

Projeto Integrador Extensionista III

Ciência da Computação
1º Sem. 2024

Parte 1

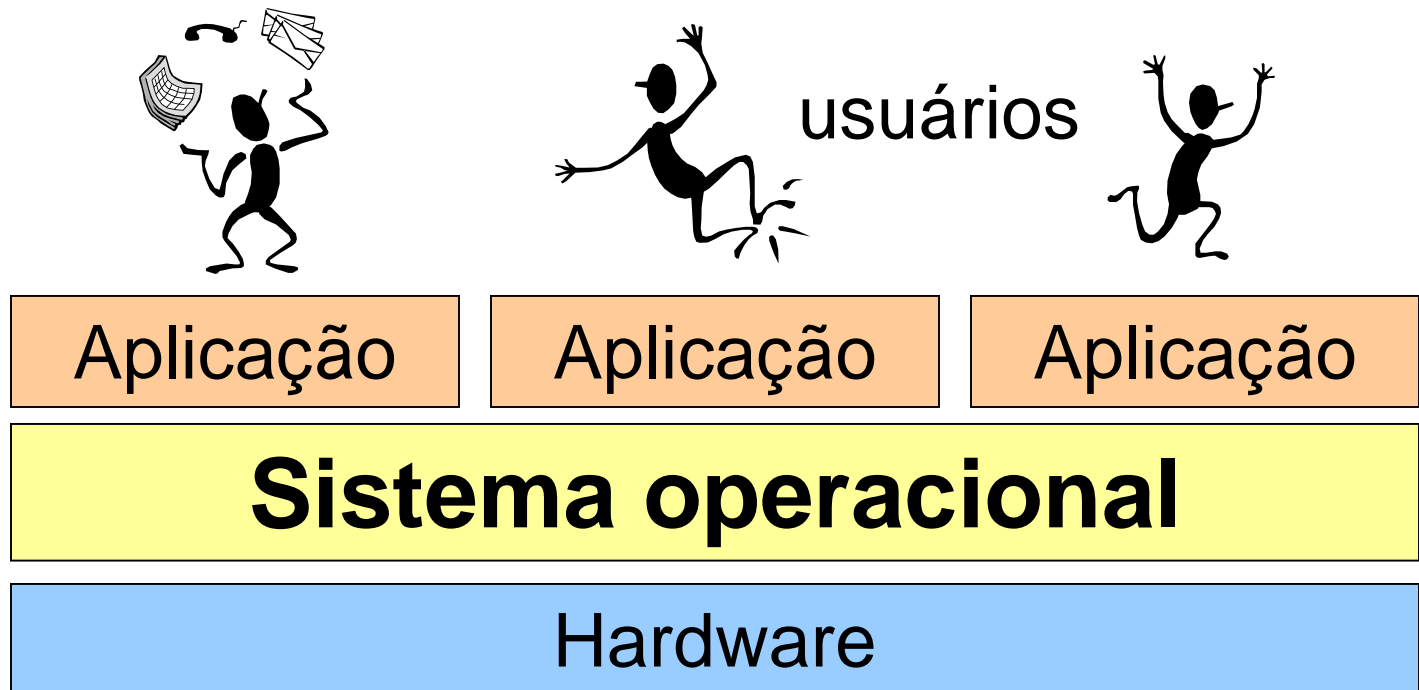
Conceitos de Sistemas Operacionais

Prof. Alencar Melo Jr., Dr. Eng.
IFTM – *Campus* Ituiutaba

alencar@iftm.edu.br

Sistema operacional

- Camada de *software* entre o *hardware* e as aplicações dos usuários





Funções do sistema operacional

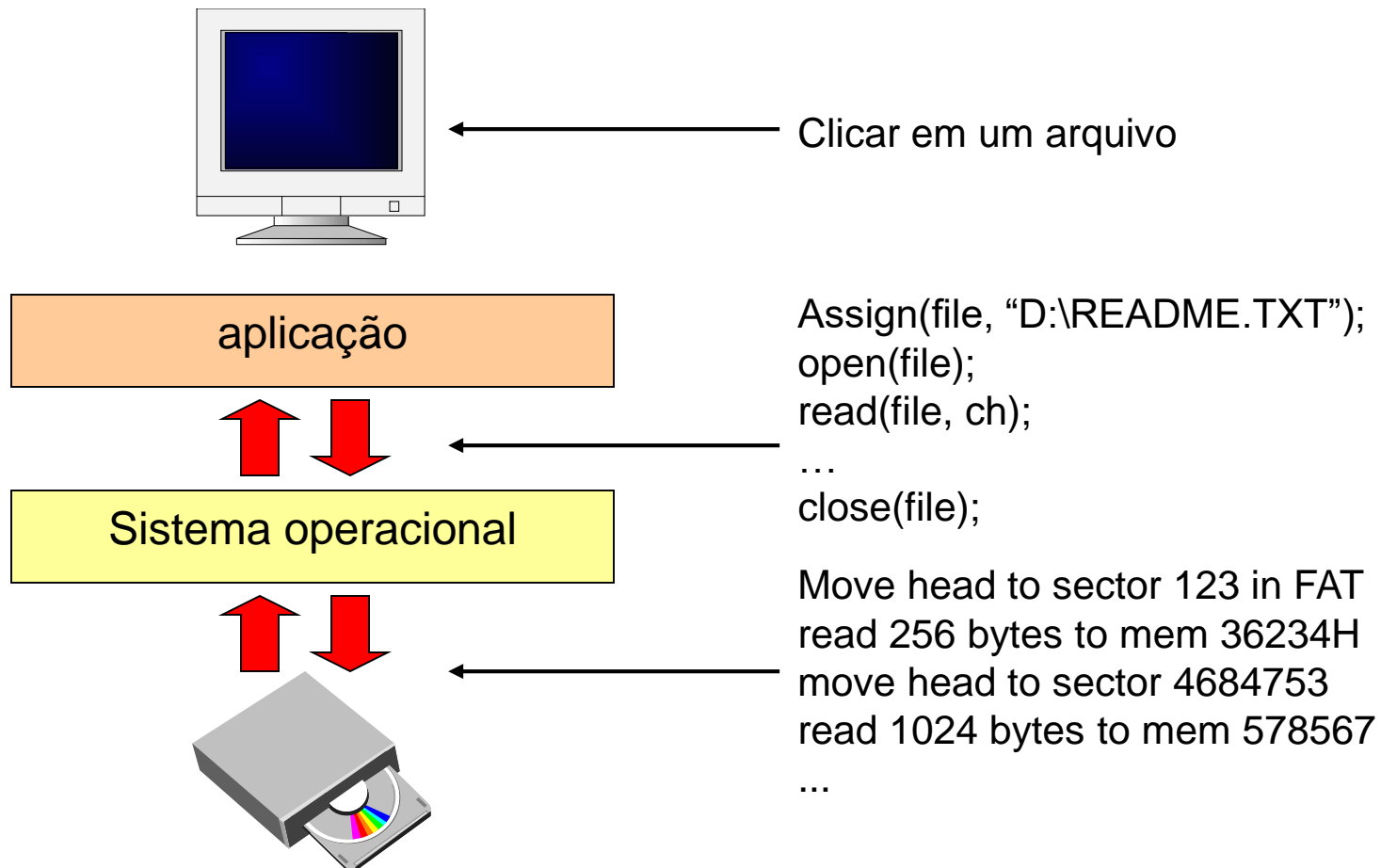
- A partir da máquina real (hardware), criar uma **máquina abstrata**.
- **Gerenciar os recursos** de hardware disponíveis às aplicações, de modo **conveniente** e **eficiente**.
 - Qual aspecto é mais importante para um SO do lado servidor, conveniência ou eficiência? E do lado cliente?



O SO como máquina abstrata

- Ocultar a complexidade do hardware.
- Oferecer interfaces padronizadas de acesso ao hardware.
- Permitir uma visão homogênea de dispositivos distintos.

