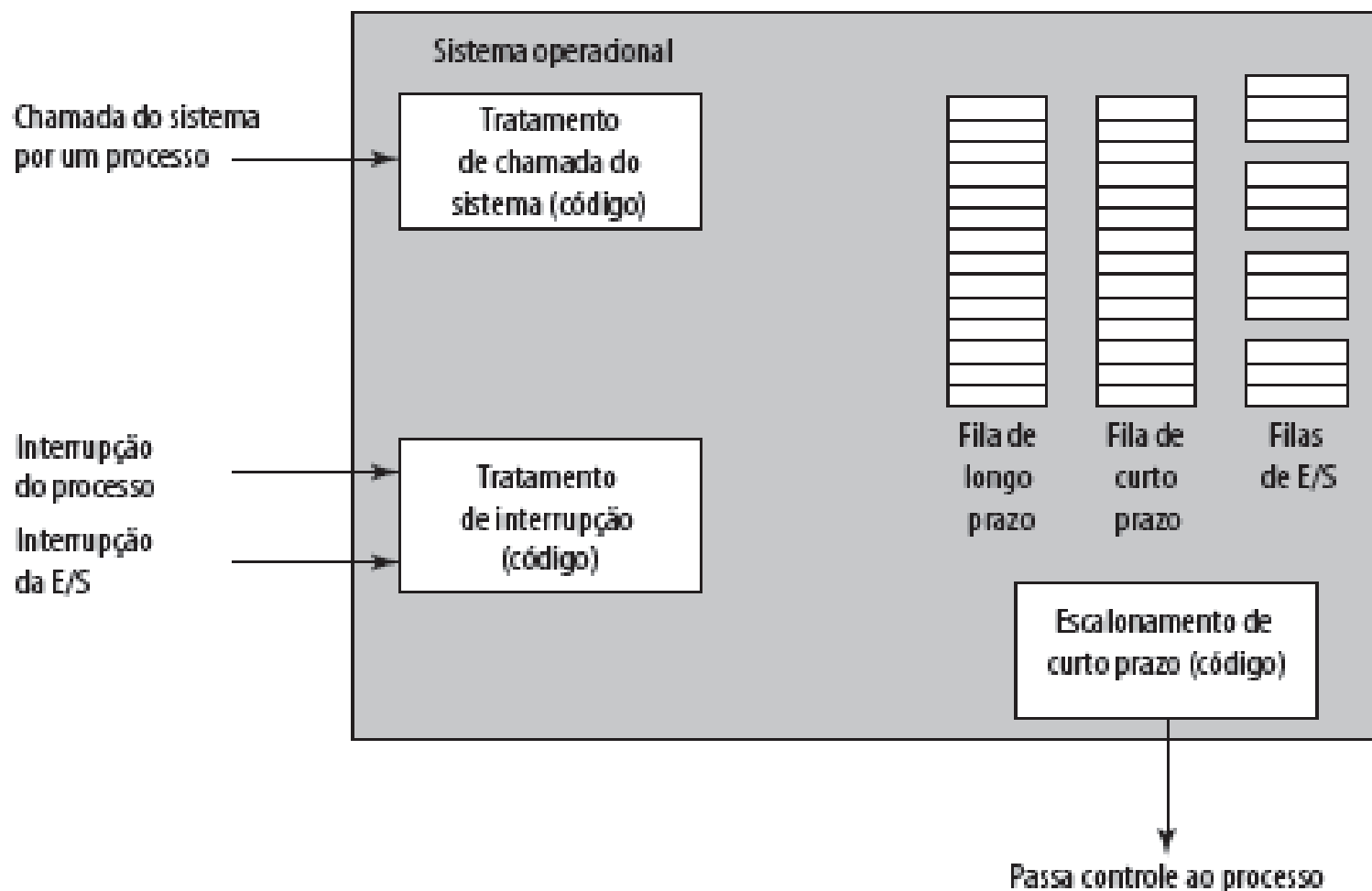
- Identificador.
- Estado.
- Prioridade.
- Contador de programa.
- Ponteiros de memória.
- Dados de contexto.
- Status de E/S.

