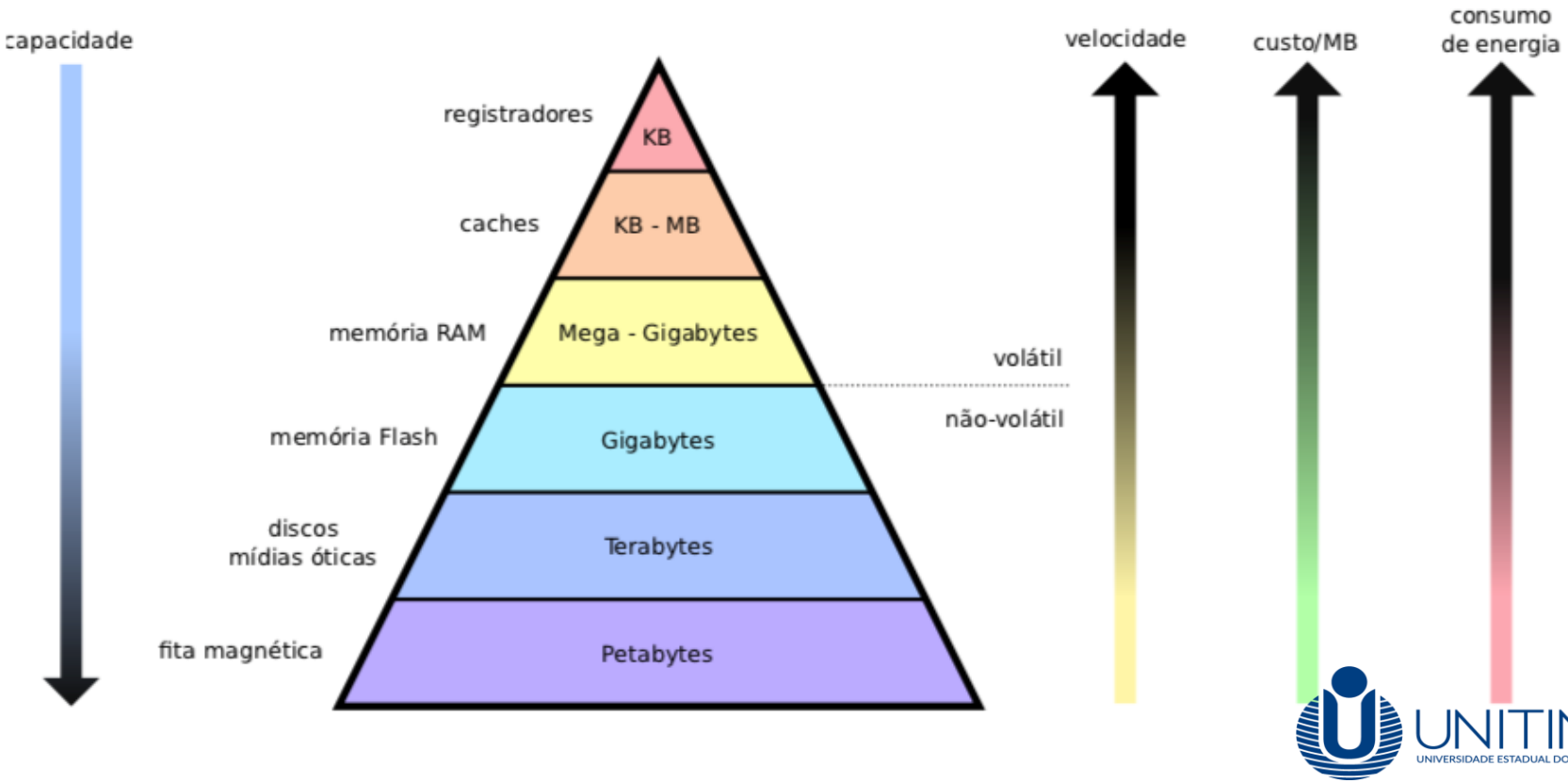
SISTEMAS OPERACIONAIS

Prof. Me. Napoleão Póvoa Ribeiro Filho

UNITINS

GERENCIAMENTO DE MEMÓRIA



UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

GERENCIAMENTO DE MEMÓRIA

- **Tempo de acesso**: tempo necessário para iniciar uma transferência de dados de/para um determinado meio de armazenamento.
- **Taxa de transferência**: indica quantos bytes por segundo podem ser lidos/escritos naquele meio, uma vez iniciada a transferência de dados.

GERENCIAMENTO DE MEMÓRIA

Meio	Tempo de acesso	Taxa de transferência
Cache L2	1 ns	1 GB/s (1 ns por byte)
Memória RAM	60 ns	1 GB/s (1 ns por byte)
Memória *flash* (NAND)	2 ms	10 MB/s (100 ns por byte)
Disco rígido SATA	5 ms (tempo para o ajuste da cabeça de leitura e a rotação do disco até o setor desejado)	100 MB/s (10 ns por byte)
DVD-ROM	de 100 ms a vários minutos (caso a gaveta do leitor esteja aberta ou o disco não esteja no leitor)	10 MB/s (100 ns por byte)

UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

MEMÓRIA FÍSICA

- A quantidade de memória RAM instalada, disponível em um computador.
- O conjunto de endereços de memória que um processador pode produzir é chamada de espaço de endereçamento.
- O espaço de endereçamento do processador é independente da quantidade de memória RAM disponível no sistema, podendo ser muito maior que esta.

ESPAÇO DE ENDEREÇAMENTO

- O processador acessa a memória RAM através de barramentos de dados, de endereços e de controle.
- O barramento de endereços (como os demais) possui um
- número fixo de vias, que define a quantidade total de endereços de memória que podem ser gerados pelo processador

UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

ESPAÇO DE ENDEREÇAMENTO

- Por exemplo, um processador Intel 80386 possui 32 vias de endereços, o que o permite acessar até 2^{32} bytes (4 GBytes) de memória, no intervalo[0... 2^{32} -1]
- Já um processador Intel Core i7 usa 48 vias para endereços e portanto pode endereçar até 2^{48} bytes, ou seja, 256 Terabytes de memória física.

UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

MEMÓRIA VIRTUAL

Para ocultar a organização complexa da memória física e simplificar os procedimentos de alocação da memória aos processos, os sistemas de computação modernos implementam a noção de memória virtual, na qual existem dois tipos de endereços de memória distintos: endereços físicos e endereços lógicos.

UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

MEMÓRIA VIRTUAL

Endereços físicos (ou reais) são os endereços dos bytes de memória física do computador. Estes endereços são definidos pela quantidade de memória disponível na máquina.