O SO como máquina abstrata

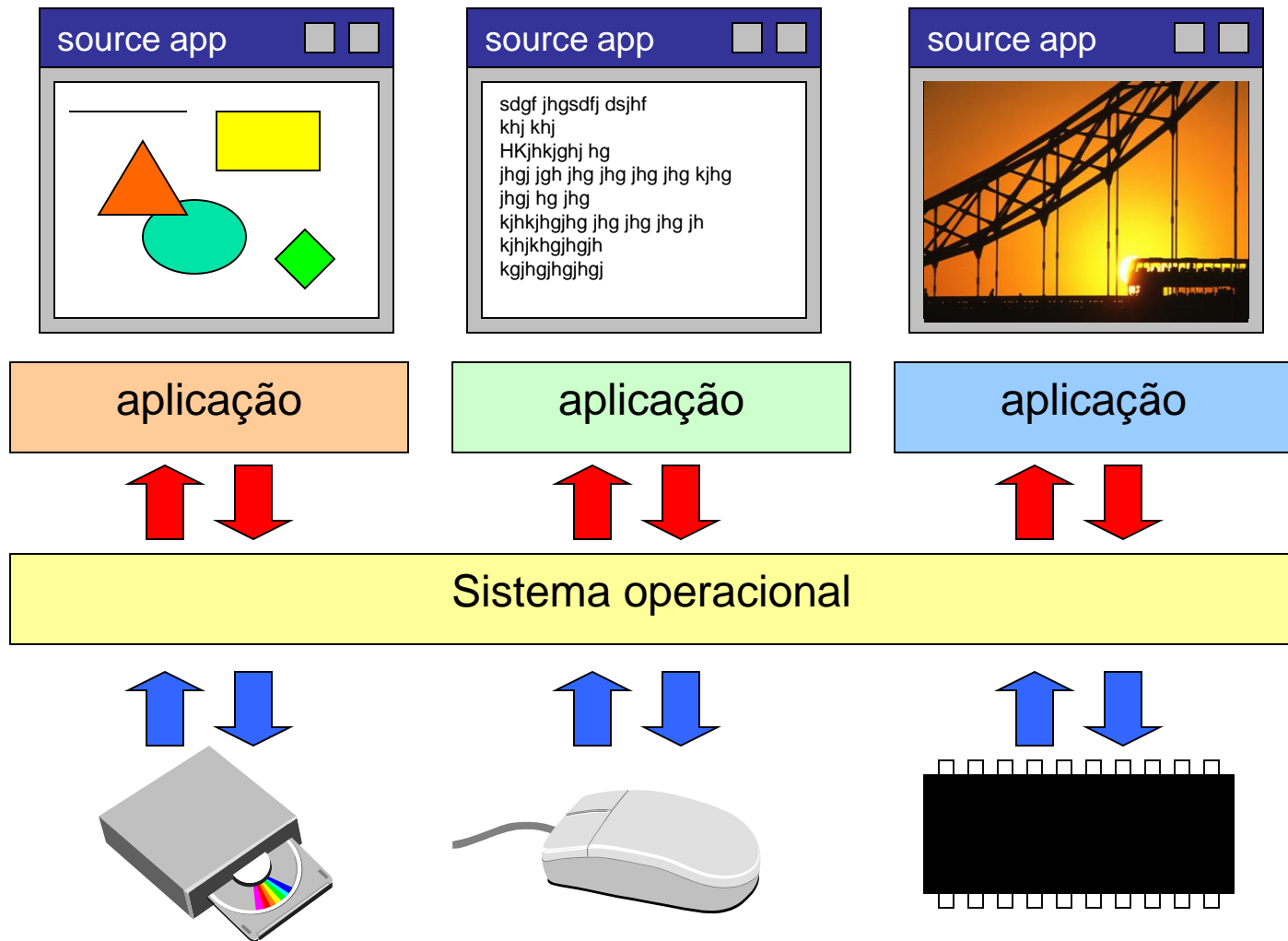




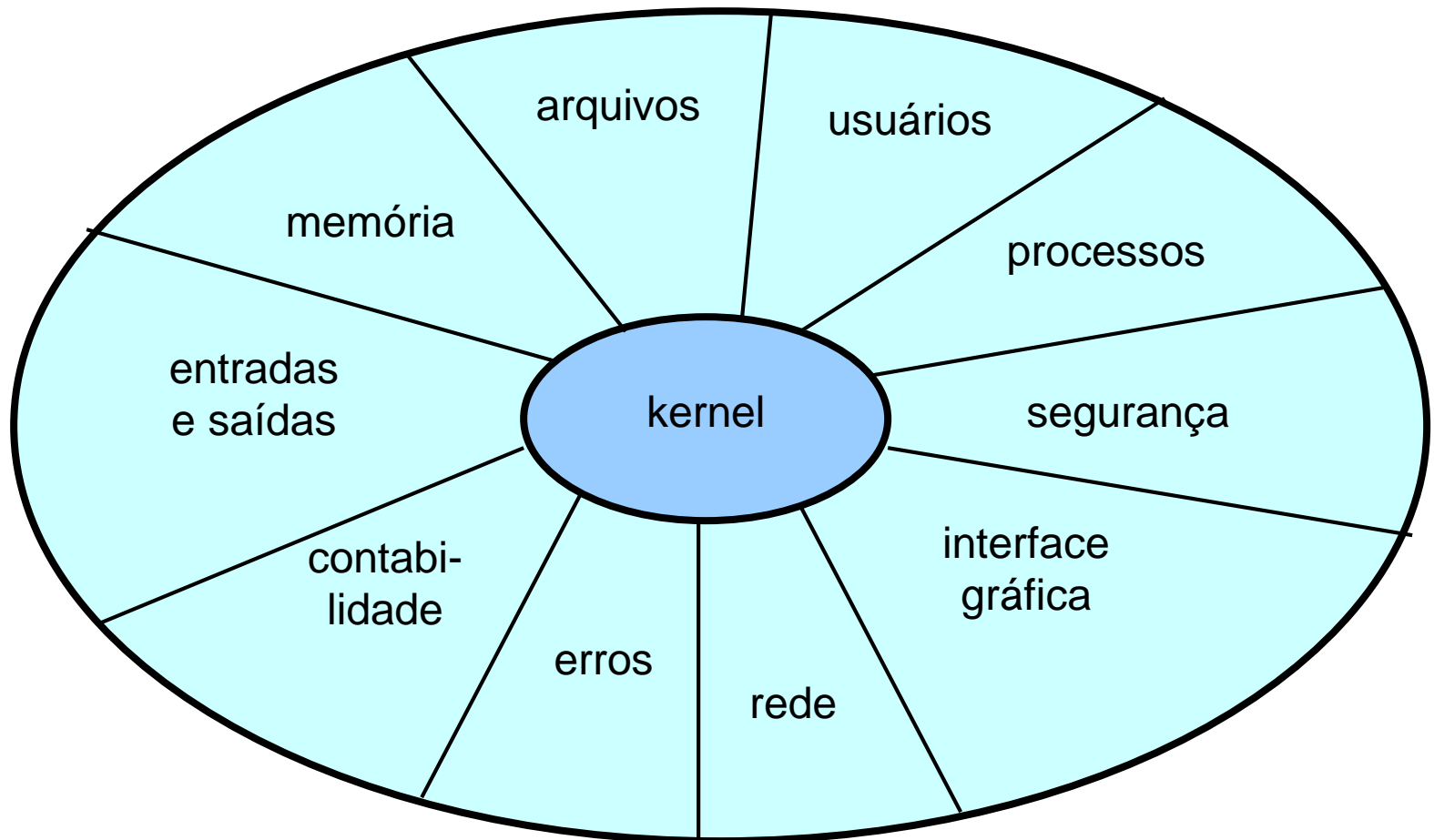
O SO como gerente de recursos

- Recursos da máquina
 - processadores
 - espaço em memória
 - arquivos
 - conexões de rede
 - dispositivos externos
- Controle de acesso
 - equilibrar uso
 - evitar conflitos

O SO como gerente de recursos



Componentes de um SO





O kernel

- Concentra o acesso ao hardware
 - drivers de dispositivo
 - gerência de acesso
- Provê funções básicas
 - operações de acesso ao hardware
 - noção básica de processos
 - comunicação entre processos (IPC)
- É executado em modo privilegiado
- A porta de entrada para o kernel são as **chamadas para o Sistema** (*system calls*).



