Curso de Engenharia de Computação
Sistemas Operacionais

INSTITUTO MAUÁ DE TECNOLOGIA
MAUÁ

Gerenciamento de Memória – Parte I

Abstração de Memória e Memória Virtual

Slides da disciplina Sistemas Operacionais
Curso de Engenharia de Computação
Instituto Mauá de Tecnologia – Escola de Engenharia Mauá
Prof. Marco Antonio Furlan de Souza

Gerenciamento de Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Uso de memória em sistemas operacionais**
  - **Lei de Parkinson** (humanos):
    - *O trabalho se expande de modo a preencher o tempo disponível para a sua realização.*
  - **Lei do uso de memória em sistemas operacionais**
    - *Os programas se expandem para preencher a memória disponível para mantê-los*.

Gerenciamento de Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Conceitos**
  - **Memória é finita e hierárquica**
    - Desejo dos programadores – **memória disponível infinita** – **não** é **possível**;
    - Existem **diversos níveis** de **memória**: **registradores** da CPU, **caches** e **RAM**;
    - Quem gerencia a memória em um sistema operacional é um processo denominado de **gerenciador de memória** que executa as seguintes tarefas genéricas:
      - **Rastrear** quais **partes** da **memória** estão em **uso**;
      - **Alocar memória** para os processos quando eles precisarem;
      - **Desalocar memória** quando processos forem concluídos.

Gerenciamento de Memória

INSTITUTO MAUÁ DE TECNOLOGIA

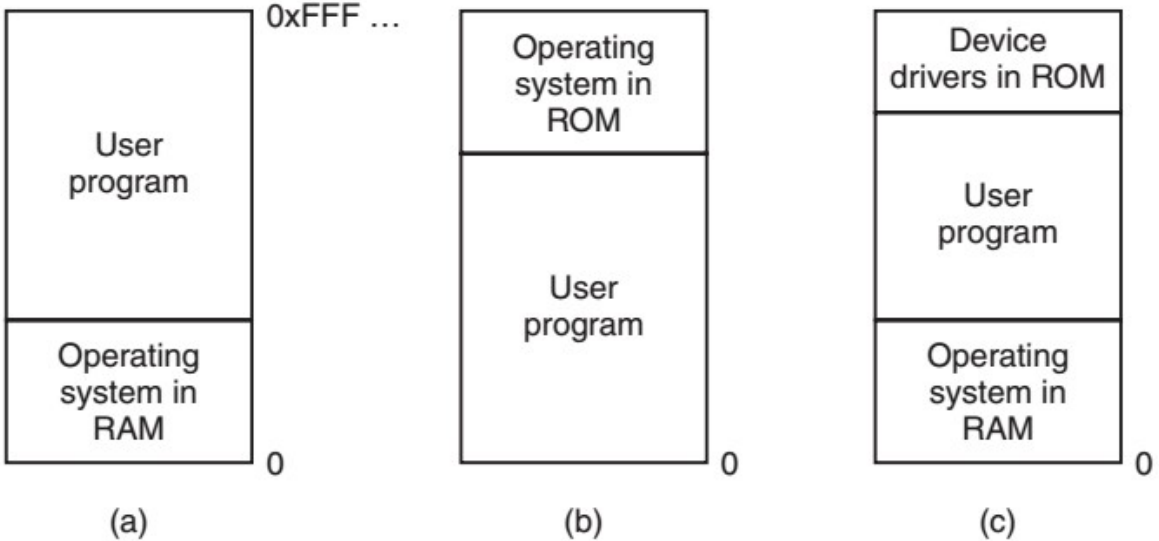
- **Abstração de memória**
  - **Sem abstração de memória**
    - **Antigos computadores** e **arquiteturas simples** (como em alguns sistemas embarcados) não possuíam sistemas operacionais com o **conceito** de **abstração** de **memória**;
    - Neste caso, o **programa acessa diretamente** a **memória física**;
    - Assim, **em** uma **instrução como** `MOV REGISTER1,1000`, ocorre a **cópia** do **conteúdo** da localização da **memória** física `1000` para `REGISTER1`;
    - Nessas condições, **não era possível ter dois programas em execução na memória ao mesmo tempo**!
    - Isso é devido à possível sobreposição de escrita na memória – leva à falha na execução dos programas.

Gerenciamento de Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Abstração de memória**
  - **Sem abstração de memória**

Opções de leiaute
(a) Sistema operacional na parte inferior da memória na RAM;
(b) Sistema Operacional na ROM no topo da memória;
(c) Drivers de dispositivo no topo da memória em uma ROM e o resto do sistema em RAM
O único modo de se obter **paralelismo** é implementar threads dentro do único programa – mas é uma **solução** extremamente **limitada**.



Gerenciamento de Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Abstração de memória**
  - **Sem abstração de memória**
    - Para **executar vários programas sem abstração** de **memória**, o que o **sistema operacional** precisa fazer é **salvar todo** o **conteúdo** da memória do **programa atualmente** em **execução** em um **arquivo** de **disco**, e **depois**, **inserir** e **executar** o outro **programa**;
    - Mas pode-se manter **diversos programas** em **execução** na **memória** com a **adição** de um **hardware especial**;
    - Por **exemplo**, o **IBM360** introduziu um sistema em que a memória era **dividida** em **blocos** de **2KB** cada e foi atribuída uma **chave de proteção** de **4 bits** mantida em **registradores especiais** dentro da **CPU** para referenciar esses blocos;
    - Assim, uma máquina com **1MB** de **memória precisava** apenas de **512 desses registros** de **4 bits** (512 chaves, 512 x 2KB = 1024 MB, 1 chave por bloco) para um total de 256 bytes (512 x 4 bits x 1 byte / 8 bits = 256 bytes) de armazenamento de chaves.