Endereços lógicos (ou virtuais) são os endereços de memória usados pelos processos e pelo sistema operacional e, portanto, usados pelo processador durante a execução. Estes endereços são definidos de acordo com o espaço de endereçamento do processador.

MEMÓRIA VIRTUAL

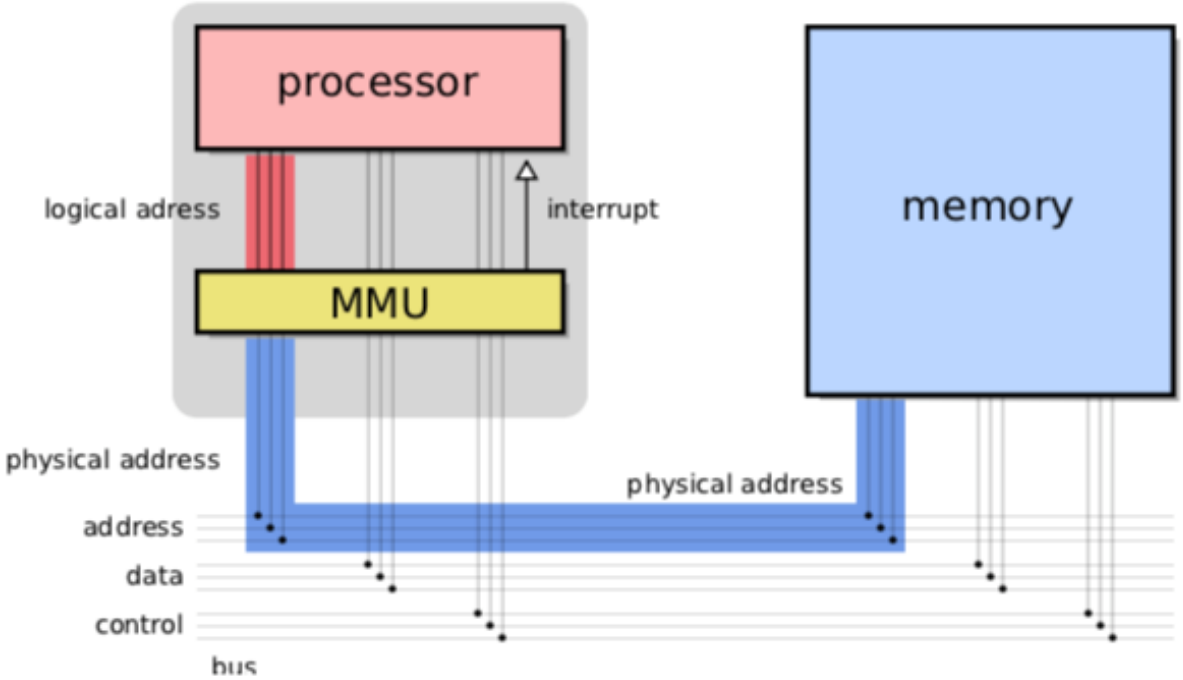
- Ao executar, os processos “enxergam” somente a memória virtual.
- Assim, durante a execução de um programa, o processador gera endereços lógicos para acessar a memória.
- Esses endereços devem então ser traduzidos para os endereços físicos correspondentes na memória RAM, onde as informações desejadas se encontram.

MEMÓRIA VIRTUAL

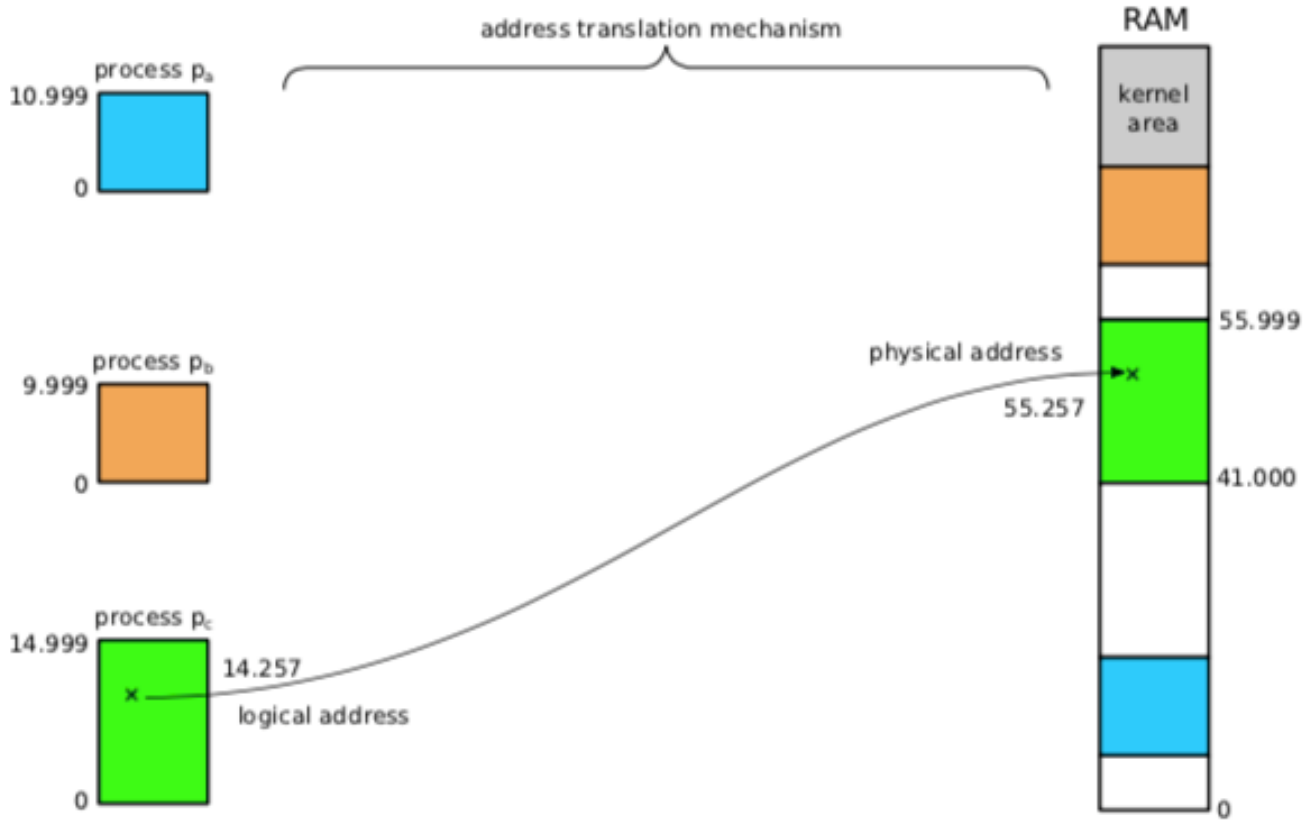
- Por questões de desempenho, a tradução de endereços lógicos em físicos é feita por um componente específico do hardware do computador, denominado Unidade de Gerência de Memória (MMU – *Memory Management Unit*).