Programas e processos

- Programa:

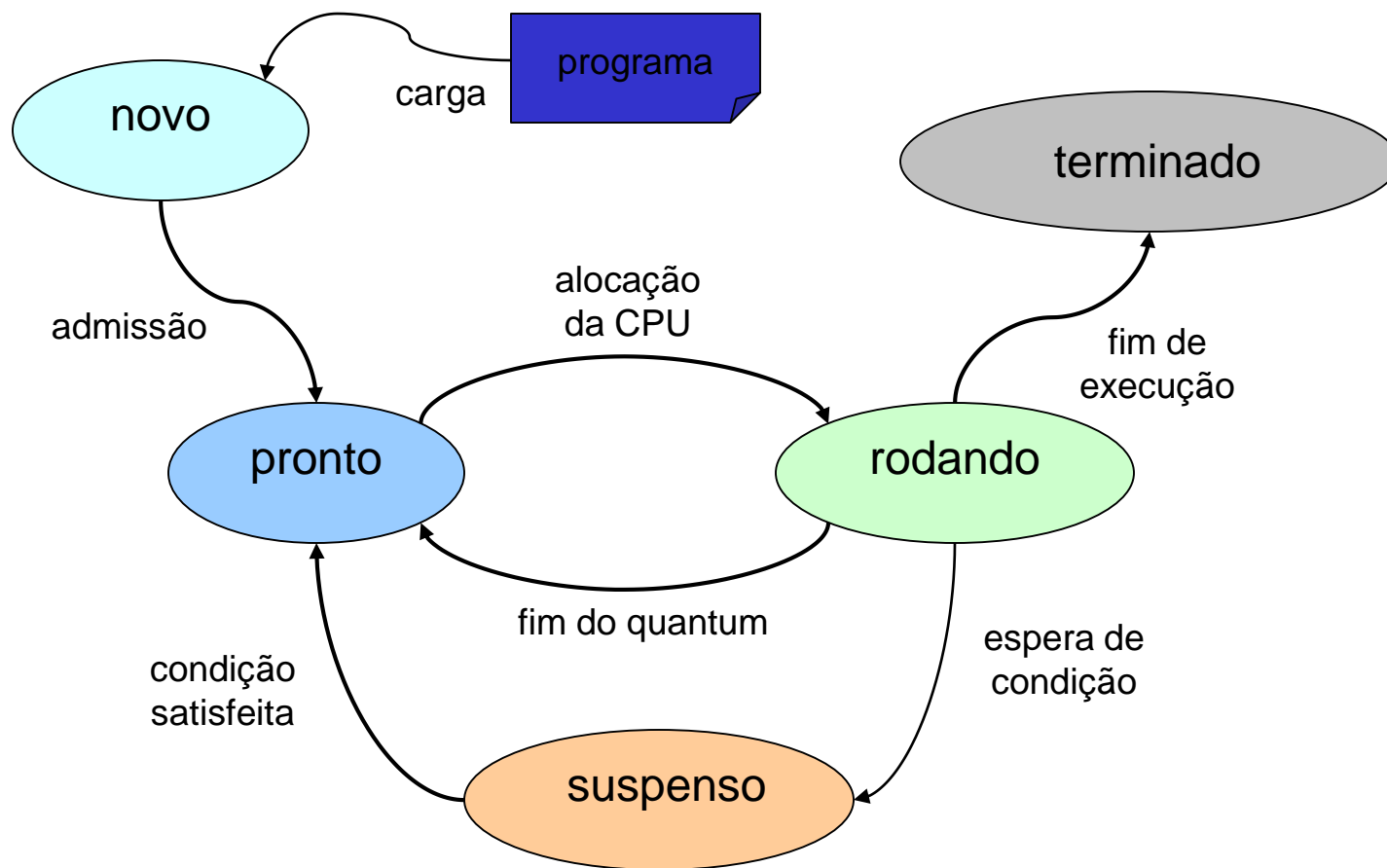
- noção estática
- conjunto de instruções
- não possui um estado interno

- Processo:

- noção dinâmica
- programa em execução
- possui um estado interno

- Processo é um programa em execução.

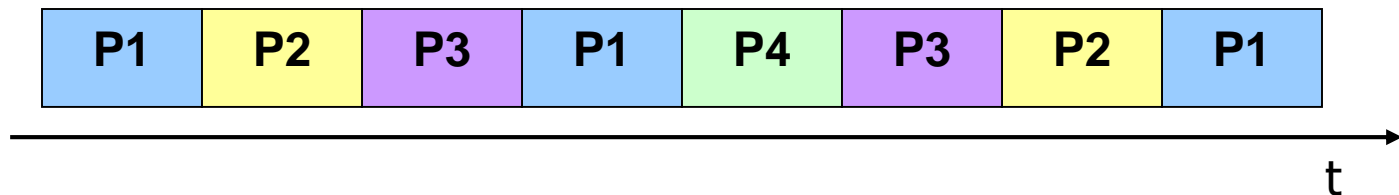
Estados dos processos





Compartilhamento de tempo

- Processos prontos compartilham CPU
- Ilusão de uma CPU por processo
- Cada processo recebe uma pequena fatia de tempo (< 20 ms)





Processo = contexto + atividade

- Contexto:
 - estruturas de dados (variáveis em memória)
 - descritores de arquivos e sockets
 - outros recursos exclusivos
- Atividade:
 - fluxo de execução
 - registradores, pilha de execução
- Alternar processos = trocar contextos

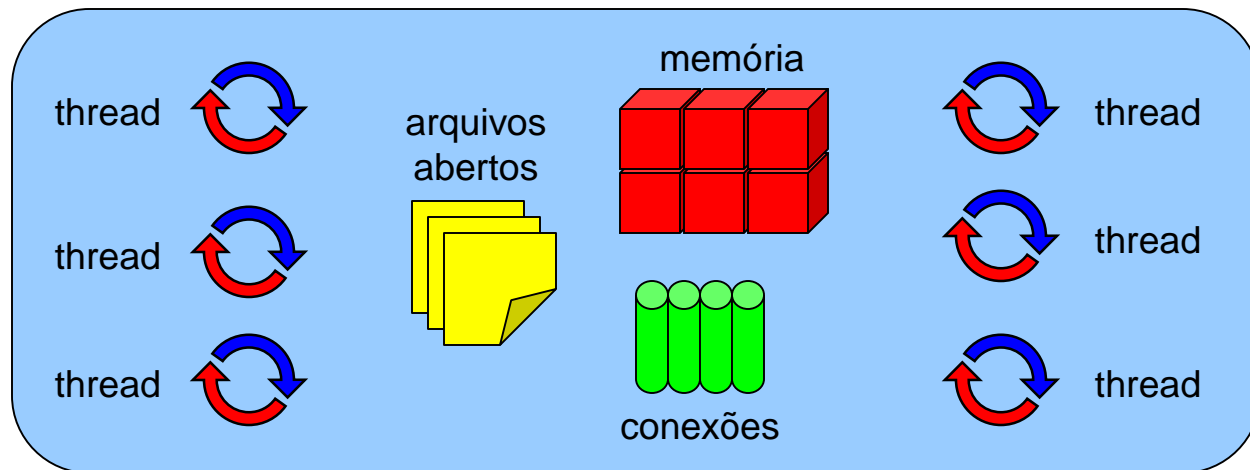


A troca de contexto

- A troca de contexto é uma operação cara (+/- 10% do tempo de CPU)
- Para trocar de atividade, precisamos sempre trocar de contexto ?
 - Aplicações com concorrência interna
- Solução = threads:
 - múltiplas atividades em **um mesmo contexto**

Threads

- Permitem separar contexto de tarefa
 - **Processo:** unidade de **contexto**
 - **Thread:** unidade de **atividade**





Vantagens das threads

- Menor custo para alternar atividades
 - Trocar de thread é muito mais rápido que trocar de processo
- Permite trocas mais frequentes
 - melhor tempo de resposta das aplicações
- Maior facilidade para construir aplicações com múltiplas atividades
 - Mais adequado para implementar diversas atividades que compartilham recursos comuns



Desvantagens das threads

- Nem todos os sistemas as suportam
- Falta de padronização
- Exigem maior controle de concorrência no acesso aos recursos comuns
- Risco de dados inconsistentes
- Aplicações menos robustas
 - se uma thread travar, a aplicação pode travar



Implementação de threads

- No kernel:
 - o núcleo do SO oferece os recursos para criar e gerenciar múltiplas threads por processo
 - Presente nos sistemas operacionais modernos
- Por bibliotecas:
 - o núcleo do SO somente conhece processos
 - trocas de contexto explícitas (programadas)
 - Presente em SOs antigos (SunOS, Netware, AIX etc.)



