

PCS 3446 - Sistemas Operacionais

Prof. João José Neto

AULA 02

Administração de memória física

Administração de memória

- Para quê?
 - Alocar dados e instruções do usuário e do sistema
 - Aumentar o rendimento do espaço disponível
 - Minimizar operações de entrada/saída em disco
 - Maximizar as execuções dos programas recebidos
 - Blindar o computador das ações dos programas

Administração de memória

- Funções
 - Maximizar o número de processos na memória
 - Viabilizar distribuição balanceada de recursos
 - Alternar dados/programas entre memória e disco
 - Viabilizar a execução de programas grandes
 - Proteger entre si os programas em execução

Memória física

- É um banco para padrões binários (palavras)
- Cada padrão binário ocupa uma palavra
- A cada palavra associa-se um endereço único
- Permite leitura e gravação desses padrões
- É o depósito primário para dados e instruções
- É a fonte das instruções a processar
- É de onde o programa busca e modifica dados

Administração da memória física

- Espaço físico de endereçamento é o conjunto de todas as palavras disponíveis em uma memória.
- O espaço físico é determinado pelo meio físico que implementa a memória de um computador.
- Espaço lógico de um programa é o número de palavras ocupado pelos padrões que formam seus dados e suas instruções.
- A administração da memória física é o protocolo adotado pelo sistema operacional que determina como distribuir os espaços lógicos dos programas pelo espaço físico da memória do computador.

Técnicas de administração de mem. física

- Estudam-se a seguir as técnicas usuais de administração de memória física:
 - Alocação contígua simples
 - Alocação contígua com proteção
 - *Overlay*
 - *Swapping*

Alocação Contígua Simples

- Dá ao programa acesso total ao espaço físico
- Não dá proteção ao programa nem ao sistema
- Programa pequeno subutiliza o espaço físico
- Compatível apenas com programas com espaço lógico não superior ao espaço físico
- Pode ser melhorada com o uso de técnicas auxiliares como as de *Overlay* ou de *Swapping*, descritas adiante.