- Na maioria dos processadores atuais, a MMU se encontra integrada ao chip da própria CPU.

MMU

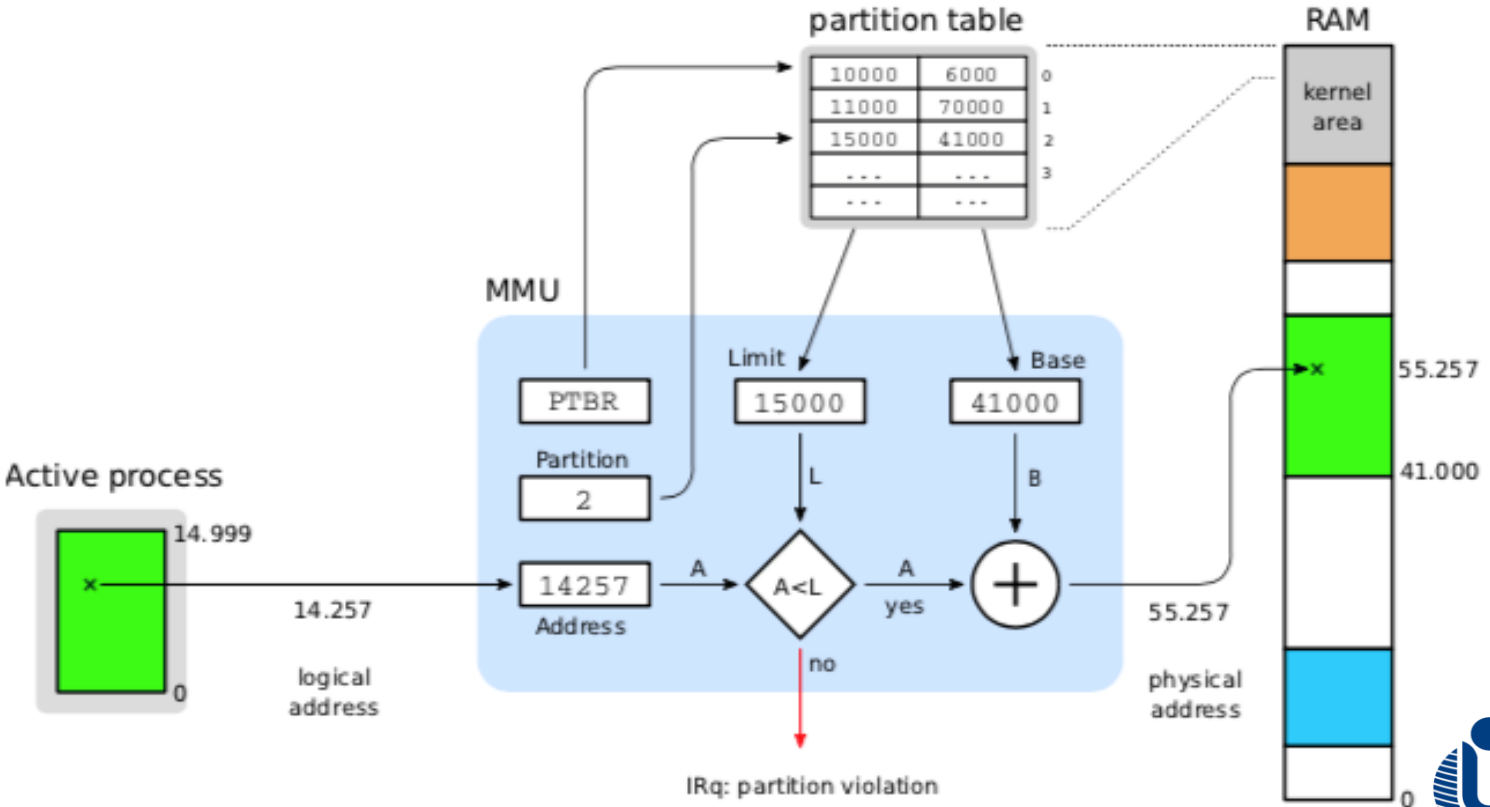


TRADUÇÃO DE ENDEREÇO



UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

TRADUÇÃO DE ENDEREÇO



GERÊNCIA DE MEMÓRIA

O gerente de memória é um componente do SO que aloca memória principal para os processos e gerencia a hierarquia de memórias.

Suas tarefas são:

- garantir o isolamento mútuo entre processos
- manter o registro das áreas de memória em uso
- alocar memória RAM para novos processos
- fazer swapping transparente entre memória principal e disco

GERÊNCIA DE MEMÓRIA

- atender requisições de aumento de memória
- manter o mapeamento de memória virtual para memória física
- implementar políticas de alocação de memória para os processos

CLASSES DE MMU

- Aqueles que movem processos entre memória principal e secundária durante a execução(utilizando técnicas com o swapping e paginação)

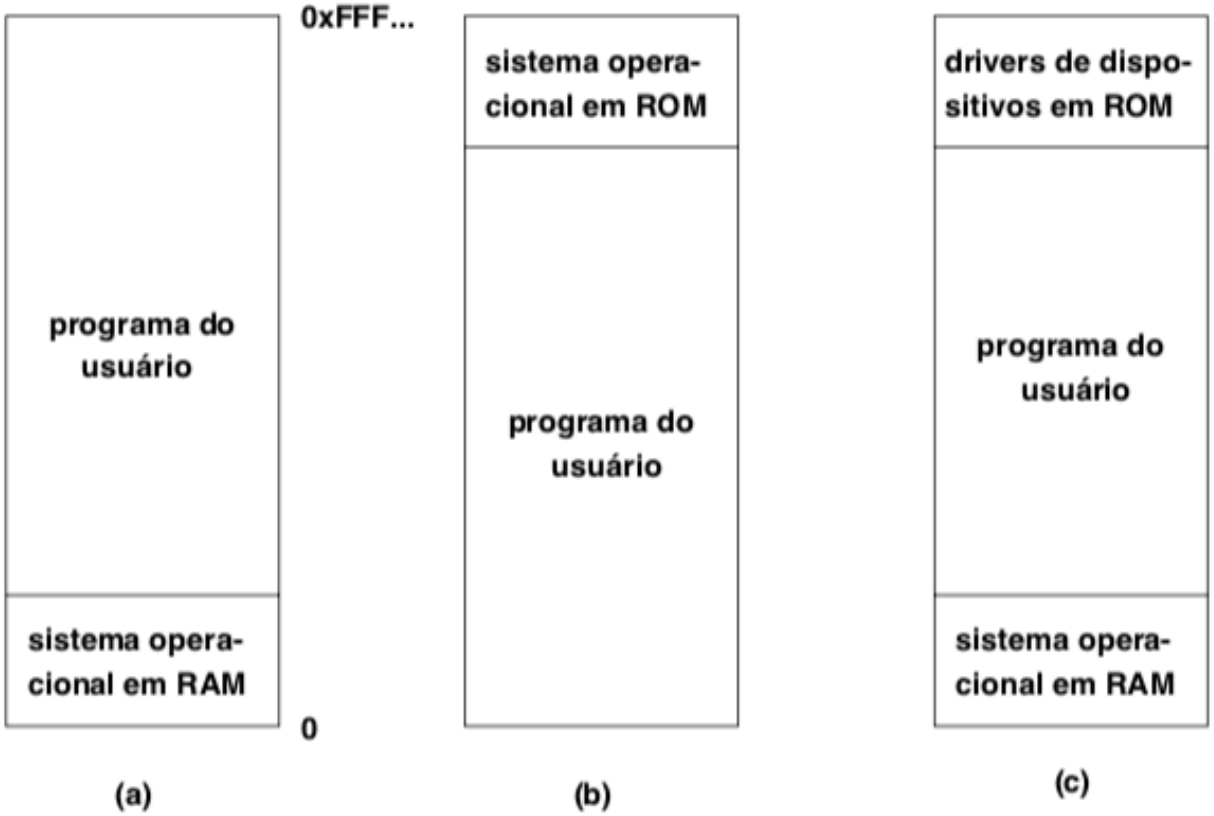
- Aqueles que mantêm os processos fixos em memória primária

