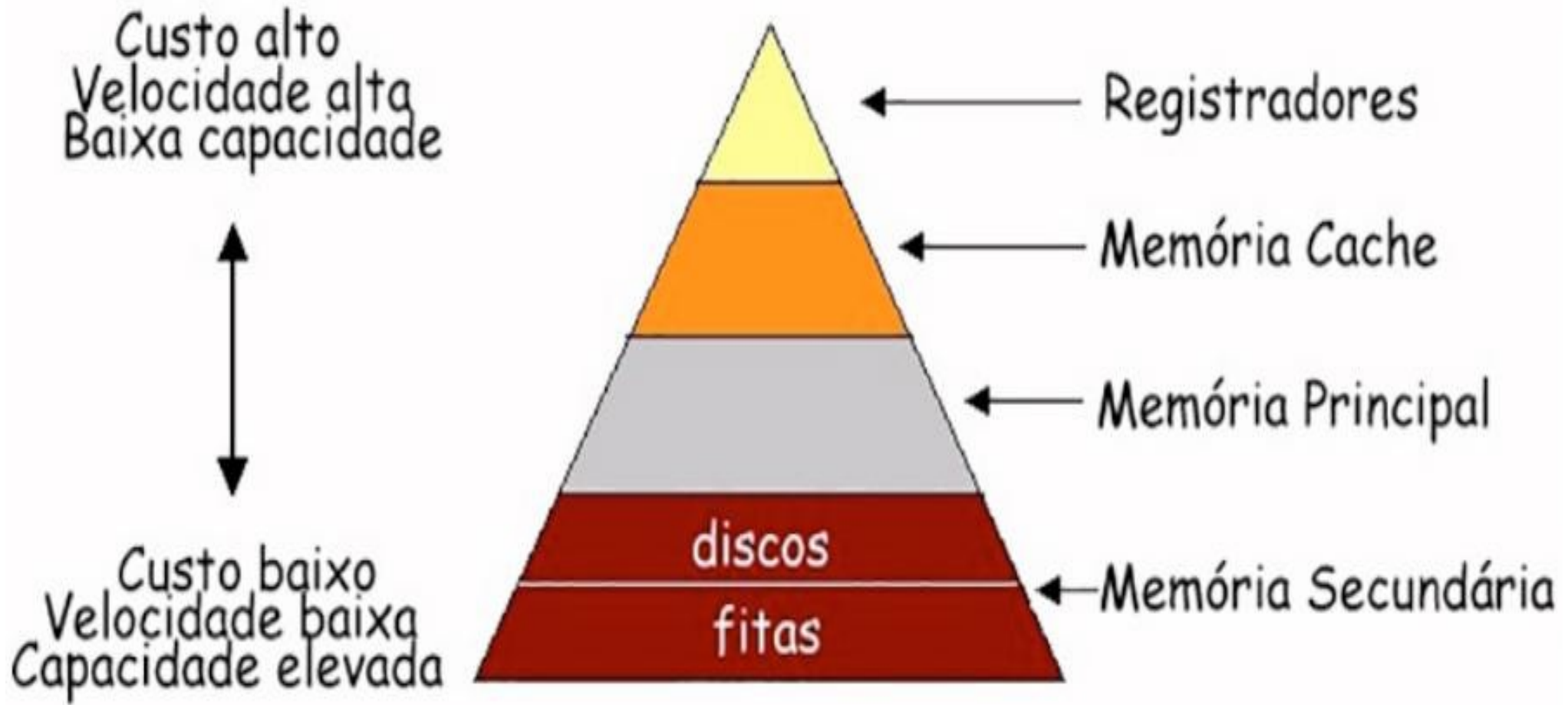


Gerenciamento de memória

Gerenciamento de memória

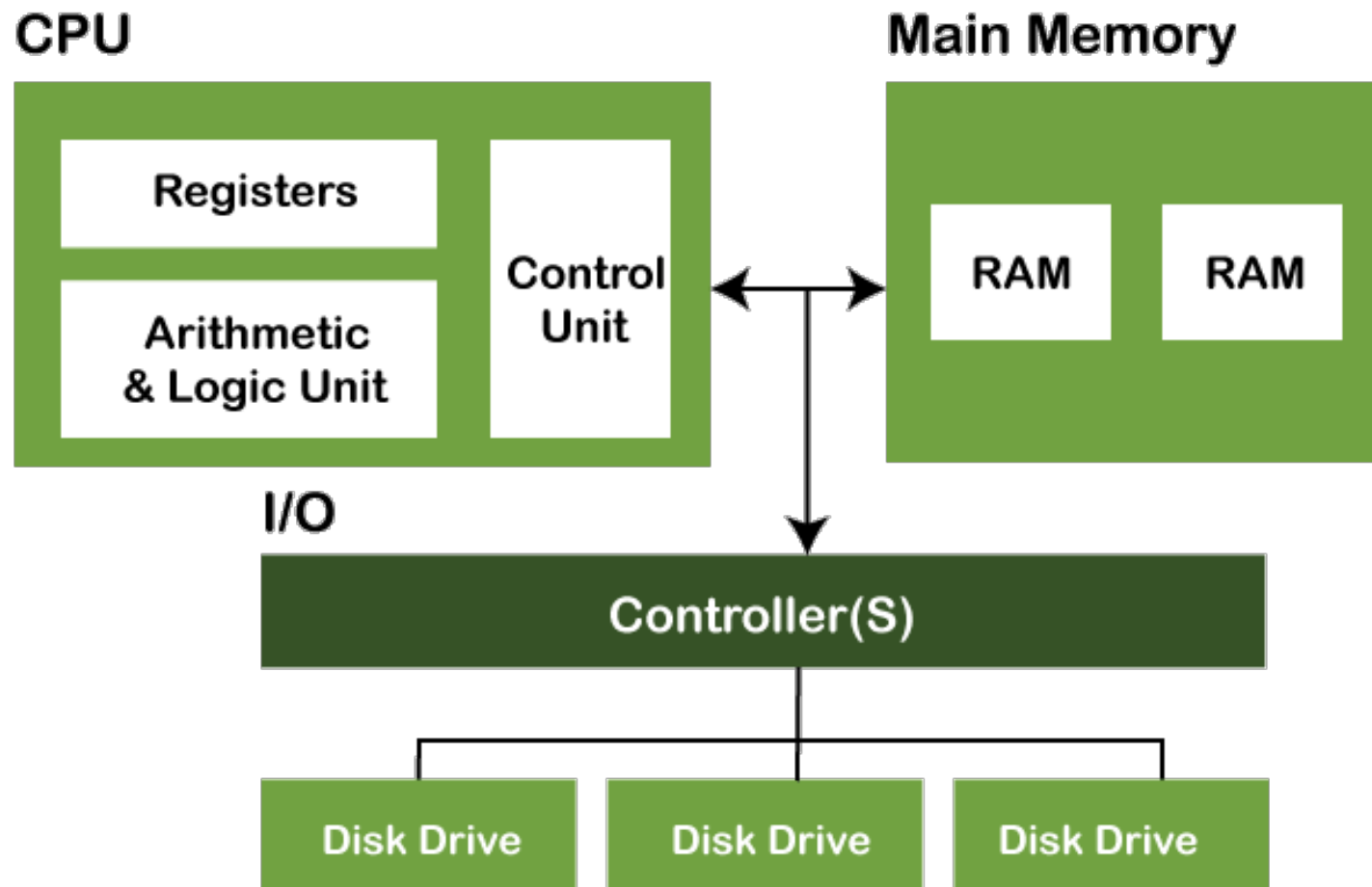
- Permite que vários processos sejam executados ao mesmo tempo, mantendo um bom desempenho do sistema
- O SO deve proteger as áreas de memória utilizadas pelos processos de forma que, se um programa tente acessá-la indevidamente, o sistema seja impedido
- Conceito de hierarquia de memória combinando
 - memória cache
 - memória principal RAM
 - memória secundária
 - baixo custo, alto armazenamento, não volátil, ex.: disco