Exemplo de *Alocação Contígua Simples*

**Loader e outras partes residentes
do Sistema Operacional**

**Área contígua de memória,
ocupada pelo programa**

**Área restante de memória,
não utilizada**

Alocação contígua simples de um programa na memória física

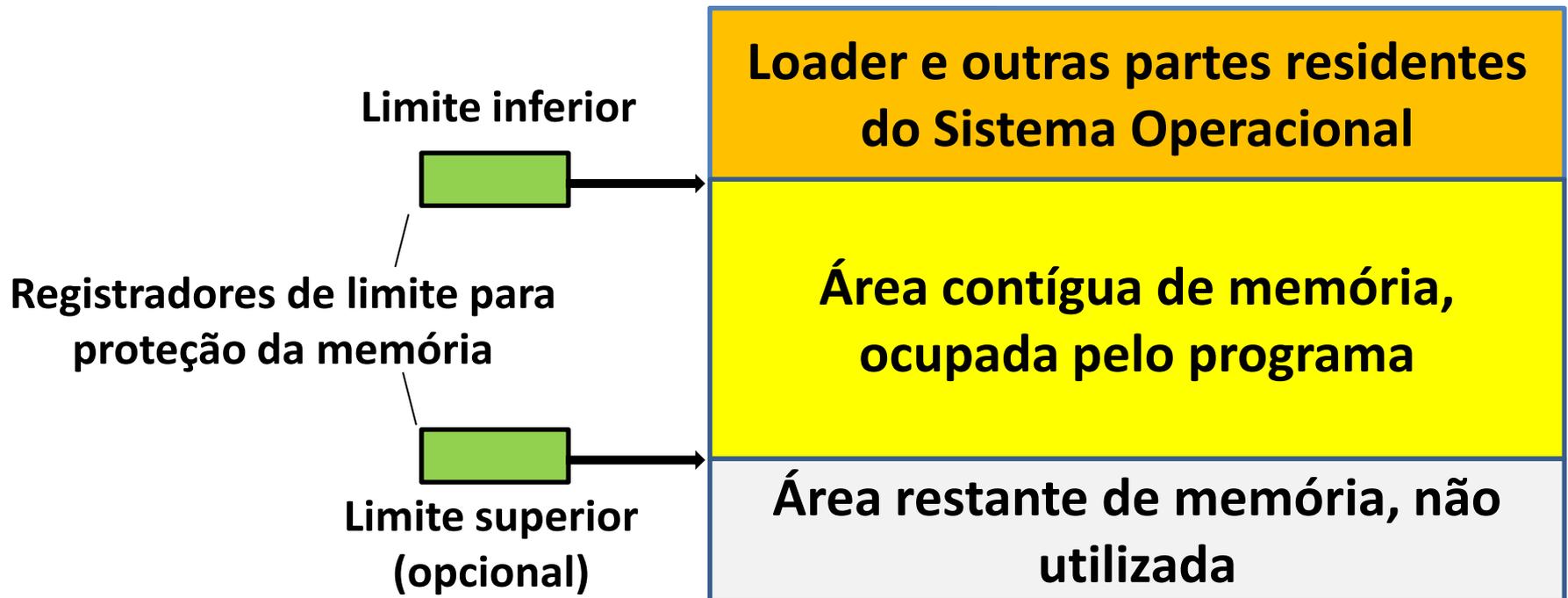
Observação

- O *Loader*, conforme foi estudado nos sistemas de programação, implementa exatamente esta técnica de alocação contígua simples, pois, presumindo que está autorizado a fazer alocações onde for mais conveniente, ao receber um código objeto absoluto, armazena cegamente na região de memória física indicada os bytes desse código, sem questionar se essa região de memória já estava ou não sendo ocupada por outro programa.

Alocação Contígua com Proteção

- Dá ao programa acesso apenas ao espaço físico em que estiver alocado seu espaço lógico
- Para isto, conta com um (ou dois) registradores de limite, em hardware, que permite(m) monitorar a atividade do programa, confinando-a ao seu espaço lógico de endereçamento.
- Programas pequenos subutilizam o espaço físico: suscitando grandes áreas ociosas na memória física
- Compatível apenas com programas com espaço lógico não superior ao espaço físico disponível
- Isso se supera com o uso de técnicas auxiliares como as de *Overlay* ou de *Swapping*, descritas adiante.