Gerenciamento de Memória

INSTITUTO MAUÁ DE TECNOLOGIA

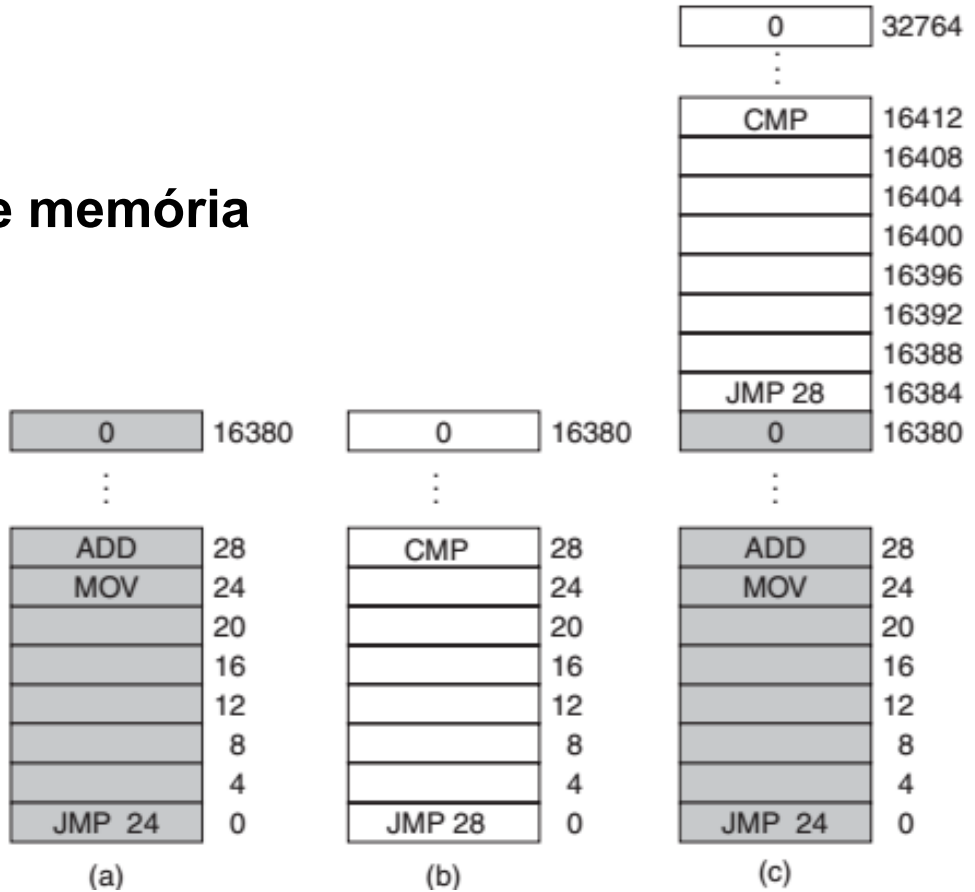
- **Abstração de memória**
  - **Sem abstração de memória**
    - Um registrador denominado **PSW** (*Program Status Word*) também continha uma **chave** de **4 bits**;
    - O hardware do IBM 360 **interceptava** qualquer **tentativa** de um **processo** em execução para **acessar** a **memória** com um **código** de **proteção diferente** da **chave** presente no PSW (todo endereço de memória tem uma chave);
    - Assim, os **processos** do **usuário** eram **impedidos** de **interferir** uns **com** os **outros** e com o **próprio sistema operacional**;
    - Dessa forma, o **problema** de **acesso** à **memória** com vários programas em execução simultânea foi **resolvido**, mas ainda **persiste** o **problema** da **realocação**.

Gerenciamento de Memória

- **Abstração de memória**
  - **Sem abstração de memória**
    - **Problema da realocação de memória**

Em **(a)** e **(b)** tem-se dois processos que utilizam 16KB de memória para executar.

Quando ambos são armazenados na memória **(c)** , os valores de constantes de posição de memória (por exemplo, em `JMP`) precisam ser realocadas, senão referenciarão posições inválidas!



Gerenciamento de Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Abstração de memória**
  - **Sem abstração de memória**
    - **Problema da realocação de memória**
      - **Solução** do IBM 360: **realocação estática**;
      - Quando um **programa** era **carregado** no **endereço 16384**, a **constante 16384** era **adicionada** a cada **endereço** do programa **durante** o processo de **carregamento** (assim, `JMP 28` tornava-se `JMP 16412`, etc.);
      - Esta **não é uma solução muito geral** e **aumenta o tempo de carga do programa**;
      - Além disso, **requer informações extras** em todos os programas **executáveis** para indicar quais **palavras** contêm **endereços** (realocáveis) e quais não contêm.

Gerenciamento de Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Abstração de memória**
  - **Sem abstração de memória**
    - **Ainda existem sistemas assim?**
      - Dispositivos como **rádios**, **máquinas de lavar** e **fornos** de **microondas** estão repletos de software (em **ROM**) e, na maioria dos casos, o software utiliza a memória absoluta;
      - Isso **funciona porque** todos os **programas** são **conhecidos antecipadamente** e os **usuários não** estão **livres** para **executar** seu próprio software nesses dispositivos;
      - **Em alguns casos existe um sistema operacional**, mas é **apenas** uma **biblioteca** que está **vinculada** ao **programa aplicativo** e fornece **chamadas** do **sistema** para **executar** E/S e **outras tarefas comuns**.

Gerenciamento de Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Abstração de memória**
  - **Espaço de endereços**
    - Os **principais problemas** de **não** se **abstrair** a **memória** são proteção e realocação;
    - **Um espaço de endereço** é o **conjunto** de **endereços** que um **processo** pode **usar para endereçar memória**;
    - **Cada processo tem** seu **próprio espaço** de **endereço**, **independente daqueles pertencentes** a **outros processos** (exceto em **algumas circunstâncias especiais** em que os **processos** desejam **compartilhar** seus espaços de endereço);
    - Assim, o **endereço de memória** `28` em **dois programas distintos refere-se** normalmente a **dois endereços físicos distintos** também.

Gerenciamento de Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Abstração de memória**
  - **Espaço de endereços**
    - Quando se utiliza espaço de endereços, uma solução mais apropriada para realocação é utilizar um sistema denominado **realocação dinâmica**;
    - Ela **mapeia** o **espaço de endereços** de cada **processo em** uma **parte diferente** da **memória física** assim:
      - Utilizam-se **dois registradores**, geralmente **chamados** de **base** e **limite**;
      - **Quando** um **processo é executado**, o **registrador base** é **carregado** com o **endereço físico**, onde o **programa começa** na **memória**;
      - O **registrador limite** é **carregado** com o **comprimento do programa**. Se o programa tem um comprimento de 16KB, esse valor é 16384.