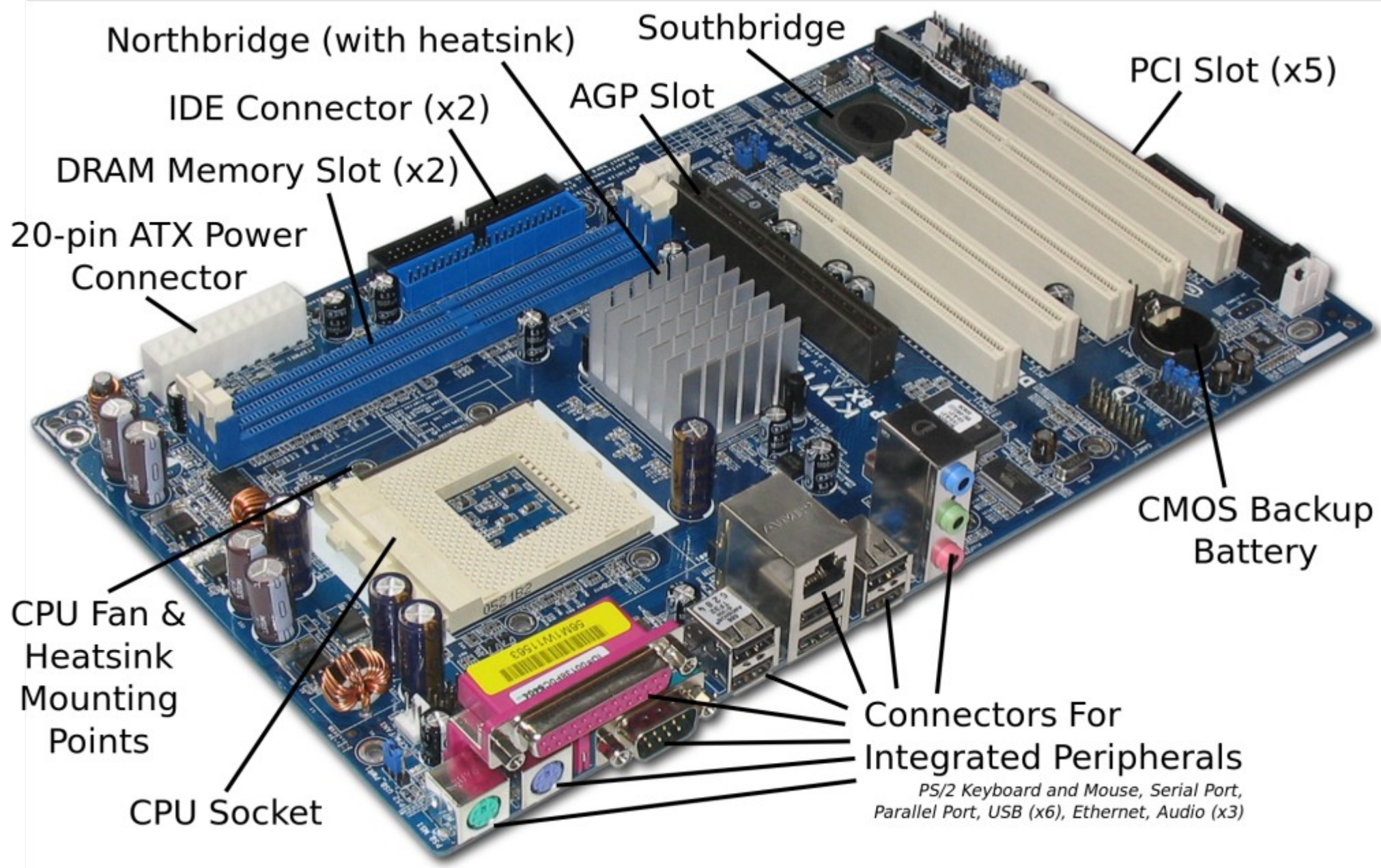
Hierarquia de memórias



Hierarquia de Memória

- Existe uma hierarquia de memória que combina diferentes tipos
- Registradores
- Memória Cache
- Memória Principal (RAM)
- Memória Secundária (disco/fita)
- Quanto mais alto na pirâmide: Mais caro, mais rápido, menor capacidade
- Quanto mais baixo na pirâmide: Mais barato, mais lento, maior capacidade





Memória: Recurso Histórico e Complexo

- Desde os primeiros sistemas operacionais, a memória era vista como um recurso caro e escasso
- Mesmo com a redução de custo e aumento da capacidade, gerenciar seu uso continua sendo fundamental e complexo
- A tendência das aplicações é consumir cada vez mais memória durante a execução

Gerenciador de Memória

- A hierarquia de memória é controlada pelo gerenciador de memória
- Ele gerencia quais partes da memória estão em uso e quais não estão
- É responsável por alocar memória quando processos precisam e liberá-la após o término
- Controla a troca de processos entre a memória e o disco quando a memória principal não é suficiente

Programação Eficaz e Gerenciamento de Memória ⁹

- Em linguagens de programação, especialmente como C/C++, é preciso realizar o gerenciamento de memória manual
- Mesmo em linguagens que não exigem gerenciamento manual, é importante conhecer as restrições e capacidades do gerenciador de memória
- Esse conhecimento é fundamental para uma programação eficaz

Monoprogramação: A Técnica de Overlay¹⁰

- Técnica utilizada em sistemas monoprogramáveis como o MS-DOS
- Consiste em dividir um programa em módulos
 - Módulos de código ou dados
- Esses módulos utilizam a mesma área da memória principal
- Um módulo só é sobreposto quando o anterior não é mais necessário
- Era uma técnica definida pelo próprio programador