MONOPROGRAMAÇÃO

- Consiste em ter somente um processo na memória durante toda a sua execução.
- Caso a memória seja insuficiente, o programa tem sua execução rejeitada (overlap).
- Quando um processo termina sua execução, o SO reassume a CPU e espera por um novo comando para carregar outro processo na memória já liberada pelo primeiro.

UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

MONOPROGRAMAÇÃO



UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

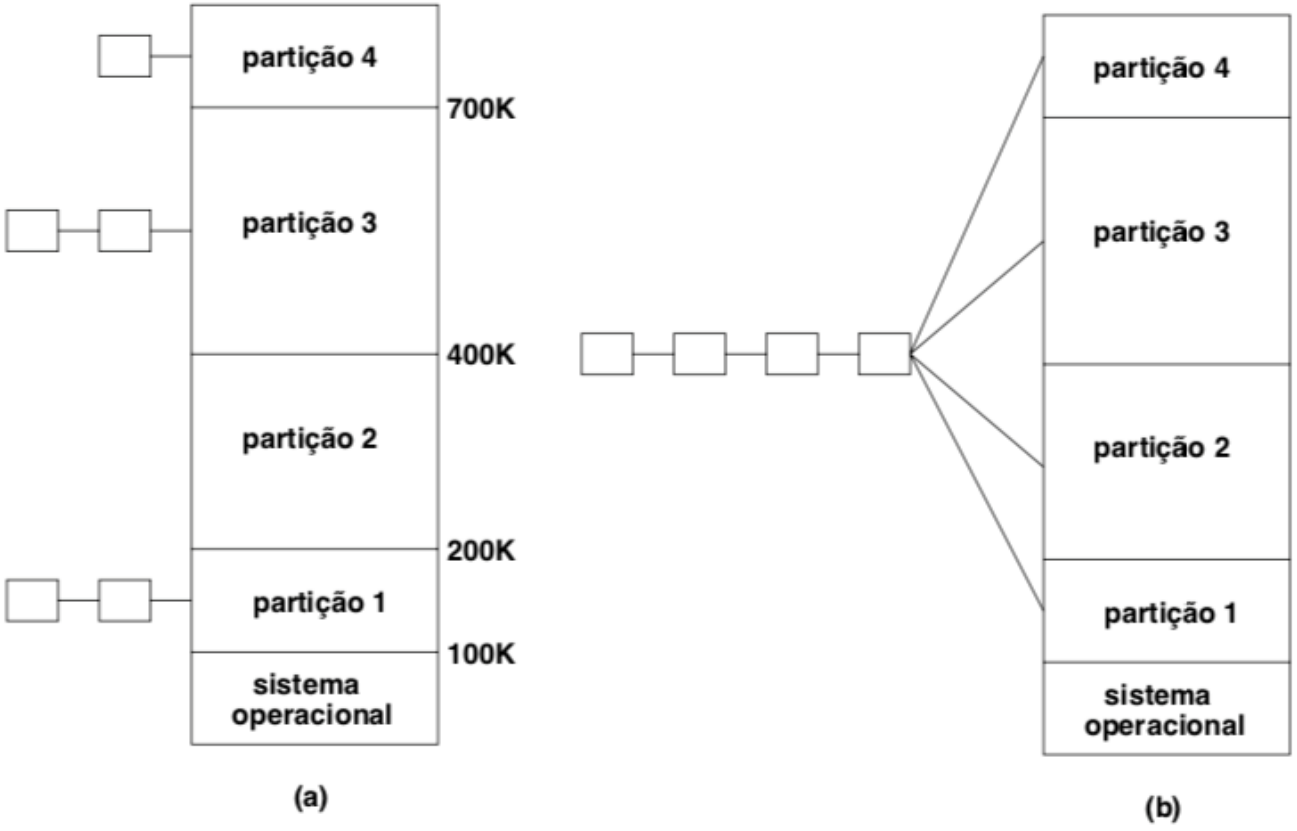
MULTIPROGRAMAÇÃO

- Adequado para serviços interativos para vários usuários simultaneamente
- Permitir um melhor uso da CPU principalmente quando processos gastarem muito tempo executando E/S.

PARTIÇÕES FIXAS

- Consiste em dividir a memória em *n* partições.
- Essas partições podem ser estabelecidas na configuração do SO
- Quando um processo se inicia, ele é colocado na fila de entrada para ocupar a menor partição de tamanho suficiente para acomodá-lo.
- Com partições fixas, qualquer espaço não usado pelo processo é perdido.

PARTIÇÕES FIXAS



PARTIÇÕES FIXAS

- Esse sistema foi usado pelo OS/360 nos grandes mainframes da IBM por muitos anos.
- Era chamado de MFT (*Multiprograming with Fixed number of Task*)
- Os processos que chegam são colocados em uma fila até que uma partição adequada seja liberada, quando então são carregados e executados.