Gerenciamento de Memória

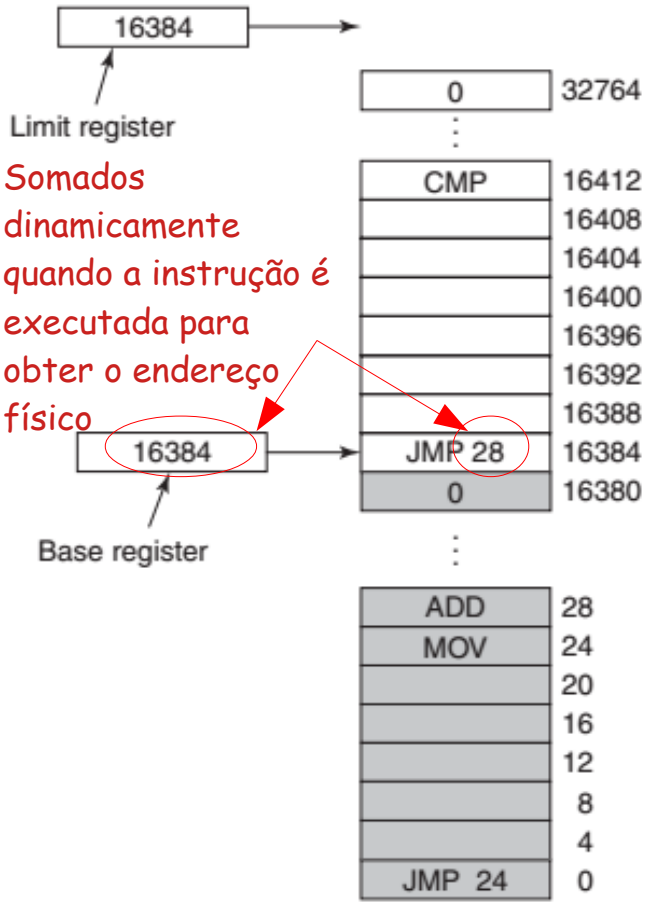
- **Abstração de memória**
  - **Espaço de endereços**
    - **Exemplo de realocação dinâmica**

Aqui tem-se dois programas;

O primeiro inicia na memória física na posição 0 e tem comprimento de 16KB.

O segundo incia na memória física na posição 16KB e também tem comprimento de 16KB.

Os registradores base e limite são ajustados com os valores que delimitam o espaço de endereço do programa que está em execução.



Gerenciamento de Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Abstração de memória**
  - **Espaço de endereços**
    - **Realocação dinâmica com base e limite**
      - **Toda vez que um processo faz referência à memória**, seja para **buscar** uma **instrução** ou **ler** ou **escrever** uma **palavra** de **dados**, o **hardware** da **CPU adiciona automaticamente** o **valor base** ao **endereço gerado** pelo **processo antes** de **enviar** o **endereço** para **fora**, no **barramento** de **memória**.
      - **Simultaneamente**, o hardware **verifica** se o **endereço** oferecido é **igual** ou **maior** do que o **valor** no **registrador** de **limite**, caso em que uma **falha** é **gerada** e o **acesso** é **abortado**.

Gerenciamento de Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Abstração de memória**
  - **Espaço de endereços**
    - **Realocação dinâmica com base e limite**
      - Uma **desvantagem** de **relocação** usando registradores de **base** e **limite** é a **necessidade** de realizar uma **adição** e uma **comparação** em cada **referência** de **memória**;
      - As **comparações** podem ser feitas **rapidamente**, mas as **adições** são mais **lentas**;
      - Se a **memória física** do computador for **grande** o **suficiente** para conter **todos** os **processos**, este tipo de solução funciona perfeitamente!
      - Mas, se não houver memória suficiente para todos os processos, outra solução precisa ser encontrada…

Gerenciamento da Memória

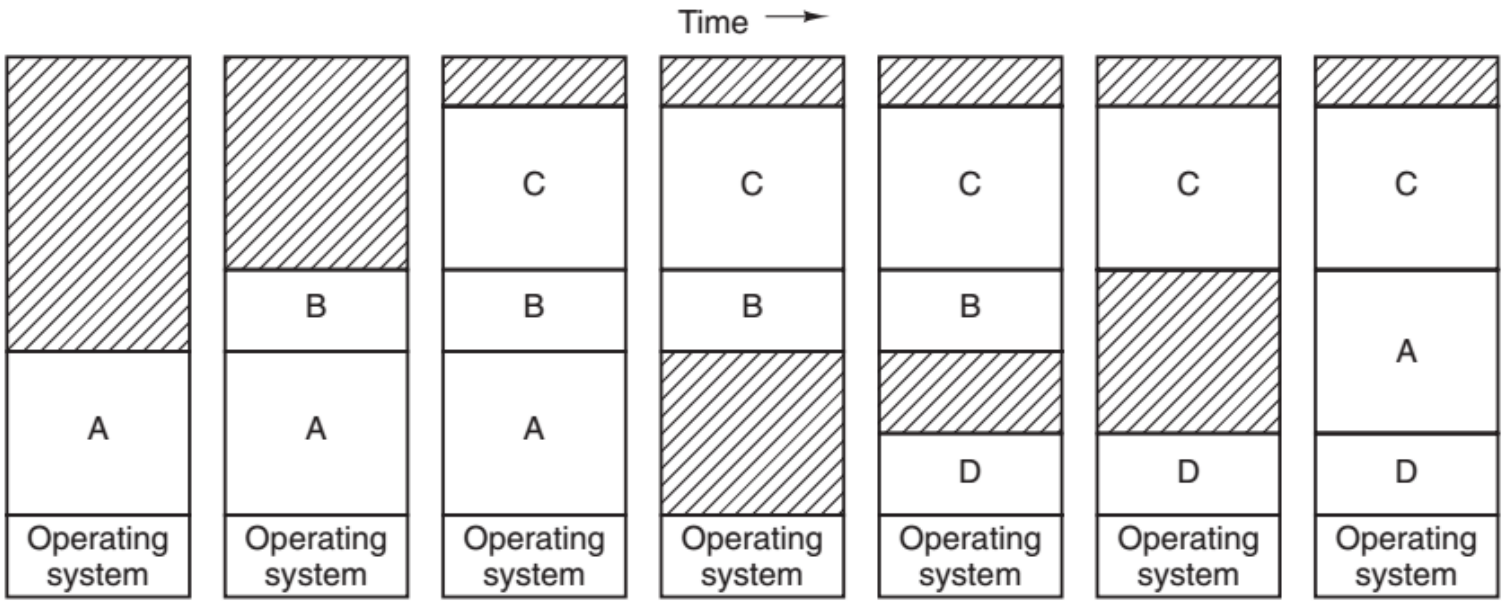
INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Manter todos** os **processos** na **memória** o **tempo todo requer** uma **grande quantidade** de memória e não pode ser feito se não houver memória suficiente;
  - **Duas abordagens gerais** para **lidar** com a **sobrecarga** de **memória**:
    - **Swapping**: consiste em **carregar todo programa** na memória, **executá-lo** por um **tempo** e **depois colocá-lo** de **volta** no **disco**. Os **processos inativos** são **armazenados** em disco, de **modo** que **não ocupam memória quando não** estão em **execução** (embora alguns deles acordem periodicamente para fazer seu trabalho e, em seguida, voltem a dormir);
    - **Memória virtual**: **permite** que os **programas** sejam **executados mesmo quando** estão **parcialmente** na **memória** principal.