Diagrama do bloco de controle de processo



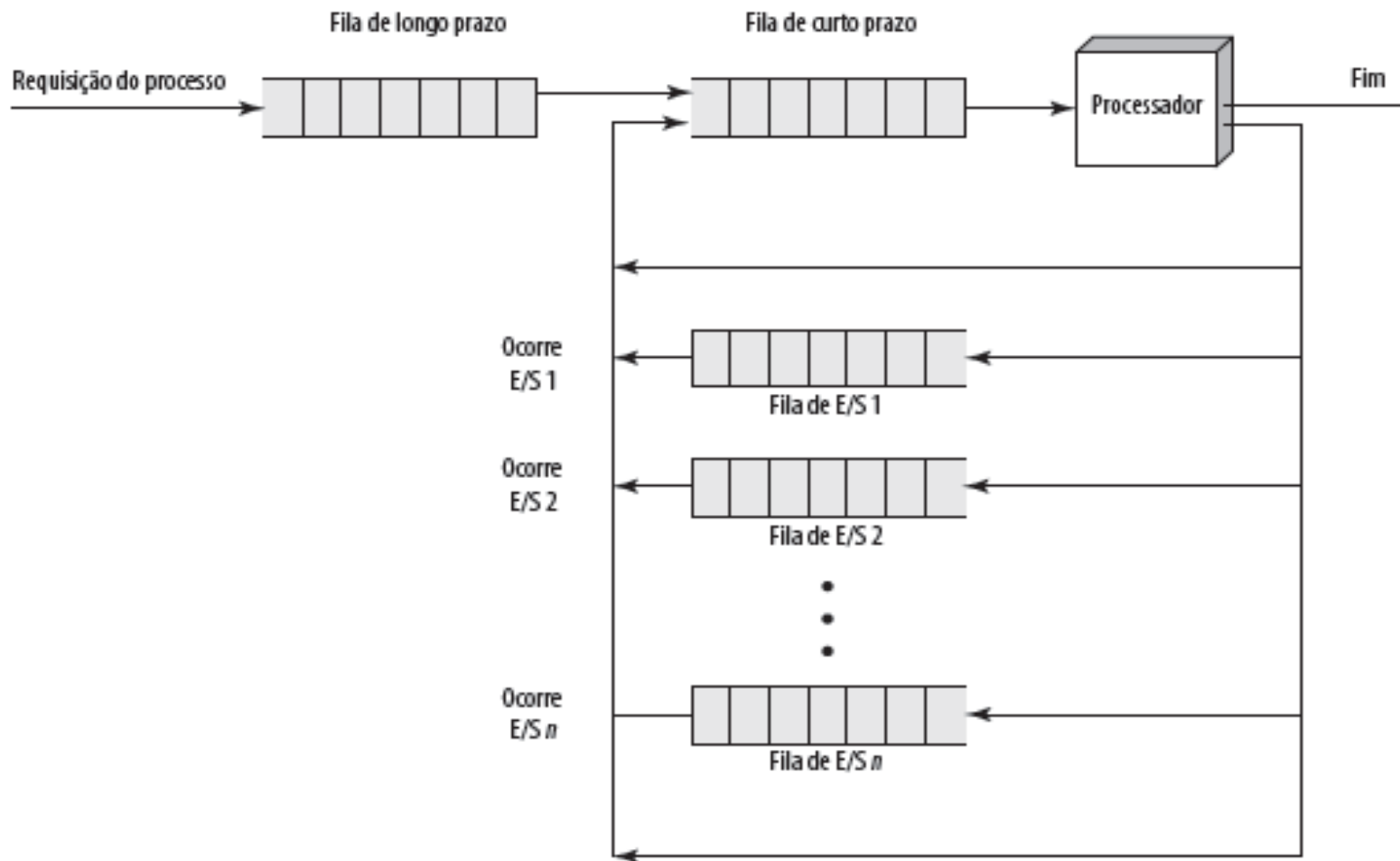
Fonte: Arquitetura e Organização de Computadores – William Stallings

Principais elementos do sistema operacional



Fonte: Arquitetura e Organização de Computadores – William Stallings

Escalonamento de processador



Fonte: Arquitetura e Organização de Computadores – William Stallings



- Uniprogramação:
 - Memória dividida em duas.
 - Uma para sistema operacional (monitor).
 - Uma programa atualmente em execução.
- Multiprogramação:
 - Parte do “usuário” é subdividida e compartilhada entre processos ativos.