Gerência de memória

- Proteger contexto dos processos
 - Cada processo tem sua área de memória
 - evitar interferências entre processos
 - evitar acessos de memória indevidos
- Flexibilizar o uso da RAM
 - aplicações usam muita memória
 - gerenciar memórias virtuais (discos)
 - compartilhar memória entre processos



Gerência de memória

- Computadores têm memória limitada
- Grandes sistemas e aplicações
 - Windows XP: ~ 100 Mbytes RAM
 - Outlook 2K: ~ 30 Mbytes RAM
 - Windows 10: ~ 1-2 Gbytes RAM
 - Ubuntu: ~ 0,7-1,5 Gbytes RAM
- Em suma:
 - Memória é um recurso escasso
 - Seu uso deve ser otimizado



Memória real X virtual

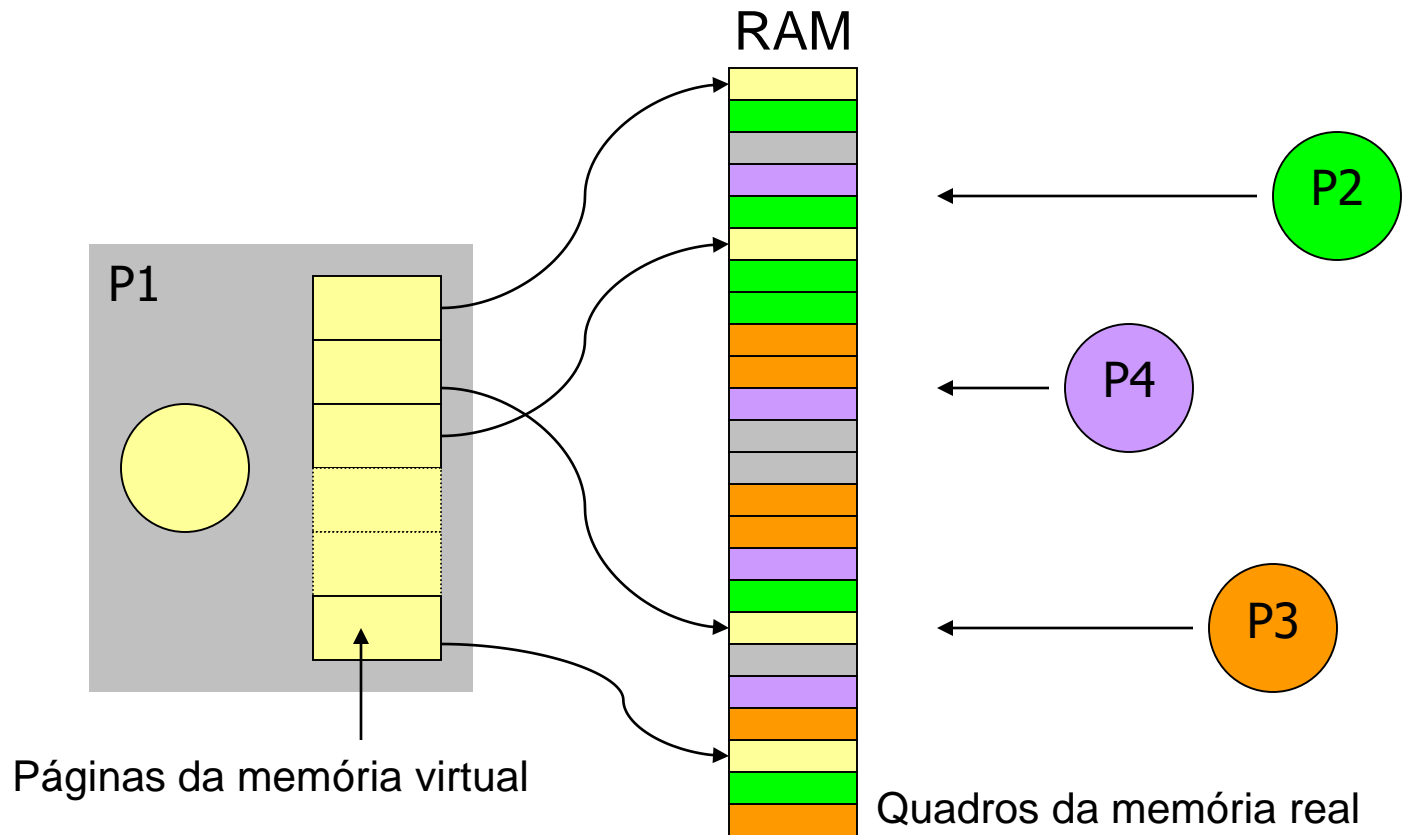
- Memória real:

- quantidade de RAM no computador
- acesso direto só permitido ao kernel
- dividida em “quadros” com ~ 4 Kbytes

- Memória virtual:

- imagem da memória vista pelos processos
- maior que a RAM (CPUs 32 bits: 4 Gbytes)
- estruturada em “páginas” com ~ 4 Kbytes