Gerenciamento da Memória

- **Estratégias para lidar com alto consumo de memória**
  - **Swapping**

Inicialmente, apenas o processo A está na memória. Em seguida, os processos B e C são criados ou trocados do disco. O processo A é armazenado em disco. Então D entra e B sai. Finalmente A entra novamente. Como A está agora em um local diferente, os endereços contidos nele devem ser realocados (pode-se utilizar a técnica de base e limite)



Gerenciamento da Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Swapping**
    - Quando o **swapping cria** vários "**lacunas**" na memória, é **possível combiná-los** em uma **grande lacuna de memória**, movendo todos os **processos para baixo** (endereços baixo de memória), **tanto quanto possível**;
    - Essa **técnica** é conhecida como **compactação de memória**. **Geralmente não** é **aplicada** porque **requer muito tempo** de **CPU**. Por **exemplo**, em uma **máquina** de **16GB** que pode **copiar 8 bytes em 8 ns**, levaria cerca de **16 segundos** para **compactar** toda a **memória**;
    - Neste ponto, outro **problema** é o **tamanho** da **alocação**: se os processos são **criados** com um **tamanho fixo** de memória que nunca muda, então o **processo** de **alocação** é **simples.**

Gerenciamento da Memória

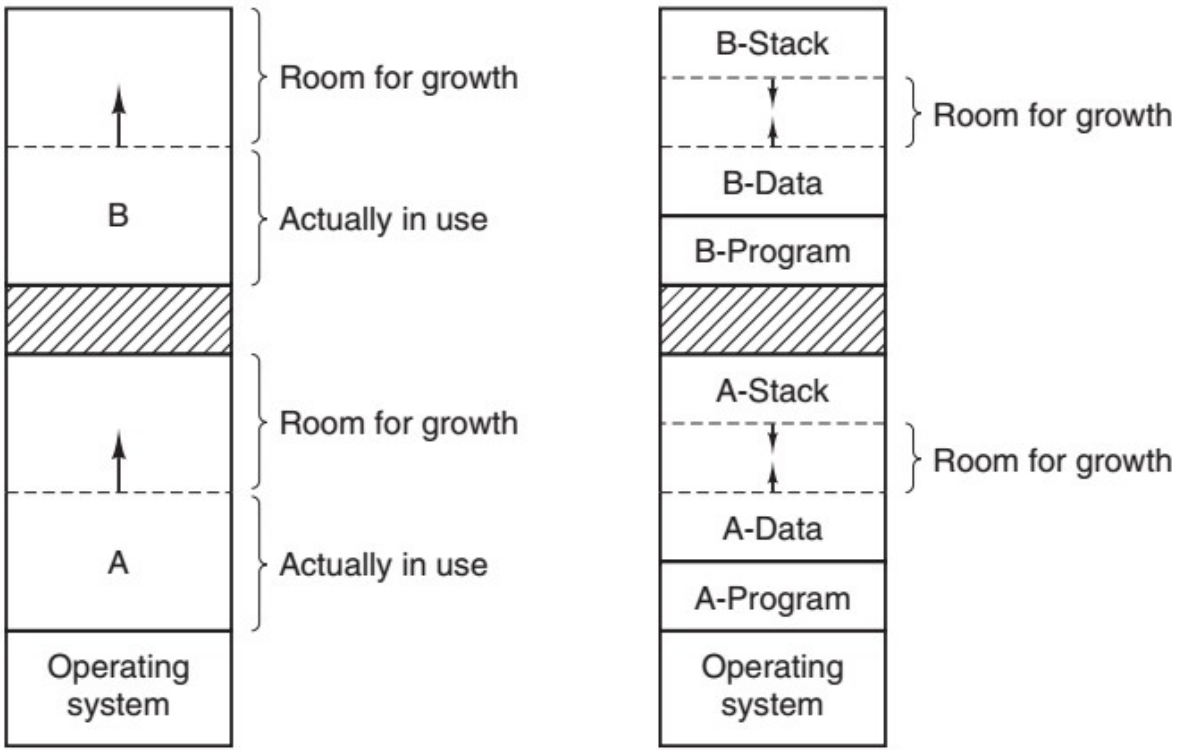
INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Swapping**
    - Se, **no entanto**, os **segmentos** de **dados** dos **processos** podem **crescer dinamicamente** como resultados de **alocações** no **heap**, ocorrerá um **problema** sempre que um **processo** tentar **crescer**;
    - Assim, **se** uma "**lacuna**" de **memória** é **adjacente** ao **espaço ocupado** pelo **processo** na **memória**, esta **lacuna** pode ser **alocado** e a **memória** do **processo** pode **crescer** com a **anexação** desta **lacuna**;
    - Por **outro lado**, se o **processo** for **adjacente** a outro **processo**, o **processo** que **quiser** "**crescer**" em **memória** terá que ser **movido** para uma **lacuna** na **memória grande** o **suficiente** para **armazená-lo**, **ou um** ou **mais processos** terão que ser **armazenados** em **disco** para **criar** uma **lacuna** de **memória** de **tamanho suficiente** para o processo;
    - Se um **processo não pode crescer** na **memória** e a **área** para **troca** no **disco** está **cheio**, o **processo** terá que **suspenso** até que **algum espaço** seja **liberado** (ou possa ser eliminado).

Gerenciamento da Memória

- **Estratégias para lidar com alto consumo de memória**
  - **Swapping**

Exemplo: alocação de espaço para um segmento de dados crescente e espaço para uma pilha crescente e um segmento de dados crescente.