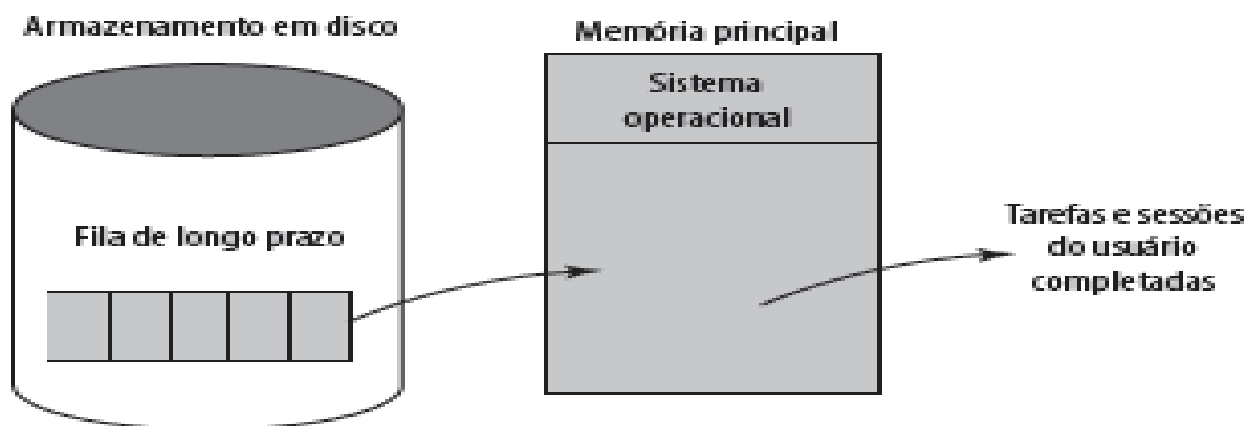
- Problema: E/S é tão lenta, em comparação com a CPU, que até mesmo em sistema de multiprogramação a CPU pode estar ociosa na maior parte do tempo.
- Soluções:
 - Aumentar memória principal.
 - Cara.
 - Leva a programas maiores.
 - Swapping**

O que é swapping?

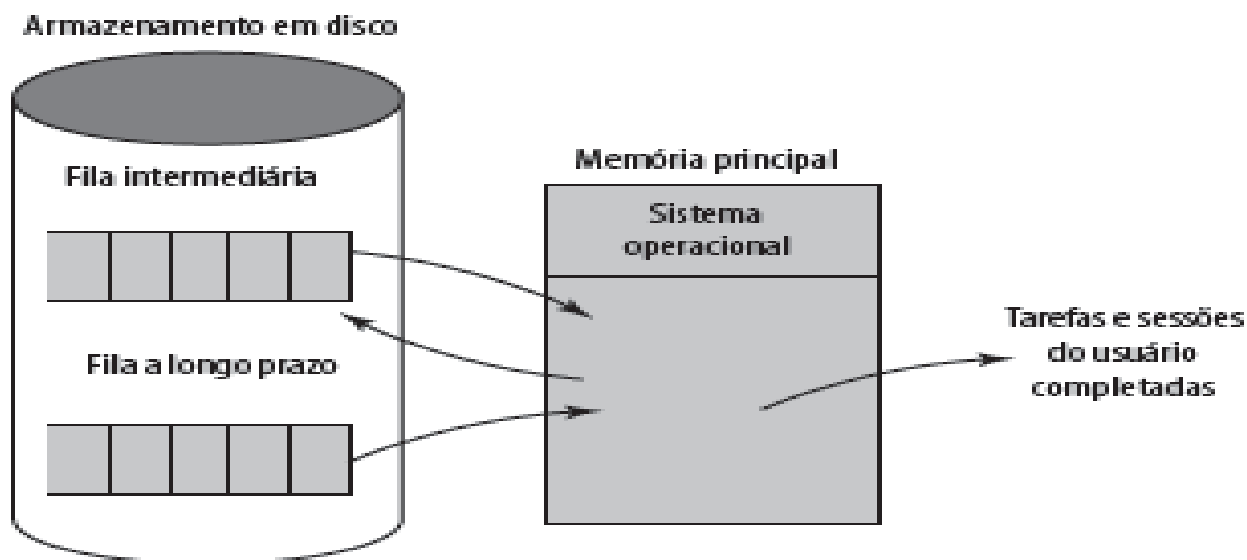


- Fila a longo prazo dos processos armazenados no disco.
- Processos trocados para a memória quando existe espaço disponível.
- Swapping também é um processo de E/S.

Uso do swapping



(a) Escalonamento de tarefas simples



(b) Swapping

Particionamento da memória



- Outro problema além do Swapping.
- Dividir a memória em seções para alocar processos (incluindo sistema operacional).