Ex. de *Alocação Contígua com Proteção*

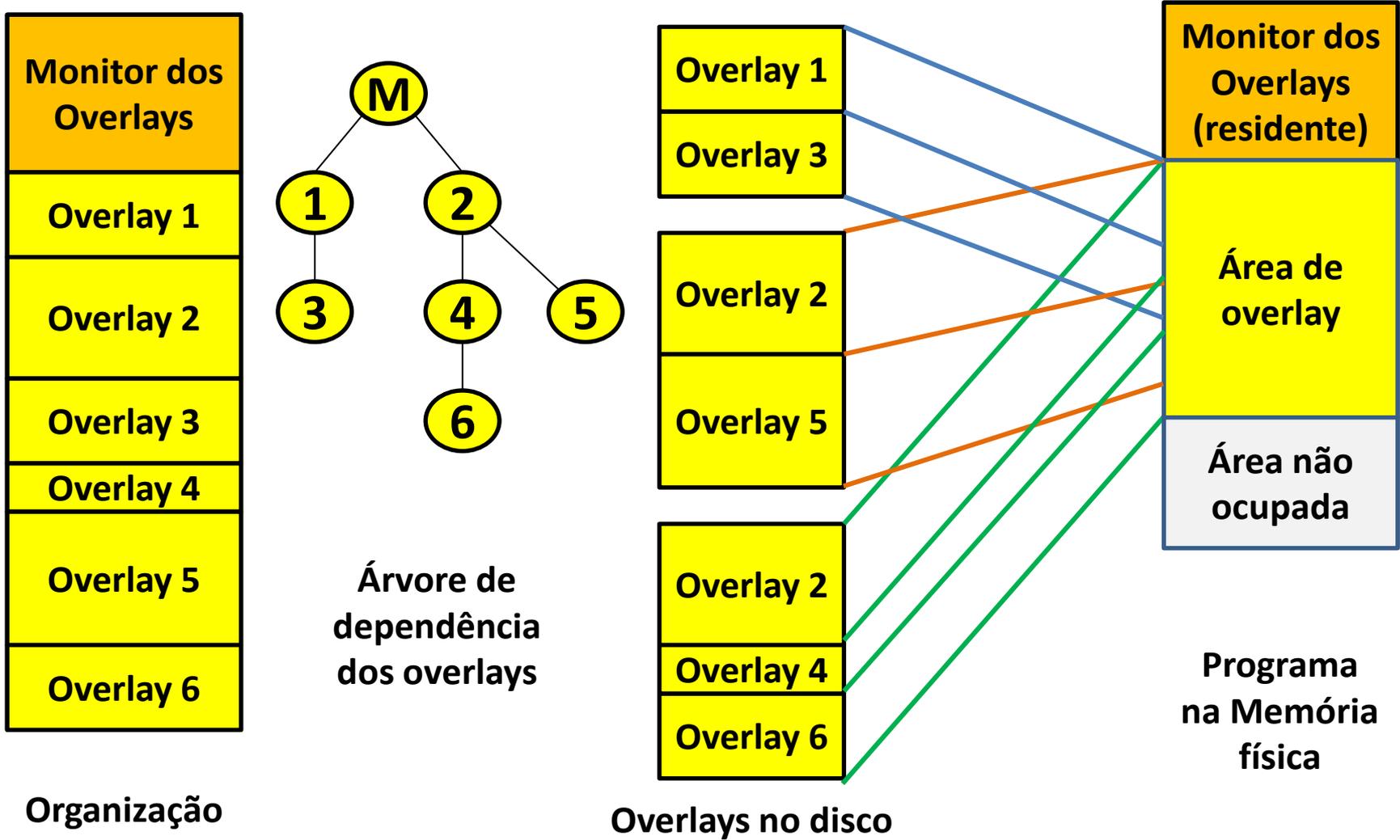


Alocação contígua simples de um programa na memória física (com proteção)

A Técnica do *Overlay*

- Nesta técnica, um monitor de *overlays* é utilizado para ler e gravar na memória e no disco, partes adequadas do programa ao longo da sua execução
- Cria-se assim a ilusão de haver um espaço de endereçamento maior que o espaço físico
- Essa técnica não é automática, cabendo apenas ao programador prever e tomar todas as providências para a correta execução do programa assim organizado.

Exemplo de *Overlay*



Organização
lógica do
programa

Overlays no disco

Observações sobre os *overlays*

- Notar, observando a árvore de dependências dos *overlays*, que embora para a execução do programa em alocação contígua simples sem *overlays* fosse necessária uma quantidade de memória igual à somatória dos tamanhos de todos os *overlays* do programa, se for usada a técnica de *overlays*, bastará uma área não menor que a máxima das somatórias dos tamanhos de todos os *overlays* que compõem os caminhos desde o monitor de *overlays* até o *overlay* que representa uma folha da árvore, naqueles caminhos (em geral muito menor que a área total do programa).
- Convém notar que esse raciocínio se aplica a áreas lógicas do programa estaticamente alocáveis, não valendo para áreas de dados de programas recursivos. Todavia, mesmo para programas recursivos, é ainda possível aplicar essa ideia ao menos para suas áreas de código e de dados estáticos.

A Técnica do *Swapping*

- Esta técnica consiste em organizar o programa como uma sucessão de blocos lógicos de código que vão se revezando convenientemente para alcançarem os objetivos do processamento.
- Em casos extremos, esses blocos chegam a ocupar todo o espaço físico disponível no computador.
- Essa técnica dá a ilusão de que toda a memória do computador é de propriedade única do programa em execução, durante sua permanência na memória física, e viabiliza programas grandes.