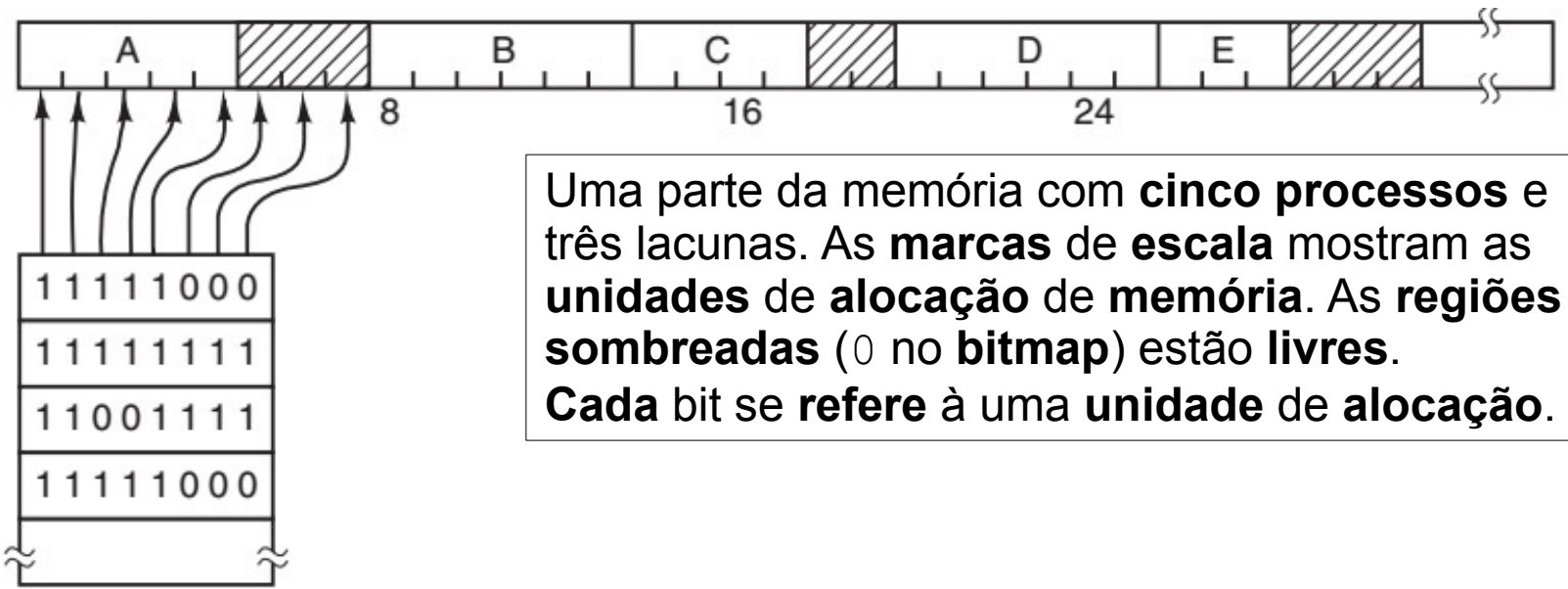
Gerenciamento da Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Métodos de gerenciamento da memória livre**
    - **Gerenciamento com mapa de bits**
      - Com **mapa de bits** (*bitmaps*), a **memória** é **dividida** em **unidades de alocação pequenas** (algumas palavras) **ou** tão **grandes** como **vários kilobytes**.
      - Um **bit** no **bitmap corresponde** a **cada unidade** de **alocação** e seu valor é `0` **se a unidade estiver livre** e `1` **se estiver ocupada**.

Gerenciamento da Memória

- **Estratégias para lidar com alto consumo de memória**
  - **Métodos de gerenciamento da memória livre**
    - **Gerenciamento com mapa de bits**



Uma parte da memória com **cinco processos** e três lacunas. As **marcas** de **escala** mostram as **unidades** de **alocação** de **memória**. As **regiões sombreadas** (0 no **bitmap**) estão **livres**.
Cada bit se **refere** à uma **unidade** de **alocação**.

Gerenciamento da Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Métodos de gerenciamento da memória livre**
    - **Gerenciamento com mapa de bits**
      - Quanto **menor** for a **unidade** de **alocação**, **maior** será o **bitmap**;
      - **Mesmo** com uma **unidade de alocação tão pequena quanto 4 bytes**, **32 bits de memória exigirão apenas 1 bit do mapa**;
      - Uma **memória de 32n bits usará n bits do mapa**, **portanto** o **bitmap ocupará apenas 1/32 da memória**;
      - **Se** a **unidade** de **alocação** escolhida for **grande**, o **bitmap será menor**, mas uma **memória considerável** pode ser **desperdiçada** na **última unidade de alocação** do **processo** se o **tamanho** do **processo não** for um **múltiplo exato** da **unidade** de **alocação**.

Gerenciamento da Memória

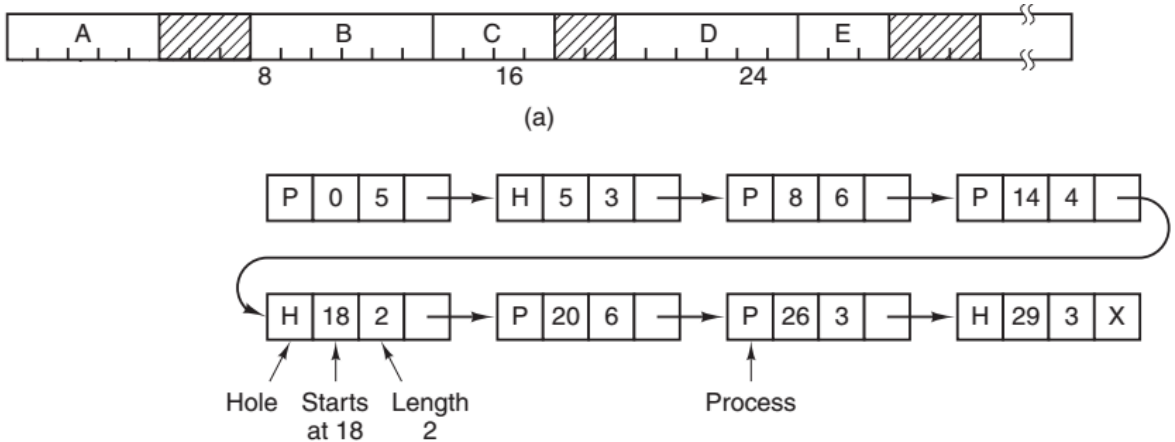
INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Métodos de gerenciamento da memória livre**
    - **Gerenciamento com mapa de bits**
      - **Vantagem**: trata-se de um **método simples** de **controlar** as **palavras** de **memória** em uma **quantidade fixa** de **memória**, pois o **tamanho** do **bitmap depende apenas** do **tamanho** da **memória** e do **tamanho** da **unidade** de **alocação**;
      - **Desvantagem**: para **carregar** a **k unidades** de **alocação** de **memória** do **processo** para a **memória**, o **gerenciador de memória deve pesquisar** o **bitmap** até **encontrar** uma **sequência** de **tamanho k** com **0 bits consecutivos** no **mapa**;
      - A **pesquisa** de **bits** em um bitmap **atrasa consideravelmente** o **processamento** de endereços da memória.

Gerenciamento da Memória

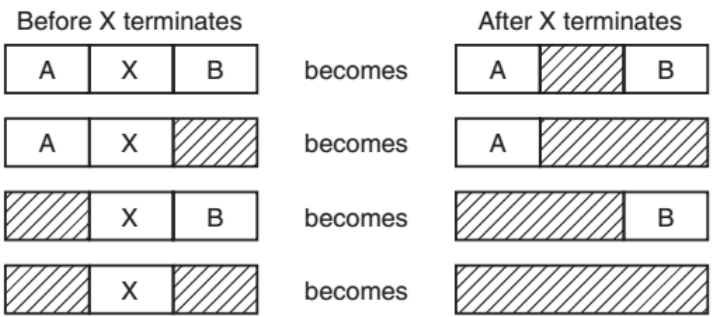
- **Estratégias para lidar com alto consumo de memória**
  - **Métodos de gerenciamento da memória livre**
    - **Gerenciamento com lista ligada**
      - Consiste em utilizar uma **lista encadeada de segmentos de memória** (**alocados** e **livres**);
      - Cada **segmento contém** ou um **processo** ou uma **lacuna** de memória existente entre dois processos:

Cada **entrada** na lista especifica uma **lacuna** (**H**) **ou processo** (**P**), o **endereço** no qual ele **inicia**, o **comprimento** e um **ponteiro** para o **próximo** item.