Memória paginada

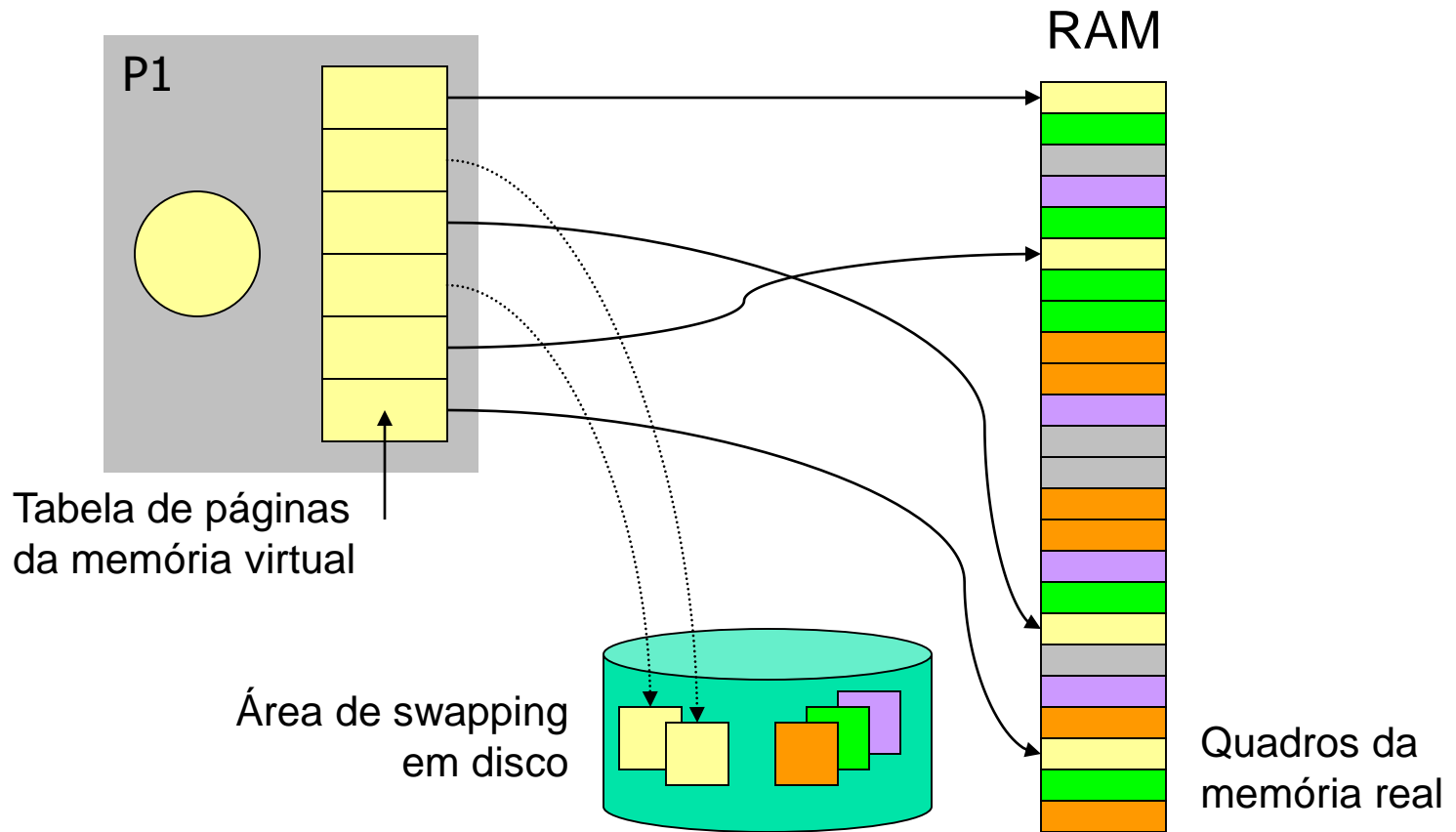




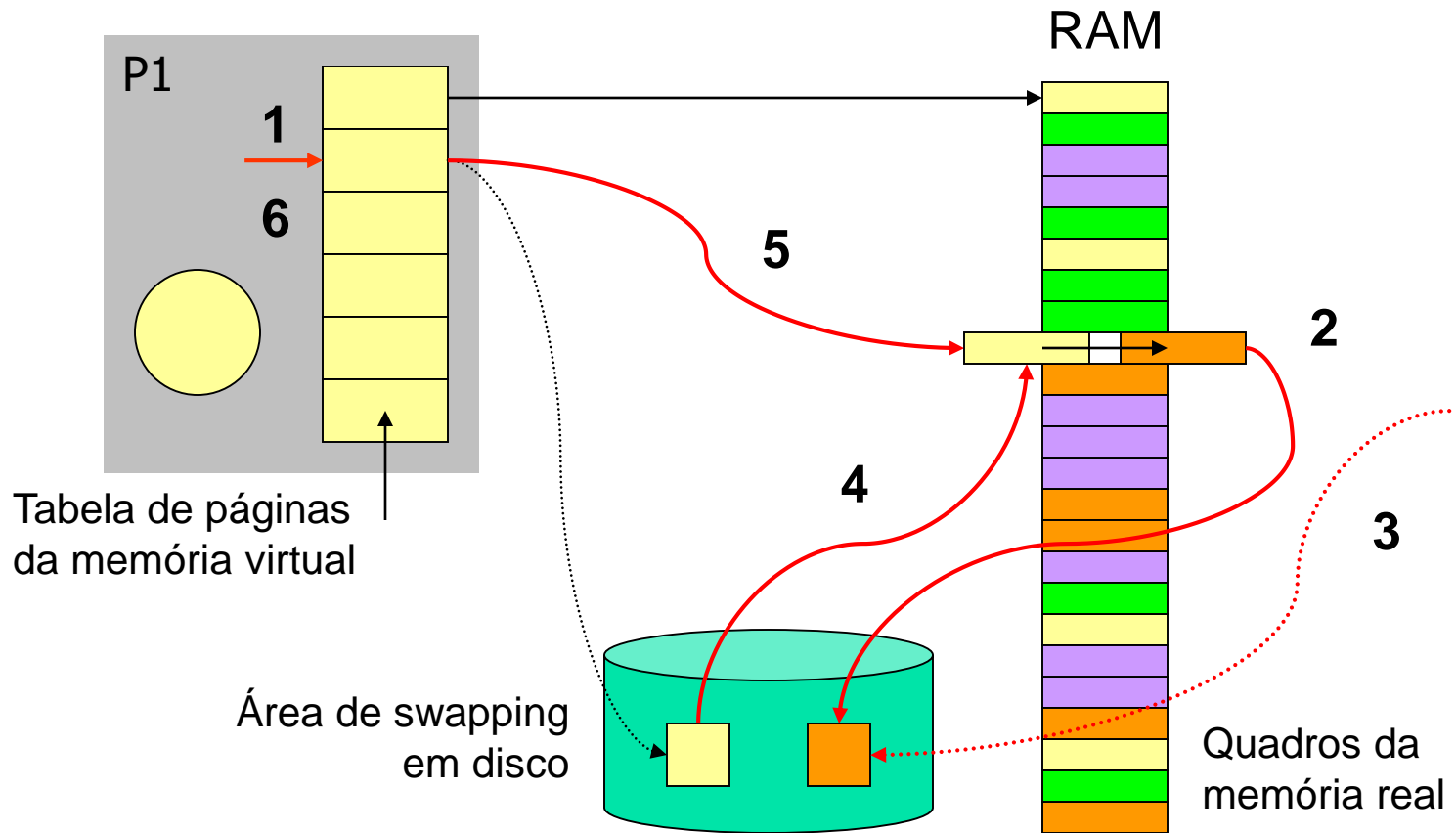
Memória paginada

- Relação entre páginas e quadros
 - tabelas mantidas pelo kernel
 - mapeamento pode ser trabalhoso
 - é necessário o auxílio do hardware
- Uso da RAM torna-se muito flexível
 - Toda a RAM é aproveitável (não há “buracos”)
 - Processos vêem os mesmos endereços
 - Possibilidade de swapping em disco
 - Quadros de RAM podem ser compartilhados

Swapping em disco



Passos do swapping





Passos do swapping

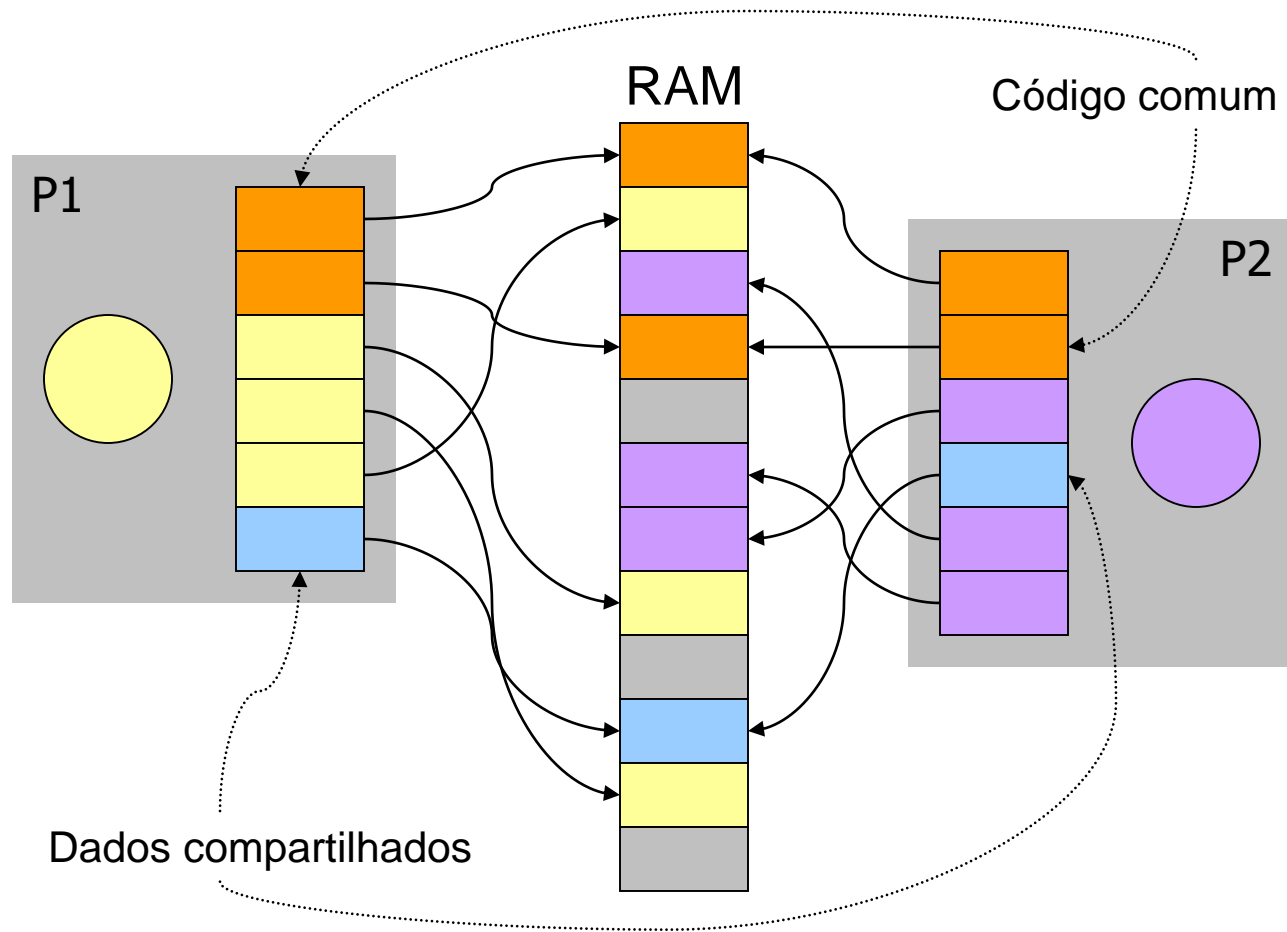
- 1: O processo tenta acessar uma área de memória não mapeada em RAM, gerando um “page fault”.
- 2: O kernel escolhe um frame de RAM a ser liberado, e envia sua página para a área de swap.
- 3: A tabela de páginas do processo “vítima” é atualizada (página enviada para o swap).
- 4: A página solicitada é carregada no quadro livre.
- 5: A tabela de página do processo é atualizada.
- 6: O processo acessa o endereço desejado.