Partições de tamanho variável



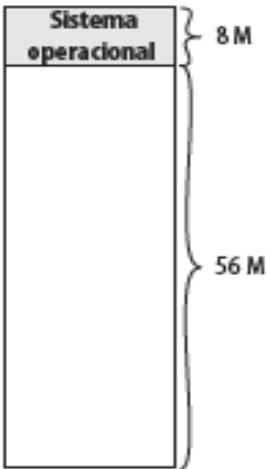
- Alocam exatamente a memória requisitada a um processo.
- Isso leva a um buraco no final da memória, muito pequeno para ser usado.
- Quando todos os processos estão bloqueados, retira um processo e traz outro.
- Novo processo pode ser menor que o processo removido.
- Outro buraco.

Partições de tamanho variável

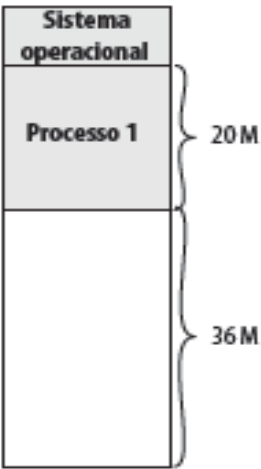


- Soluções:
 - **Aglutinação:** juntar buracos adjacentes em um grande buraco.
 - **Compactação:** de vez em quando, percorre a memória e move todos os buracos para um bloco livre (desfragmentação de disco).

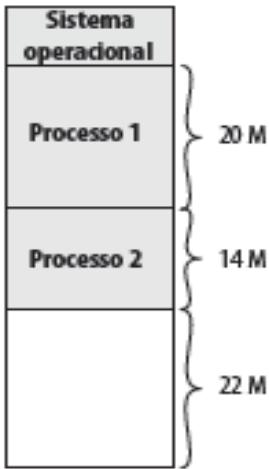
Efeito do particionamento dinâmico



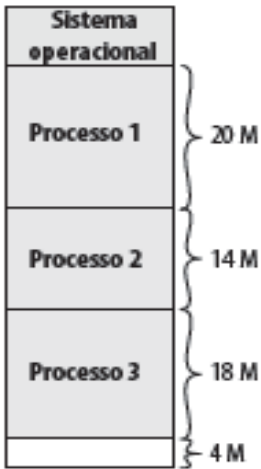
(a)



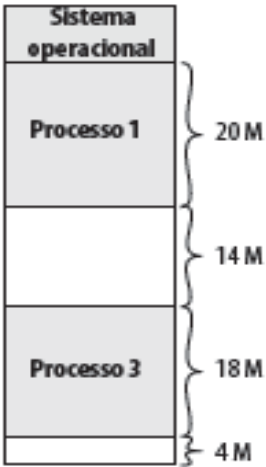
(b)



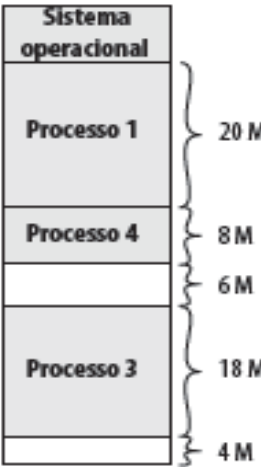
(c)



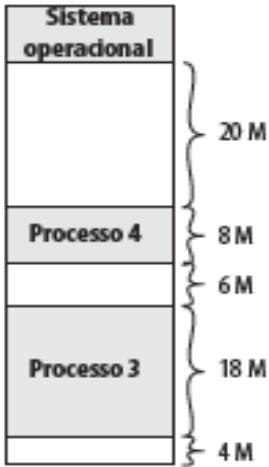
(d)



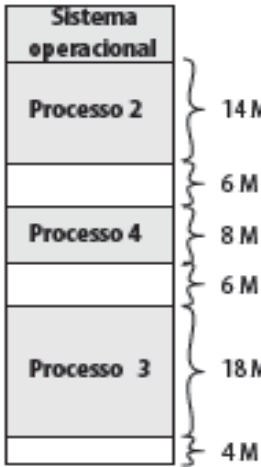
(e)



(f)



(g)



(h)



Pode ser uma solução:

- Nenhuma garantia de que o processo será carregado no mesmo local na memória.
- Instruções contêm endereços:
 - Localizações dos dados.
 - Endereços para instruções (desvio).



- Endereço lógico – relativo ao início do programa.
- Endereço físico – local real na memória (desta vez).
- Conversão automática usando endereço de base.