Gerenciamento da Memória

- **Estratégias para lidar com alto consumo de memória**
  - **Métodos de gerenciamento da memória livre**
    - **Gerenciamento com lista ligada**
      - No **exemplo apresentado**, a **lista** de segmentos é mantida **classificada** por **endereço**;
      - **Vantagem**: de que, quando um **processo** é **finalizado** ou **trocado**, a **atualização** da lista é **simples**: um **processo terminado** normalmente tem **dois vizinhos** (exceto quando está no topo ou na parte inferior da memória). **Estes** podem ser **processos** ou **buracos**, levando a quatro combinações:



Gerenciamento da Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Métodos de gerenciamento da memória livre**
    - **Gerenciamento com lista ligada**
      - **Para alocação de memória existem quatro algoritmos básicos**
        - **Primeiro que ajustar** *(first fit):* o **gerenciador** de **memória verifica** a **lista** de **segmentos** até **encontrar** uma **lacuna grande** o **suficiente**. A **lacuna** é então **dividida** em **duas partes**, **uma** para o **processo** e **outra para** o **espaço não utilizado** (se houver). O **primeiro** ajuste é um **algoritmo rápido** porque **procura** o **mínimo possível**.
        - **Próximo que ajustar** *(next fit)*. similar ao primeiro que ajustar, exceto que ele **mantém** o **controle** de **onde está** quando **encontra** uma **lacuna** adequada. Na **próxima vez** que for chamado para encontrar uma lacuna, ele **começará** a **pesquisar** a **lista** do **local** de **onde parou** da **última vez**, em vez de estar sempre no início, como faz o primeiro ajuste. Seu **desempenho** é um **pouco pior** do que o **primeiro ajuste**.

- **Estratégias para lidar com alto consumo de memória**
  - **Métodos de gerenciamento da memória livre**
    - **Gerenciamento com lista ligada**
      - **Para alocação de memória existem quatro algoritmos básicos (cont.)**
        - **Melhor ajuste** *(best fit)*: **pesquisa** a **lista inteira**, escolhe a menor lacuna que caiba o programa. O melhor ajuste **é mais lento do que o primeiro que ajustar**, pois ele deve **pesquisar toda** a **lista** toda vez que for chamado. Resulta em **mais memória desperdiçada** do que o primeiro que ajustar ou o próximo que ajustar, porque tende a encher a memória com lacunas muito pequenas e inúteis.
        - **Pior ajuste** (*worst fit*). **sempre busca** a **maior lacuna disponível**, para que a(s) nova(s) lacuna(s) criada(s) seja(m) grande(s) o suficiente para ser(em) útil(eis). Simulações provam que este algoritmo também não é muito útil.

Gerenciamento da Memória

- **Estratégias para lidar com alto consumo de memória**
  - **Métodos de gerenciamento da memória livre**
    - **Gerenciamento com lista ligada**
      - O **preço computacional** pago pelos **quatro algoritmos** apresentados é **resultado** da **desalocação** de um **processo** da lista, pois o **espaço liberado** deve ser **reintegrado** na **lista** de **lacunas**;
      - Uma forma de contornar esse problema é **trabalhar** com **duas listas**: **uma lista** de **lacunas** pode ser **mantida classificada em tamanho**, para **utilizar melhor ajuste mais rápido** e uma **lista** de **processos alocados**;
      - **Assim** que **encontra** uma **lacuna** em que se **encaixa** o **processo**, sabe-se que a **lacuna** é a **menor possível** que abrigará o processo;
      - A **vantagem** aqui é que **não** é **necessário varrer toda** a **lista**, pois a lista de **lacunas** já esta em **ordem** de **tamanho**.

Gerenciamento da Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Overlays**
    - Uma **solução adotada** na **década** de **1960** foi dividir os **programas** em **pequenos pedaços**, chamados **overlays**;
    - Quando um **programa** é **iniciado**, **carrega-se** na **memória** um **gerenciador de overlays**, que **imediatamente** carrega e executa o **overlay 0**;
    - **Quando este terminava**, ele **requisitava** ao **gerenciador** de **overlays** para **carregar** o **overlay 1 do disco**, **acima** do **overlay 0 na memória** (**se** houvesse **espaço** para isso) **ou** **sobre** o **overlay 0** (se não houvesse espaço);
    - **Desvantagens**: cabia ao programador definir a divisão do programa em overlays – nem sempre os resultados eram satisfatórios.

Gerenciamento da Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Memória virtual**
    - **Ideia básica**: cada **programa** tem seu **próprio espaço** de **endereçamento**, que é **dividido** em **partes chamadas páginas**;
    - **Cada página** é um **intervalo contíguo de endereços** – essas páginas **são mapeadas** na **memória física**, mas **nem todas** as **páginas** precisam **estar** na **memória física** para executar o programa;
    - Quando o **programa referencia** uma **parte** do seu **espaço** de **endereçamento** que **está** na **memória física**, o **hardware executa** o **mapeamento** necessário **em tempo real**;
    - Quando o **programa** faz **referência** a uma **parte** de seu **espaço** de endereço **que não** está na **memória física**, o **sistema operacional** é **alertado para buscar** a **parte** que **faltava** e então **reexecutar** a **instrução** que falhou.

Gerenciamento da Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Memória virtual**
    - A **maioria dos sistemas de memória virtual** usa uma **técnica** denominada **paginação**;
    - **Programas** fazem **referência** a um **conjunto** de **endereços** de **memória**, por **exemplo**, **quando executam** uma instrução como `MOV, REG, 1000`, que copia o conteúdo do endereço de memória 1000 para o registrador REG;
    - Esses endereços gerados pelo programa são chamados de endereços virtuais e formam o **espaço de endereço virtual**.