Compartilhamento de memória

- Processos podem compartilhar páginas
 - vários processos com o mesmo código
 - processos que precisam compartilhar dados
- Manipulação das tabelas de páginas
 - mesma posição da RAM (mesmo quadro)
 - endereço distinto (página) em cada processo

Compartilhamento de RAM





Sistema de arquivos

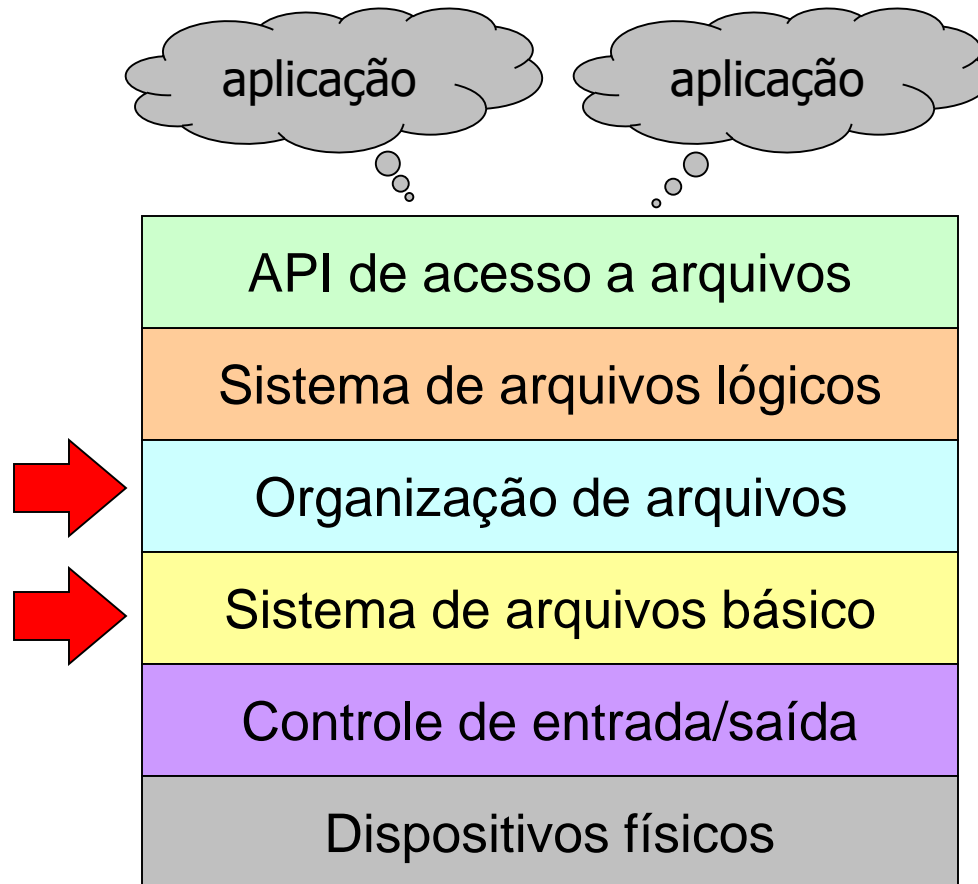
- **Dispositivos com tecnologias variadas**

- CD-ROM, DAT, HD, Floppy, ZIP
- SCSI, IDE, ATAPI, ...
- sistemas de arquivos em rede

- **Interfaces de acesso uniforme**

- visão homogênea dos dispositivos
- transparência para as aplicações

Arquit. da gerência de arquivos





Dispositivos e drivers

■ Dispositivo físico:

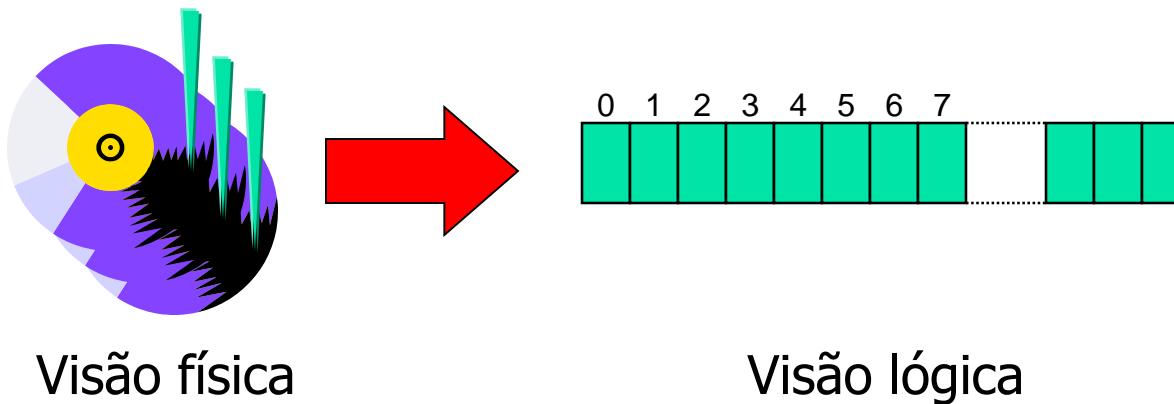
- armazenamento dos dados
- estruturados em blocos de bytes (~ 512 bytes)
- CD-ROM, hard disk, floppy, fitas

■ Driver de dispositivo:

- acesso em baixo nível aos dispositivos
- gerencia interrupções e DMA (*Direct Memory Access*)
- mapeia acessos a trilhas/setores/cabeças em operações sobre portas de E/S do dispositivo

Visão dos dispositivos

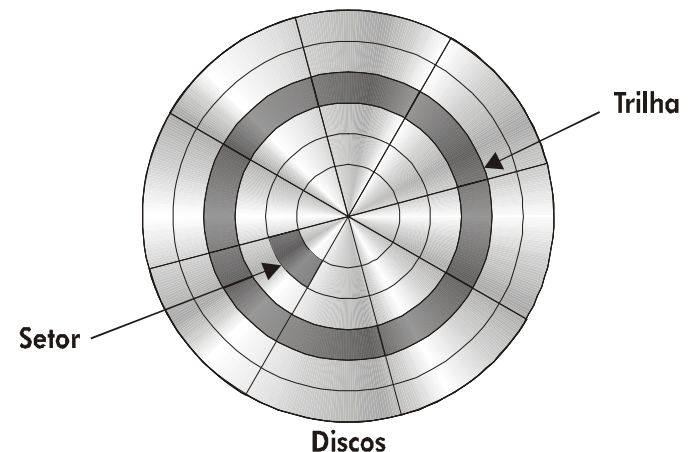
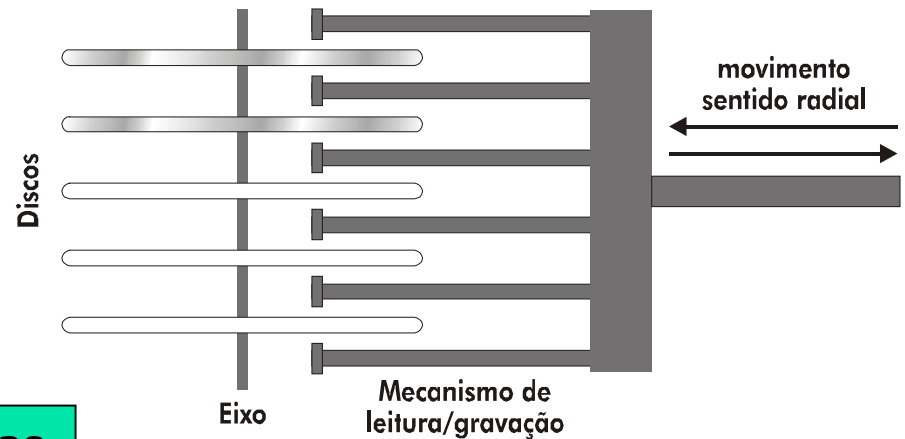
- **Visão física:** cabeças, trilhas, setores
- **Visão lógica:** vetor de blocos idênticos
- **Função do sistema de arquivos básico:**