REALOCAÇÃO E PROTEÇÃO

Multiprogramação introduz dois problemas essenciais que devem ser resolvidos:

- Realocação: não se sabe de antemão em qual região da memória o processo vai ser executado (endereço de variáveis e do código não podem ser absolutos)

- Proteção: evitar que um processo acesse a região usada por outro processo

REALOCAÇÃO E PROTEÇÃO

- Melhor solução: mapeamento para memória física ocorre em tempo de execução e é relativa a dois registradores: base e limite

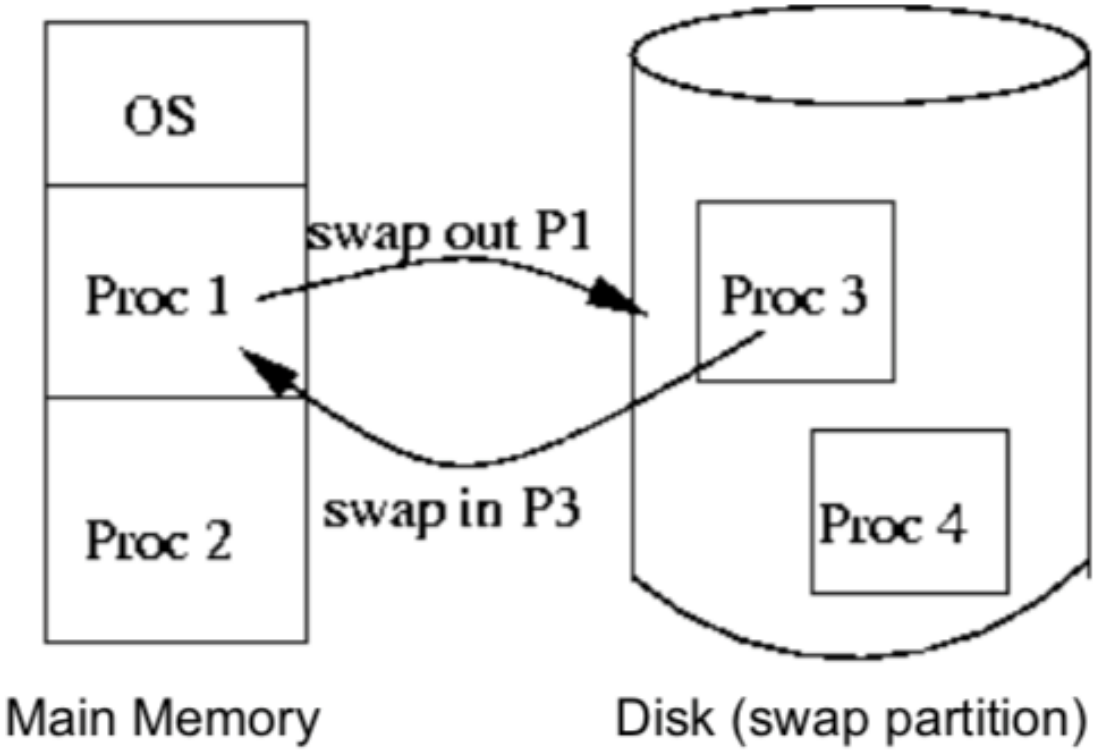
- Qualquer acesso à memória fora desses limites é considerado erro e o processo é abortado

SWAPPING

- Em sistemas de tempo compartilhado, a memória principal pode não ser suficiente para todos os processos (Ex.: muitos processos interativos orientados à E/S de muitos usuários)

- Ideia básica: usar o espaço em disco como extensão da memória RAM, e colocar lá os processos enquanto estiverem bloqueados, carregando-os de volta para a memória assim que são desbloqueados.

SWAPPING



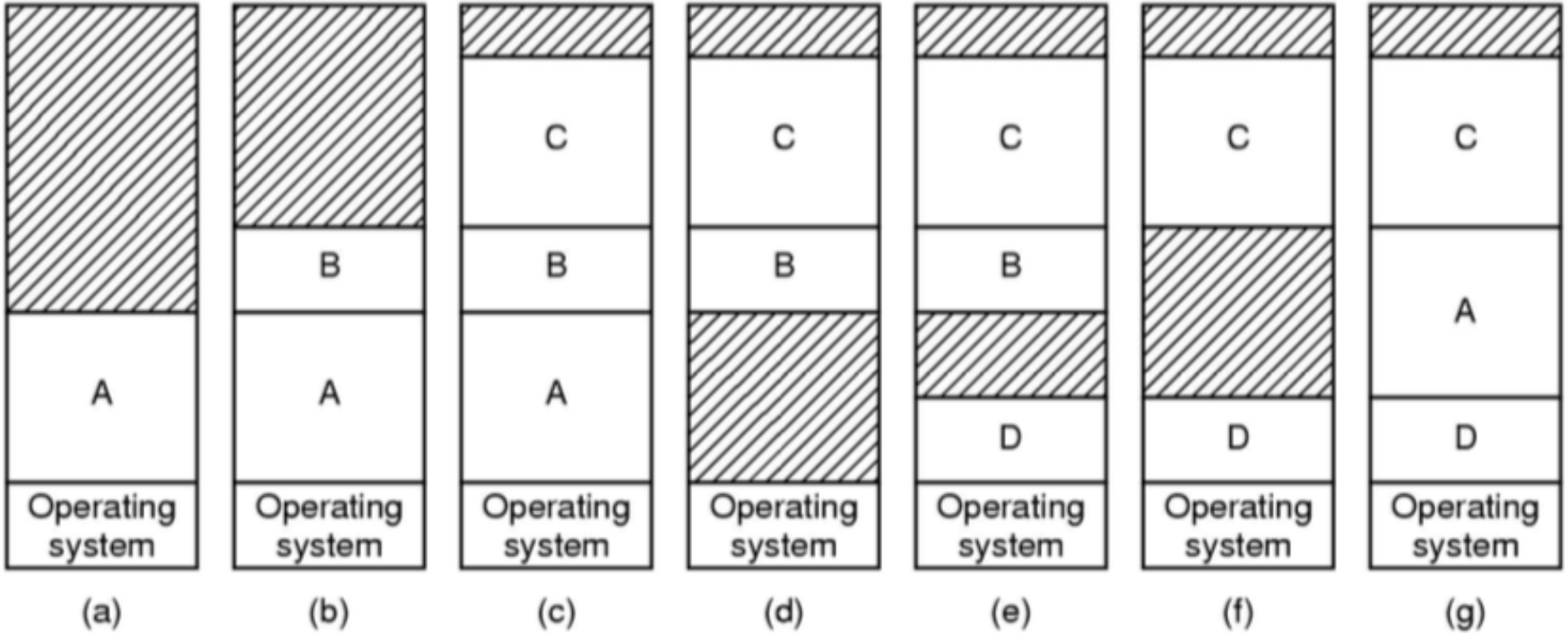
SWAPPING

- Para realizar o swapping, existem duas alternativas:
- copiar a imagem inteira (swapping)
- permitir que o processo fique parcialmente em memória, e parcialmente em disco (paginação) - memória virtual