Observações sobre o *swapping*

- As técnicas de *swapping* e de *overlays*, embora antigas, continuam sendo utilizadas nos computadores de hoje, embora em aplicações específicas:
 - Dentro dos sistemas operacionais, nas áreas críticas em que não é possível usar a memória virtual, por incompatibilidade com a exigência de operação em modo supervisor.
 - Em sistemas embarcados (devido a restrições de espaço e limitações no poder de processamento), e em aplicações em que as restrições de memória e as limitações dos recursos de hardware impedem o uso de técnicas mais sofisticadas (esses equipamentos têm comportamentos e características similares às dos sistemas antigos, anteriores às tecnologias mais avançadas, mas podem tirar proveito de boas soluções antigas para problemas que persistem desde aquela época).

Exemplo de Swapping

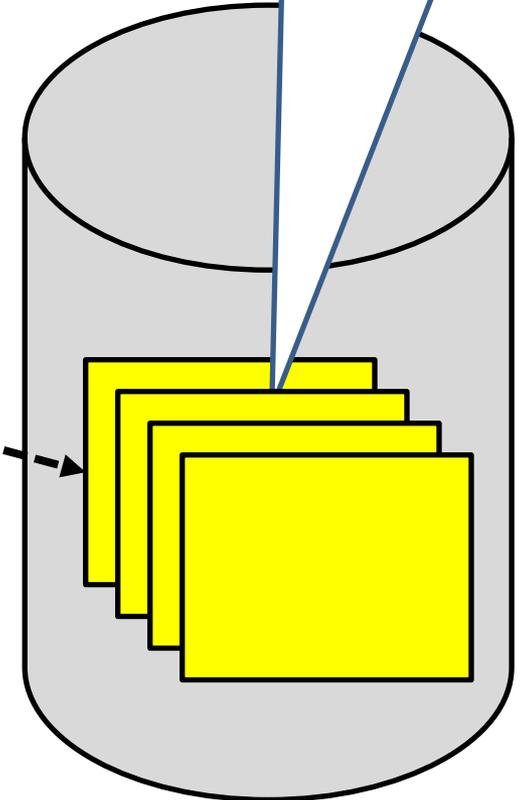
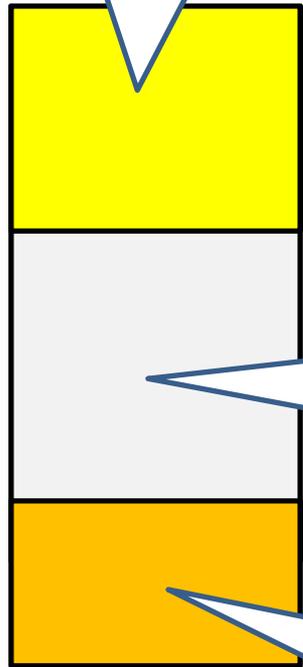
Área ocupada na memória principal por um bloco de swapping do programa do usuário

Áreas ocupadas no disco pela imagem de cada bloco de código para swapping do programa do usuário

Movimentação de conteúdo entre a memória e o disco

Área ociosa da memória principal (pode ser invadida pelos blocos de swapping)

Área ocupada na memória principal pelo sistema operacional



Swapping de um programa entre memória e disco

Comentários sobre programas grandes

- Overlays e Swapping são usados para viabilizar a execução de programas maiores que a memória física.
- Isso é importante em computadores com poucos recursos de armazenamento primário, mas como efeitos colaterais onera o tempo de processamento e aumenta a complexidade do projeto dos programas.
- Conforme a frequência das operações de *swapping*, o *overhead* devido aos tempos gastos nas transferências de informações pode mostrar-se expressivo.
- Convém notar que as técnicas de *overlay* e de *swapping* não são automáticas, sendo responsabilidade exclusiva do programador determinar a forma de operação e delimitar e dimensionar as áreas de *overlay* para o seu programa.
- Embora tenham mitigado as dificuldades de implementar programas maiores que a capacidade das memórias físicas, essas técnicas só foram sobrepujadas com a invenção das técnicas (transparentes e automáticas) de virtualização de memória.