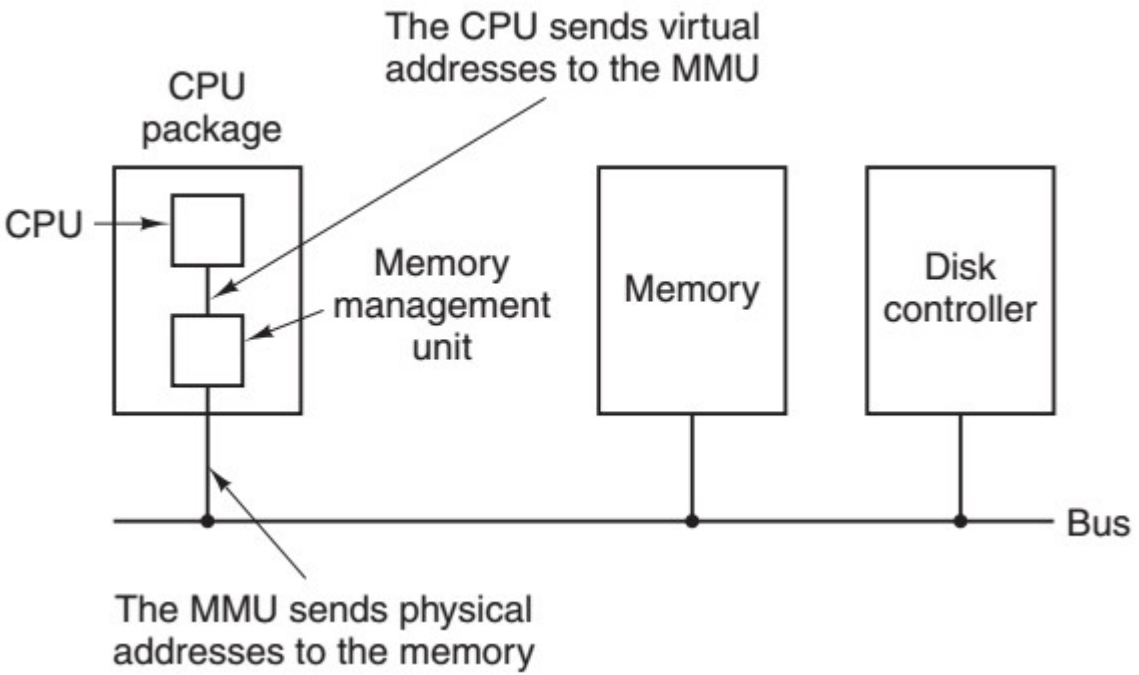
Gerenciamento da Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Memória virtual**
    - Em computadores **sem memória virtual**, o **endereço virtual** é **colocado diretamente** no **barramento** de **memória** e **faz** com que a **palavra** de **memória física** seja lida ou escrita;
    - Quando a **memória virtual é usada**, os **endereços virtuais não vão diretamente** para o **barramento** de **memória** – eles vão para uma **MMU** (*Memory Management Unit*) que **mapeia** os **endereços virtuais** para a **memória física**.

Gerenciamento da Memória

- **Estratégias para lidar com alto consumo de memória**
  - **Memória virtual**



Gerenciamento da Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Memória virtual**
    - **Exemplo de mapeamento**: considerar um **computador** que **gera endereços** de 16 bits, de 0 até 64K -1. Estes são os **endereços virtuais**;
    - **Este computador**, no entanto, **tem apenas** 32KB de memória física. Desse modo, **programas** de tamanho **64KB não** podem ser **carregados** e **executados** na **memória. No entanto**, esses programas **podem ser compilados** e **armazenados** em **disco** e carregados na demanda;
    - O **espaço de endereço virtual** consiste em **unidades de tamanho fixo chamadas páginas**;
    - As **unidades correspondentes** na **memória física** são chamadas de ***page frames***. As **páginas** e as **page frames** são **geralmente** do **mesmo tamanho**.

Gerenciamento da Memória

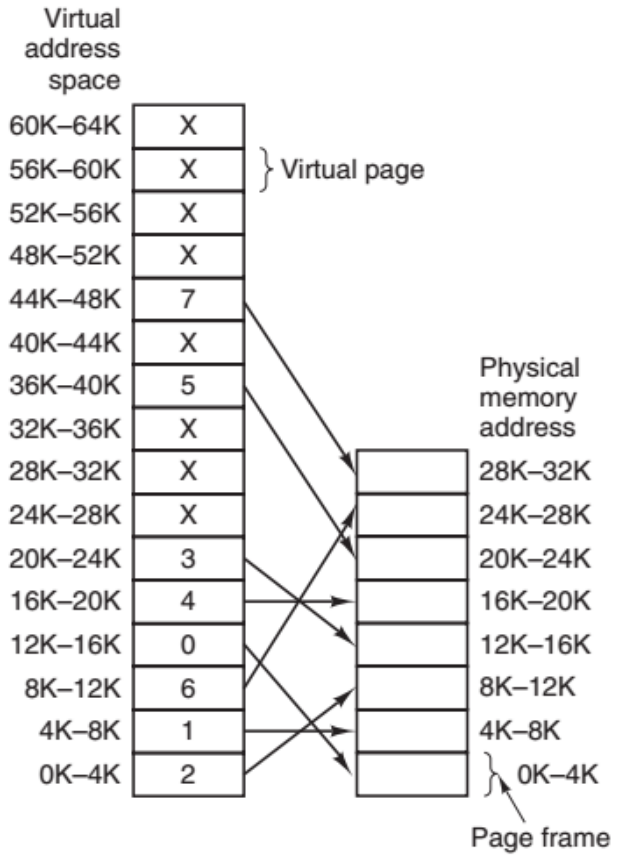
INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Memória virtual**
    - Considerar um **tamanho** de **página** de **4KB**. **Com 64KB** de **espaço virtual** de **endereçamento** e **32KB** de **memória física**, tem-se **16 páginas virtuais** e **8 page frames**;
    - **Transferências entre RAM** e **disco** estão **sempre** em **páginas inteiras** (mas muitos processadores suportam vários tamanhos de página que podem ser combinados se o sistema considera adequado);
    - Em geral é melhor usar uma única página grande, em vez de um grande número de páginas pequenas.

Gerenciamento da Memória

- **Estratégias para lidar com alto consumo de memória**
  - **Memória virtual**

O **intervalo marcado** de **0K a 4K significa** que os **endereços virtuais ou físicos nessa** página **são** de **0 a 4095**. O **intervalo** de **4K** a **8K** refere-se aos **endereços 4096** a **8191** e **assim** por **diante**.
Cada página contém exatamente 4096 endereços, **começando** em um **múltiplo** de **4096**.



Gerenciamento da Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Memória virtual**
    - No **exemplo anterior**, **quando** o **programa** tenta **acessar** o **endereço 0**, por **exemplo**, usando a **instrução** `MOV REG, 0`, o **endereço virtual** 0 é **enviado** para a MMU, que verifica que esse **endereço virtual** cai na **página 0** (0 a 4095) que, **de acordo** com o seu **mapeamento**, é a **page frame** 2 (8192 a12287);
    - Depois a **MMU** envia o endereço **8192** para o **bus** da memória física, obtendo o valor que está armazenado naquele endereço;
    - A **memória física não sabe nada sobre** a **MMU** – ela **apenas recebe** um **pedido** para **leitura** no **endereço 8192** e o **executa**!