Problema da Relocação: Exemplo e Proteção

11

- Relocação é a transferência de código para que um programa não escreva na área de outro
 - Necessária pois processos executam em diferentes endereços físicos
- Exemplo do problema: Um programa (programa 2) com uma instrução de salto pode invadir a área de memória de outro programa (programa 1) se os endereços não forem corretamente ajustados após a relocação
- A relocação durante a carga do programa não resolve o problema da proteção
- A falta de proteção permite que aplicações mal-intencionadas acessem e desviem o controle

Registradores Base e Limite: Solução Hardware

12

- Os problemas de relocação e proteção podem ser resolvidos com recursos do processador (hardware)
- Utilizam-se dois registradores no processador: Base e Limite
 - Registradores são memória rápida, cara e localizada no processador
- O registrador-base é carregado com o endereço de início da partição alocada ao processo
- O registrador-limite é carregado com o tamanho da partição alocada ao processo
- Os endereços gerados durante a execução do processo são verificados no registrador-limite para garantir que fiquem dentro da área alocada, protegendo outras áreas

Classes de gerenciamento de memória

- 2 classes
 - Sistemas que durante o processamento levam e trazem a informação da memória para o disco
 - Troca de processos e paginação
 - Sistemas que não o fazem
- A troca de processos (swapping) carrega todo o programa para a memória principal, o executa por um determinado tempo e depois o mesmo retorna para o disco
- A paginação divide a memória em partições para a execução das aplicações

Monoprogramação sem troca de processos ou paginação

- Somente um programa é executado por vez, e a memória é compartilhada entre o sistema operacional e o programa
- 3 formas:
 - O SO está utilizando o espaço de endereçamento em RAM
 - Modelo aplicado aos mainframes e minicomputadores
 - O SO está utilizando o espaço de endereçamento em ROM/Flash somente para a leitura
 - Usado em alguns computadores de mão e em sistemas embarcados
 - O sistema operacional, os drivers e os aplicativos são armazenados na memória Flash e executados a partir da RAM
 - Ex.: Android (usa a Flash para armazenamento secundário)

- O usuário tem o controle da memória toda, podendo acessar até mesmo a área do sistema operacional
- Somente um usuário tem acesso aos recursos computacionais
- Na técnica de *overlay* o programa é dividido em módulos, e cada um é executado por vez na memória