Discos magnéticos

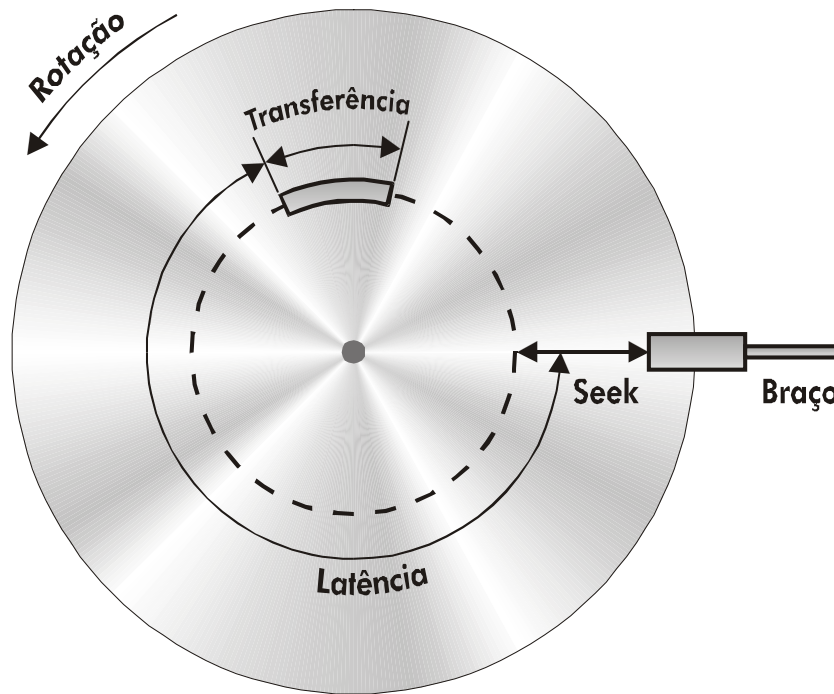
- Estrutura de um disco magnético

Cilindro vertical: definido pelas trilhas dos diferentes discos que ocupam a mesma posição vertical



Discos magnéticos

- Tempo de acesso



Causa maior impacto no desempenho do disco:
tempo de busca (*seek time*)



Discos magnéticos

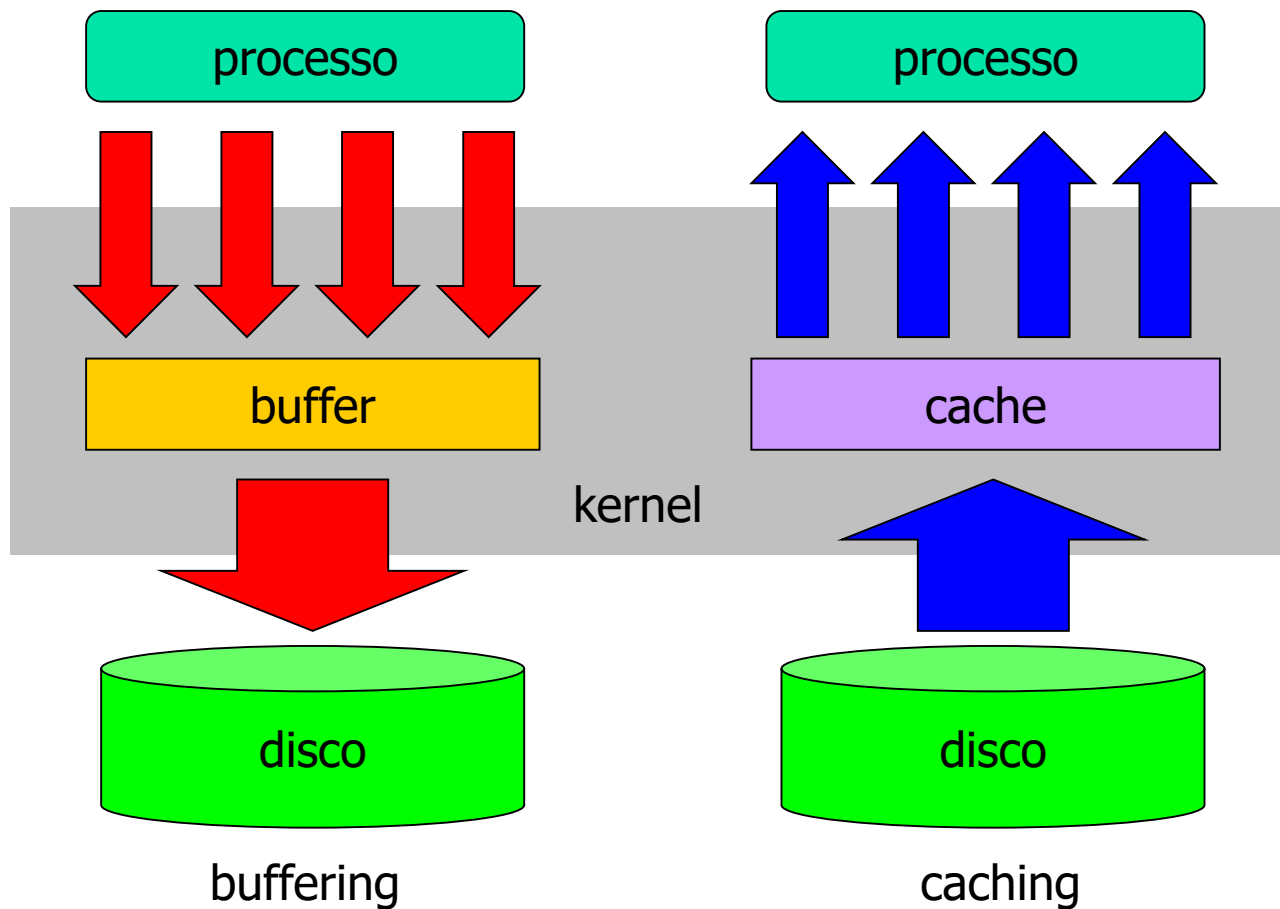
- **Formatação física:** definida pelo fabricante, especifica o formato dos setores físicos
- **Formatação lógica:** definida pelo sistema de arquivos
- **Entrelaçamento (*Interleaving*):** numera-se os setores de forma não contígua
- **Sistemas de arquivos com *journaling*:** podem se restabelecer de modo automático após uma queda



Sistema de arquivos básico

- Aciona comandos de leitura/escrita nos drivers de dispositivos.
- Mostra o dispositivo como um vetor de blocos de mesmo tamanho.
 - **Blocos lógicos** entre 512 bytes e 8 Kbytes
- Pode efetuar *buffering* e *caching*:
 - *Buffering*: otimizar acessos reais em escrita.
 - *Caching*: otimizar acessos reais em leitura.