SWAPPING

- Quando um processo é bloqueado, ele pode ser *swapped out*, e depois *swapped in* para a memória principal.
- Essa técnica permite manter um número maior de processos ativos, aumentando a utilização da CPU
- Podem surgir regiões de memória não utilizadas de qualquer tamanho
- Um mesmo processo pode ocupar diferentes partições ao longo de sua execução

SWAPPING



UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

SWAPPING

- Principal problema do swapping com partições de tamanho variável:

- manter a informação sobre espaços não utilizados

- evitar uma fragmentação externa da memória (espaços pequenos não utilizados)

SWAPPING

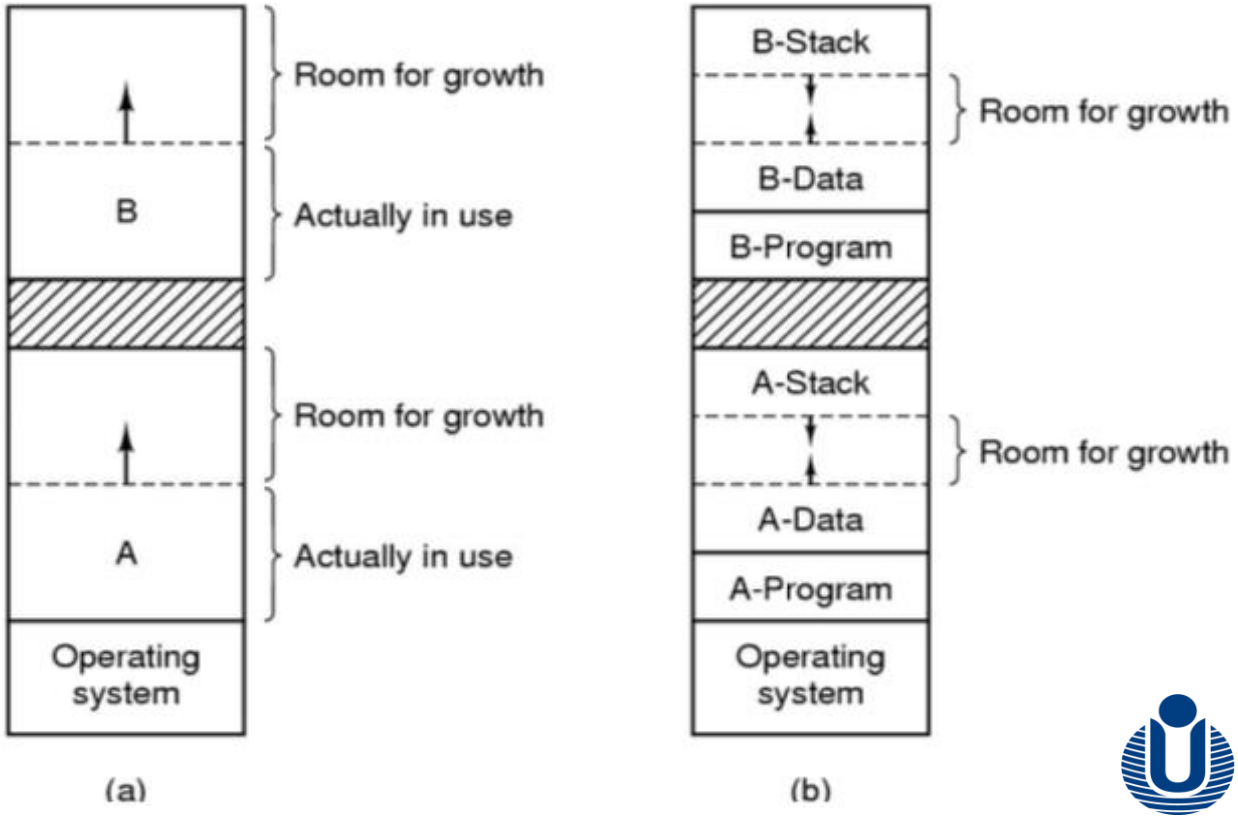
- Fragmentação: parcela da memória desperdiçada devido a imagens de processos não diretamente adjacentes
- Fragmentação interna: quando há partições fixas
- Fragmentação externa: quando não há partições

SWAPPING

Também é usado para lidar com processos que aumentam sua demanda de memória. Para esse caso, existe algumas alternativas

a) alocar uma partição para o processo que é adjacente a uma partição livre
b) alocar uma partição conjunta para a pilha e o heap, e fazê-los crescer em sentidos opostos
c) realizar um swap-out e um swap-in em uma partição maior