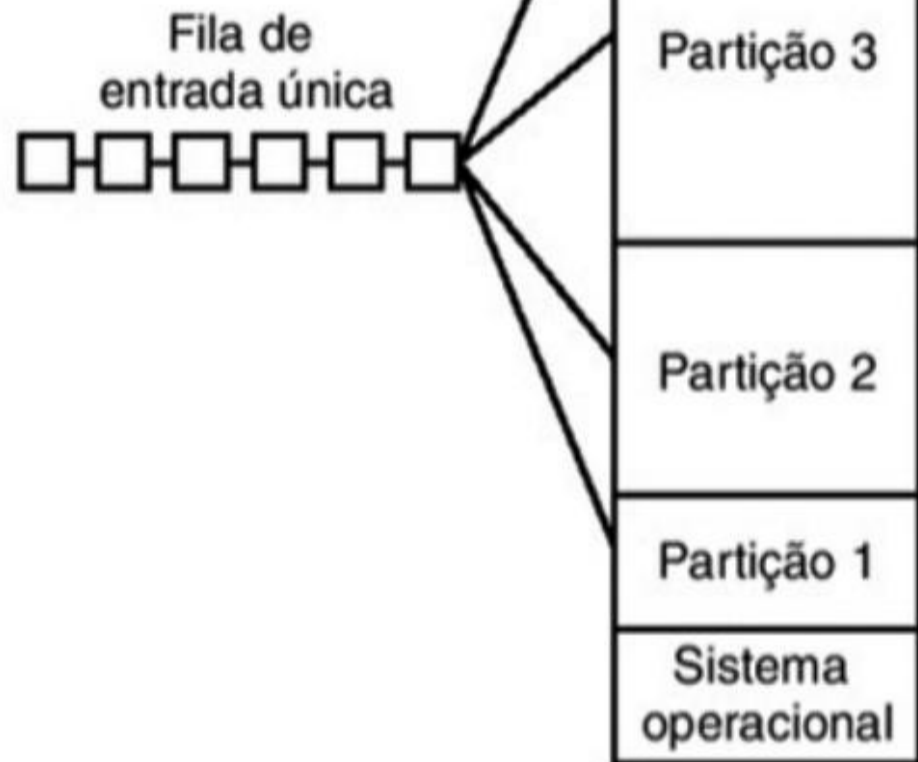
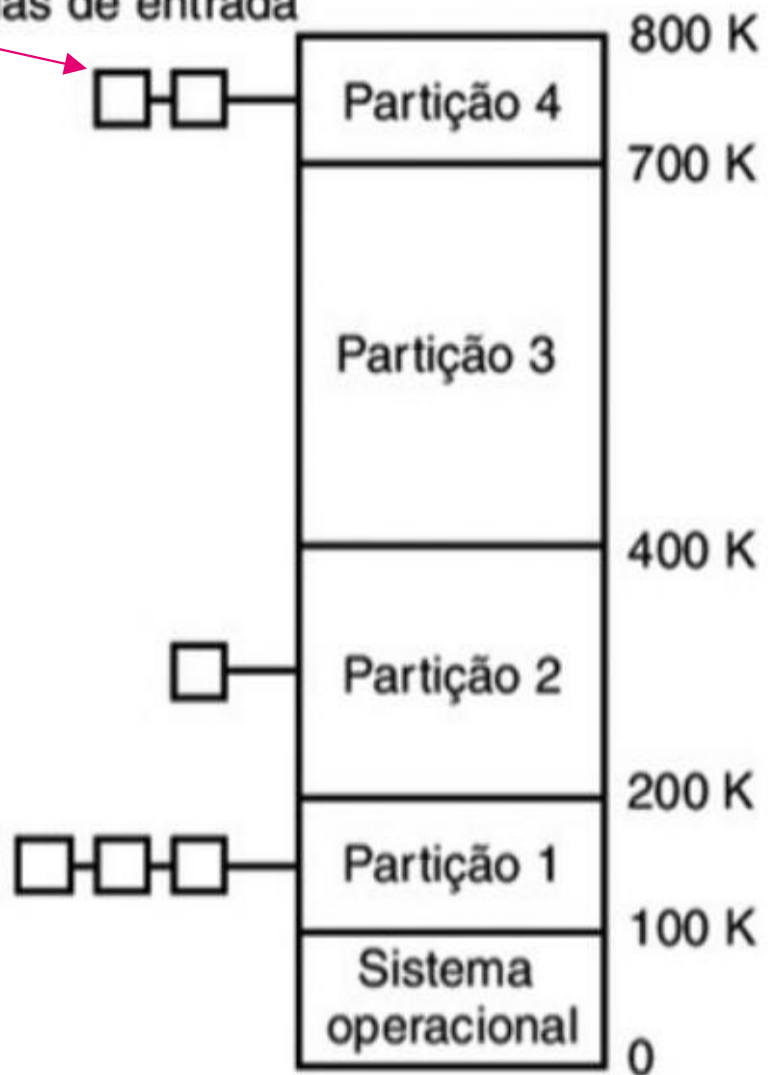
Multiprogramação com partições fixas

- A maioria dos SOs modernos permite que vários processos executem ao mesmo tempo (multiprogramação)
- Quando um processo é bloqueado aguardando uma informação de entrada/ saída
 - Outro processo poderá utilizar a CPU, aumentando a sua utilização
- A memória é dividida em **n** partições de tamanhos diferentes, podendo ser definida quando o sistema for iniciado
- Quando um processo chega para ser executado, ele é inserido em uma fila associada à menor partição suficiente para armazená-lo

Partições Fixas: Gerenciamento de Filas

- Na multiprogramação com partições fixas, a memória é dividida em partições de tamanhos definidos
- Um problema com múltiplas filas de entrada (uma por partição) é que algumas filas podem ficar cheias e outras vazias
- Uma solução é implementar uma fila única
 - Um processo mais próximo do início da fila que caiba na partição é carregado
 - Buscar o maior processo que caiba pode impedir que processos menores sejam selecionados (desperdício pode ocorrer)
 - Uma alternativa para processos pequenos é limitar o tempo máximo de espera por uma partição

Múltiplas filas de entrada



Endereços de
memória

Problemas da multiprogramação: relocação e proteção

- Relocação

- Transferência de um código de um local para outro de forma que um programa não escreva na área de outro programa
 - Necessária uma vez que processos executam em diferentes endereços na memória física

- Linker

- Liga programa principal, auxiliares, e bibliotecas
- Precisa saber o endereço inicial para relocar as instruções

- Uma implementação é relocar ao ser carregado

- O executável deve ter uma tabela de endereços

- A relocação durante a carga não resolve o problema da proteção
 - Aplicações não podem acessar outras partições
- Recursos do processador (hardware) podem resolver os problemas de relocação e proteção
 - Registradores do processador:
 - Base
 - Endereço de início da partição
 - Limite
 - Tamanho da partição