Buffering & caching

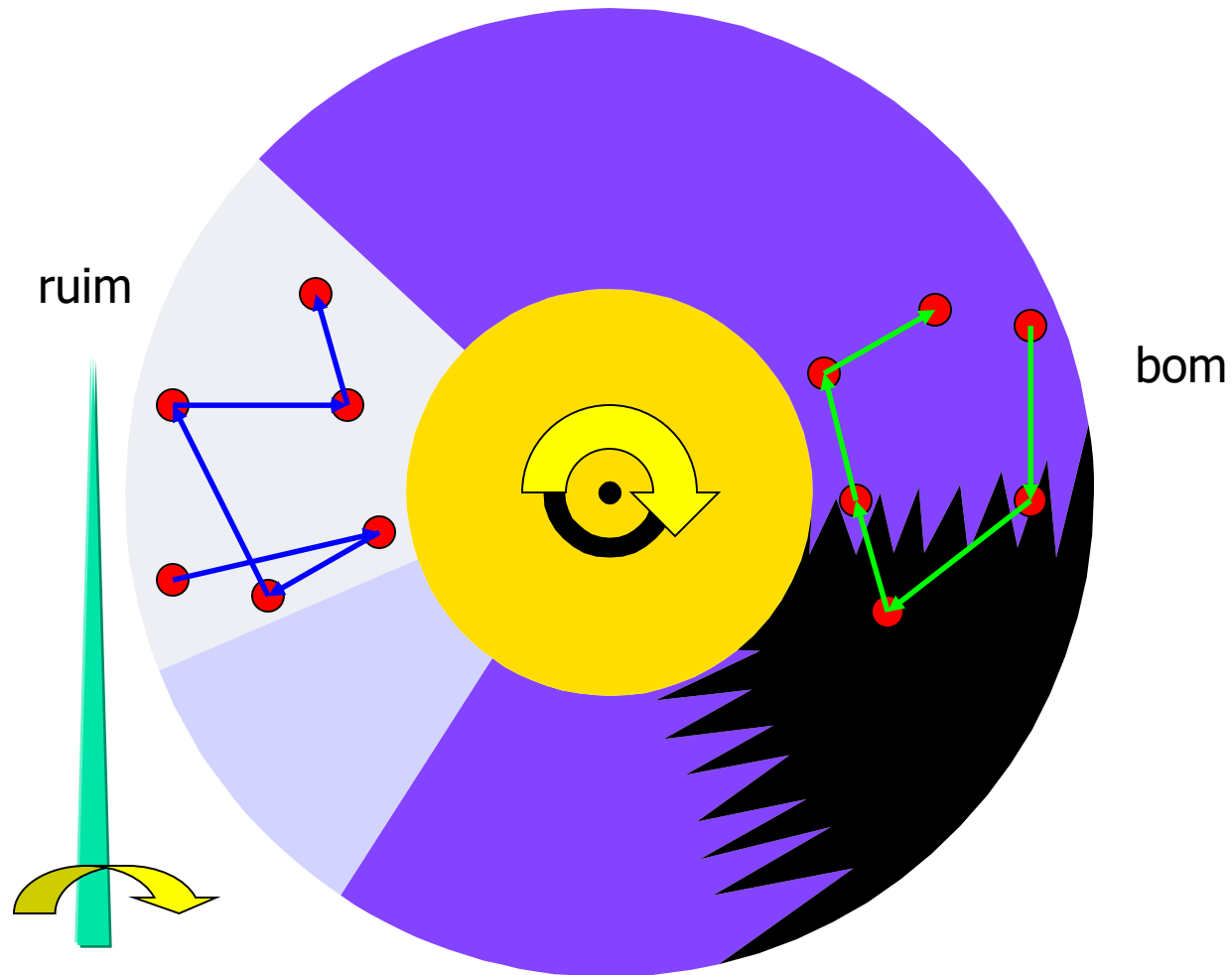




Escalonamento de disco

- Acesso ao disco por vários processos
 - processos acessam áreas distintas
 - o disco é um dispositivo LENTO
 - desempenho de I/O pode ser péssimo
- Acesso ao disco deve ser escalonado
 - escolher ordem de atendimento dos pedidos de acesso aos discos
 - buscar o melhor desempenho

Exemplos de escalonamento





Organização de arquivos

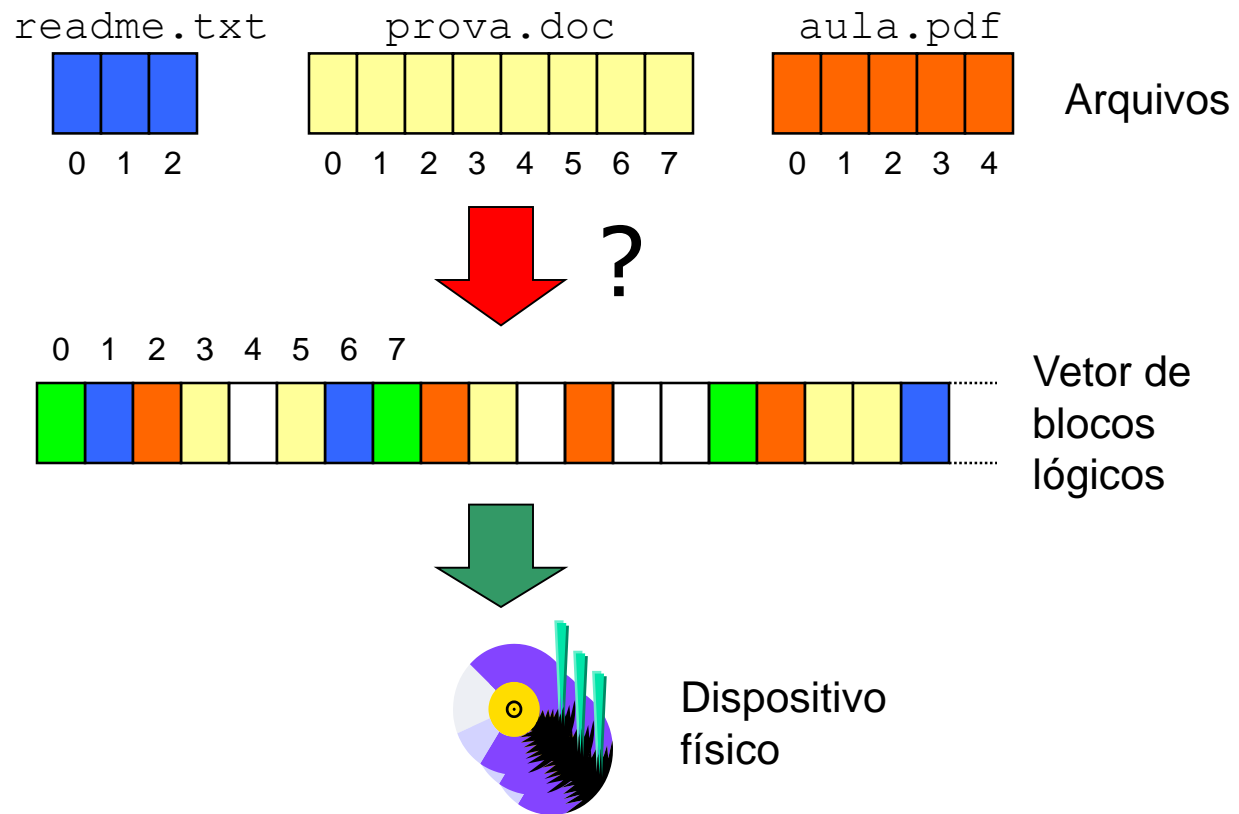
■ Problema:

- Como armazenar diversos arquivos dentro de um único vetor de blocos lógicos ?
- Cada arquivo também deve ser visto como uma seqüência de blocos lógicos.

■ Restrições:

- flexibilidade de alocação
- rapidez de acesso (seqüencial e aleatório)
- eficiência no uso do espaço real em disco

Organização de arquivos





Técnicas de alocação

- Formas de mapear os blocos dos arquivos em posições no vetor de blocos lógicos
- Alocação **contígua** de arquivos
- Alocação em **listas encadeadas**
- Alocação **indexada**



Classificação dos SO

- Executar tarefas simultâneas
 - mono-tarefa, multi-tarefas
- Suporte a várias CPUs
 - mono ou multi-processados (SMP)
- Usuários simultâneos
 - mono-usuário, multi-usuários
- Tempo de resposta
 - batch, interativos, de tempo real



Consolidação de SOs

- Diversos SOs para *psmartphones* e *netbooks* surgiram, mas apenas dois sobreviveram:
 - Android
 - iPhone OS
 - Jolicloud
 - WebOS (Palm Pre)
 - Windows Mobile
 - Symbian OS
 - RIM OS (BlackBerry)
 - Chrome OS



Questões para debate

- 1) Analise o potencial das tecnologias abertas para a realização de ações de extensão.
- 2) Dúvidas??