SWAPPING



UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

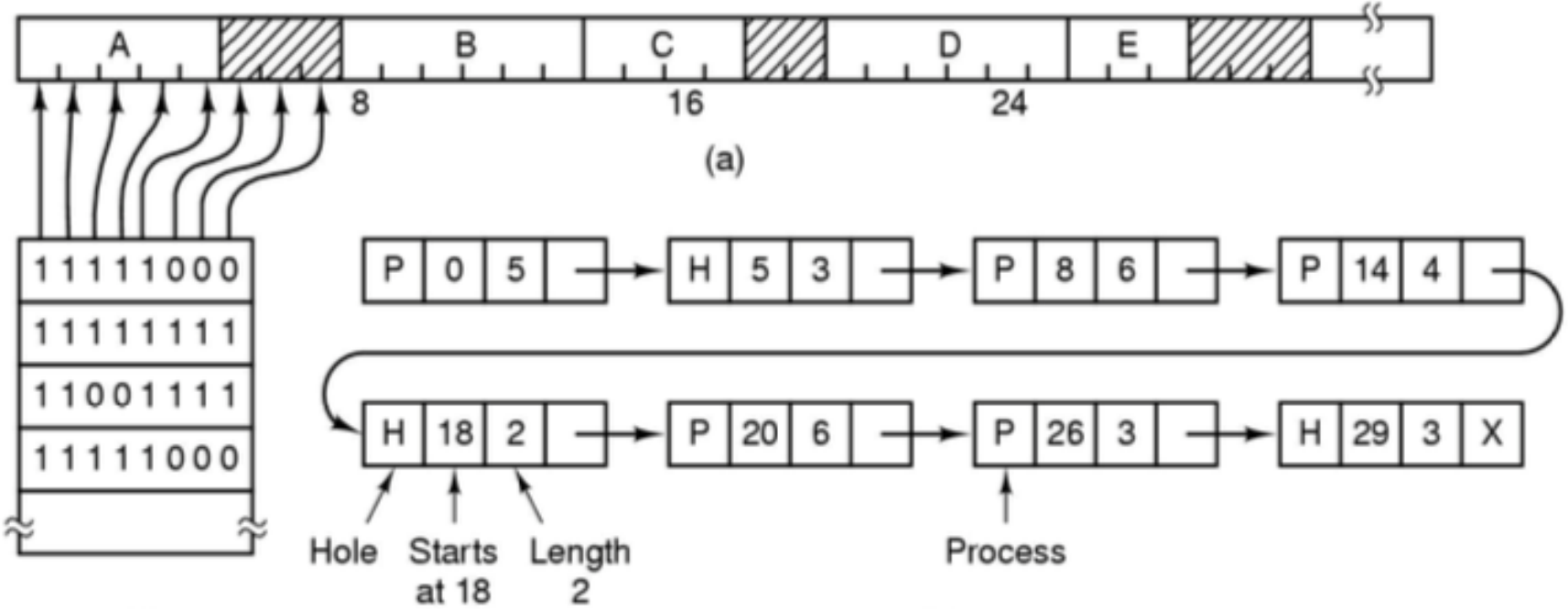
GERENCIAMENTO DE ESPAÇOS LIVRES

- Divide a memória em unidades de alocação de n bytes e representa a ocupação (livres/ocupados) de lotes de unidades usando um bit map ou uma lista encadeada

- Bit Map: armazenamento compacto e simples, mas busca por determinado tamanho de lote pode envolver análise de várias palavras (no bit map)

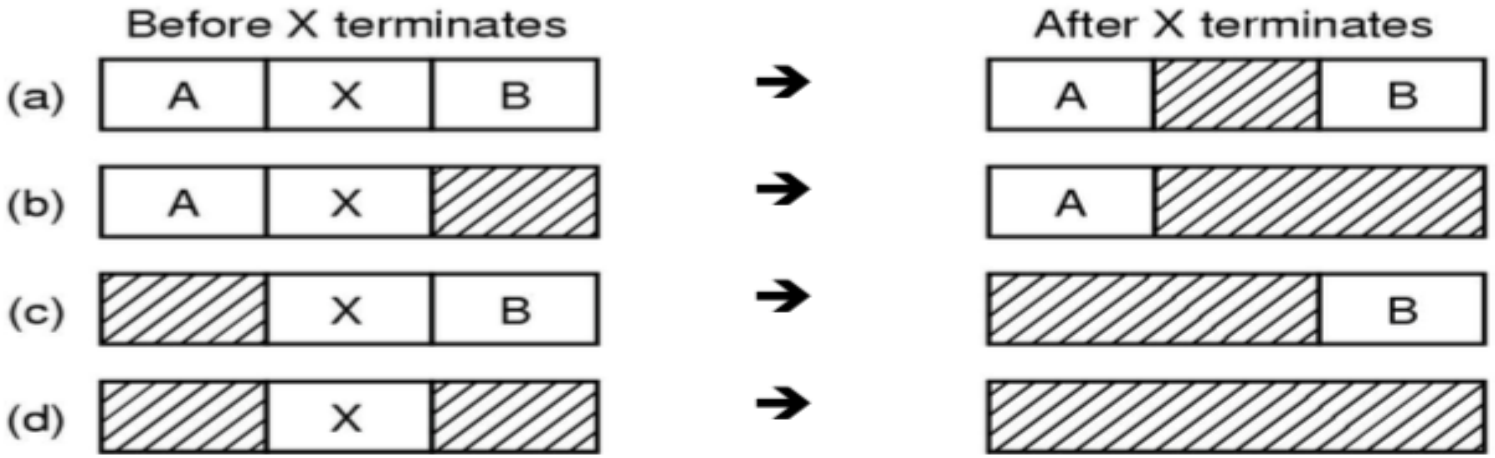
- Lista encadeada: cada nó contém o endereço inicial e o tamanho de uma partição ocupada ou livre

GERENCIAMENTO DE ESPAÇOS LIVRES



GERENCIAMENTO DE MEMÓRIA COM LISTAS

- Quando um processo é swapped out, a sua lacuna precisa ser combinada com espaços vizinhos livres
- Quando o processo é swapped in, percorre-se a lista buscando um espaço livre suficientemente grande



UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

GERENCIAMENTO DE MEMÓRIA COM LISTAS

Algoritmos para escolha de espaço livre:

- First Fit - aloca o primeiro espaço encontrado

- Best Fit - percorre toda a lista e aloca o menor possível espaço (pode deixar fragmentos muito pequenos)

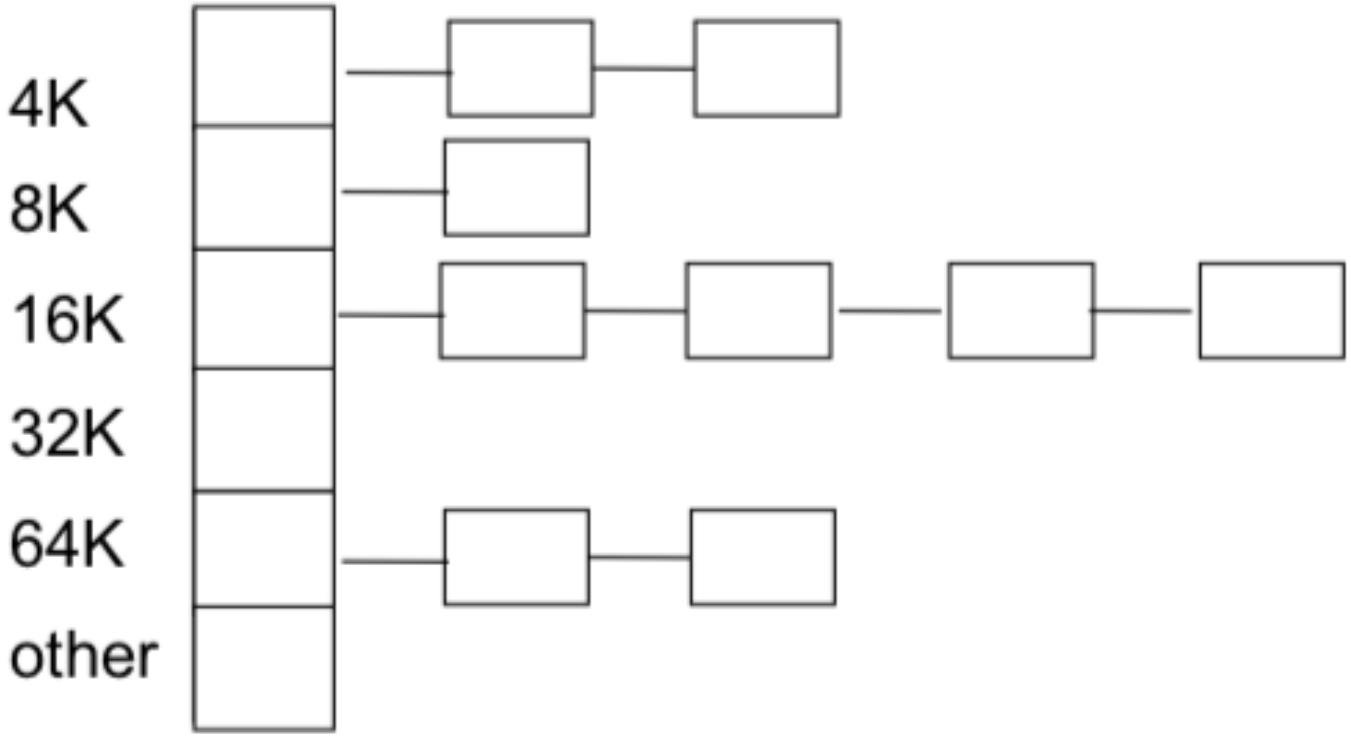
- Worst Fit - percorre toda a lista e aloca o maior possível espaço