Gerenciamento da Memória

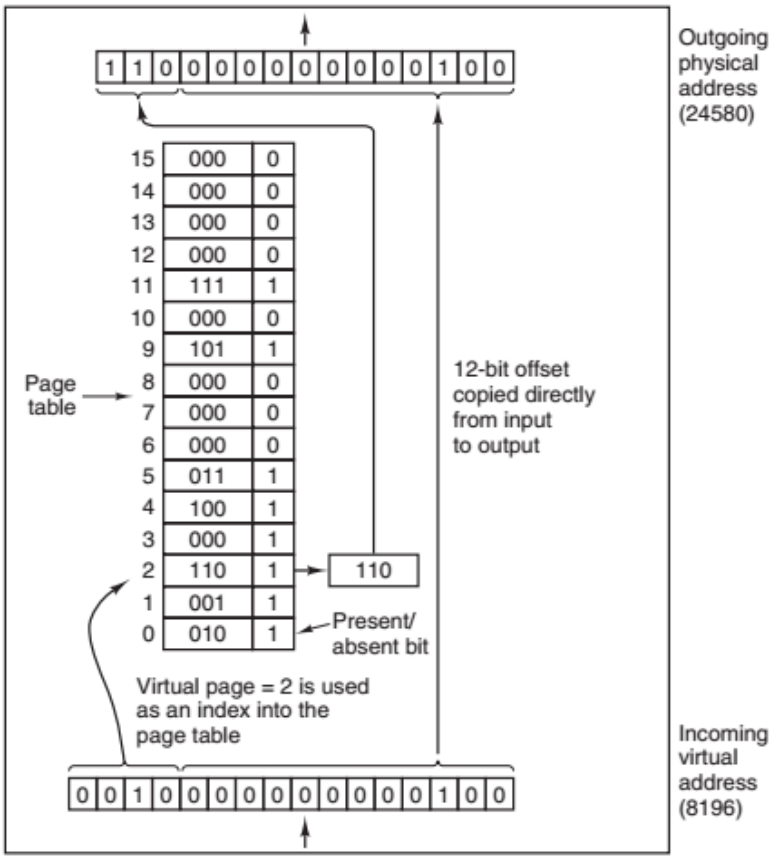
INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Memória virtual**
    - Como se tem **apenas oito frames** de **página física**, **apenas oito** das **páginas virtuais** da são mapeadas na memória física;
    - **Páginas virtuais marcadas** com uma **cruz não estão mapeadas**;
    - Em um **hardware real**, **um bit presente/ausente controla quais páginas** estão **fisicamente presentes** na **memória**;
    - Se uma **instrução fizer referência** a um **endereço não mapeado**, por exemplo, `MOV REG,32780`, a **MMU** verifica que a **página não** está **mapeada** e então **causa** à **CPU** uma **exceção** denominada **falta de página** (*page fault*);
    - O **sistema operacional** então **escolhe** a **page frame menos requisitada** e **escreve** seu **conteúdo** para o **disco** (se não estiver lá) e **então procura** (no disco) a **página** que foi **referenciada**, **altera** o **mapa** e **reinicia** a **instrução** que **gerou** a **exceção**.

Gerenciamento da Memória

- **Estratégias para lidar com alto consumo de memória**
  - **Memória virtual**

Funcionamento da MMU: o endereço virtual 8196 (001000000000100) é separado em um número de 4 bits (endereça 16 páginas) representando o número de página e um deslocamento de 12 bits. O número de página é utilizado como índice em uma tabela de páginas, recuperando o número da page frame correspondente. Se o bit presente/ausente é 0, uma exceção ocorre como já descrito. Senão, o número da page frame é copiada para os 3 bits de ordem mais alta do registrador de saída, formando um endereço físico de 15 bit, que é colocado no bus de endereço físico de memória.



Gerenciamento da Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Memória virtual**
    - **Campos típicos de uma entrada na tabela de páginas**
      - **Número** da **page frame** (já descrito);
      - **Bit presente/ausente** (já descrito);
      - **Bits** de **proteção**: para controlar se a página pode ser lida, alterada ou executada;
      - **Bits** de **modificação**: indicam se a página foi escrita – se a página foi alterada (“suja”), no momento que ela dor escolhida para dar lugar à outra, ela deve ser escrita em disco (se não foi escrita, basta descartá-la);
      - **Bit** de **referência**: indica se a página foi referenciada para leitura ou escrita;
      - **Bit de caching**: controla a leitura da página de um cache ou direto da memória.

Gerenciamento da Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Memória virtual**
    - **Aceleração da paginação**
      - **MMUs** normalmente **incorporam** em seu **hardware** um dispositivo denominado **TLB** (*Translation Lookaside Buffer*);
      - Trata-se de uma memória associativa com uma quantidade de entradas pequena (<= 256);
      - Cada entrada contém informações sobre uma página, incluindo o número da página virtual, um bit que é definido quando a página é modificada, a proteção e a page frame física em que a página está localizada;
      - Esses campos têm correspondência de um para um com os campos a tabela de páginas, exceto o número da página virtual, que não é necessário na tabela de páginas
      - Um outro bit indica se a entrada é válida (ou seja, em uso) ou não.

Gerenciamento da Memória

- Estratégias para lidar com alto consumo de memória
  - Memória virtual
    - Aceleração da paginação
      - TLB

Valid	Virtual page	Modified	Protection	Page frame
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

Gerenciamento da Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Memória virtual**
    - **Aceleração da paginação**
      - Quando um **endereço virtual** é **apresentado** à **MMU** para **tradução**, o **hardware primeiro verifica** se o seu **número** de **página virtual** está **presente** no **TLB comparando**-o com todas as **entradas simultaneamente** (ou seja, em paralelo – em hardware);
      - Se uma **correspondência válida** é **encontrada** e o **acesso não viola** os bits de **proteção**, a **page frame** é **obtida** diretamente do **TLB**, **sem** ir para a **tabela** de **páginas**;
      - Se o **número** da **página virtual** estiver **presente** no **TLB**, mas a **instrução** estiver tentando **gravar** em uma **página** somente **leitura**, uma **falha** de **proteção** é gerada;

Gerenciamento da Memória

INSTITUTO MAUÁ DE TECNOLOGIA

- **Estratégias para lidar com alto consumo de memória**
  - **Memória virtual**
    - **Aceleração da paginação**
      - **Quando** o número da **página virtual não está** no **TLB**, a **MMU detecta** a **falha** e faz uma **consulta comum** na **tabela** de **páginas**. Em **seguida**, **remove** uma das **entradas** do **TLB** e a **substitui** com a **entrada** da **tabela** de **páginas** que foi **buscada**;
      - Se essa **página** for u**s**ada **novamente**, na **segunda vez** ela **estará** no **TLB**.

Referências bibliográficas

TANENBAUM, Andrew S. **Sistemas operacionais modernos**. 3. ed. São Paulo: Pearson, 2013. 653 p.