GERENCIAMENTO DE MEMÓRIA COM LISTAS

Alternativa, para isso seria o Quick Fit

- Quick Fit: mantem listas separadas por tamanho do espaço livre
- Problema do Quick Fit: ao liberar memória, o novo espaço criado precisa ser inserido na fila correspondente (possivelmente após combinação com áreas vizinhas)

GERENCIAMENTO DE MEMÓRIA COM LISTAS



UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS

MEMÓRIA VIRTUAL

- É necessária quando o total de memória necessária para um conjunto de processos excede o tamanho da memória física. Também aqui, usa-se parte do disco como extensão da memória RAM.

- Usa a técnica de paginação

MEMÓRIA VIRTUAL

- Espaço de endereçamento lógico de cada processo e memória física são divididos em partições de mesmo tamanho fixo (páginas ou quadro de páginas)

- Em vez de fazer swap in/out de uma imagem inteira de processo, cada página pode ser movida do disco para a memória e vice-versa

MEMÓRIA VIRTUAL

- Em cada acesso a memória, o hardware verifica se a página correspondente está na memória. E se não for o caso, então um tratador de páginas faltantes (page-fault) a copia para lá.

- A espera pela cópia para a memória de uma página é equivalente a um bloqueio por E/S

PAGINAÇÃO

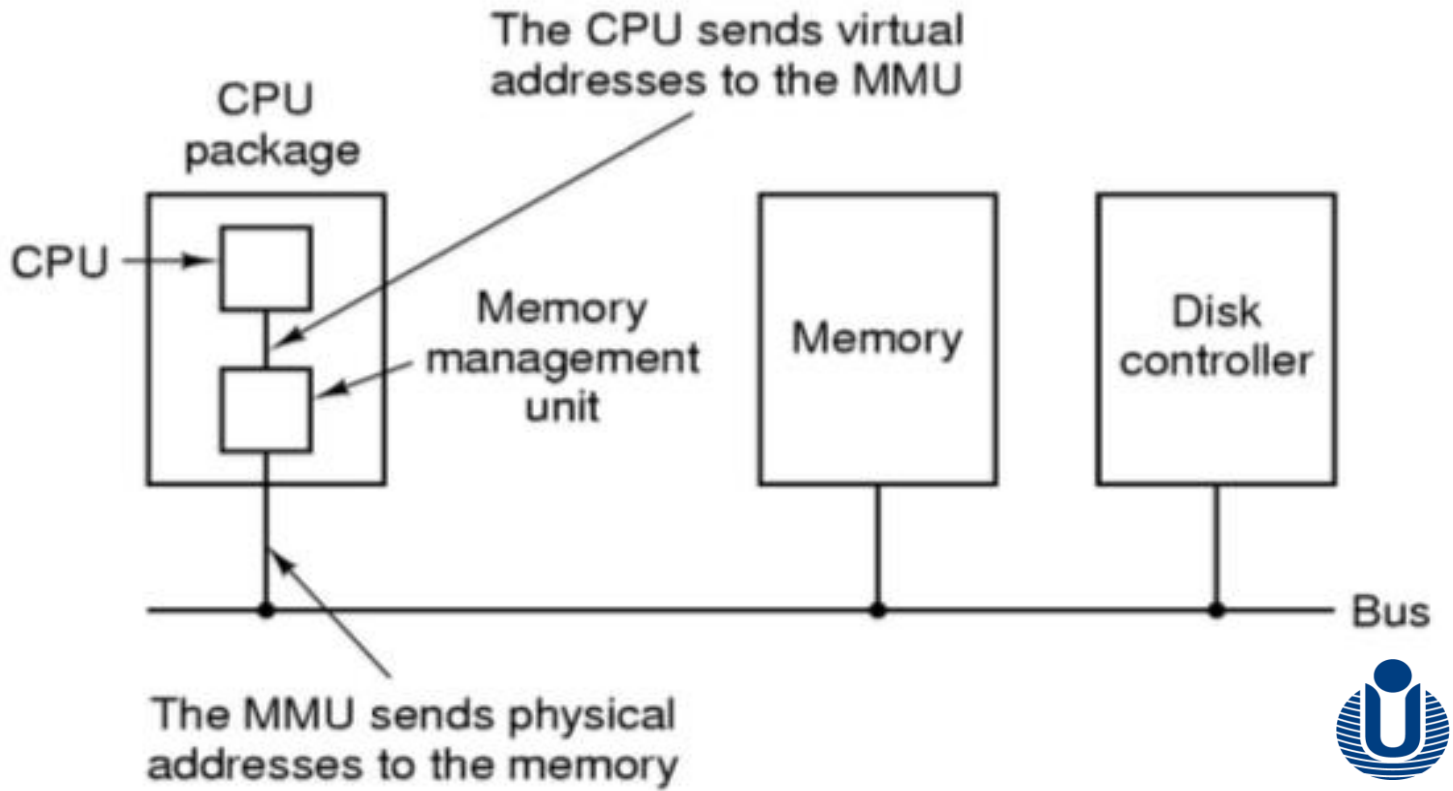
Requer a existência de suporte por hardware (MMU)

- MMU intercepta qualquer acesso à memória

- Mapeia endereços lógicos para físicos (através de uma tabela de páginas)

- Quando a página não está na memória, gera uma interrupção causando a interrupção (page fault) do processo em execução e o seu bloqueio até que a página tenha sido copiada

PAGINAÇÃO



UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS


UNITINS
UNIVERSIDADE ESTADUAL DO TOCANTINS