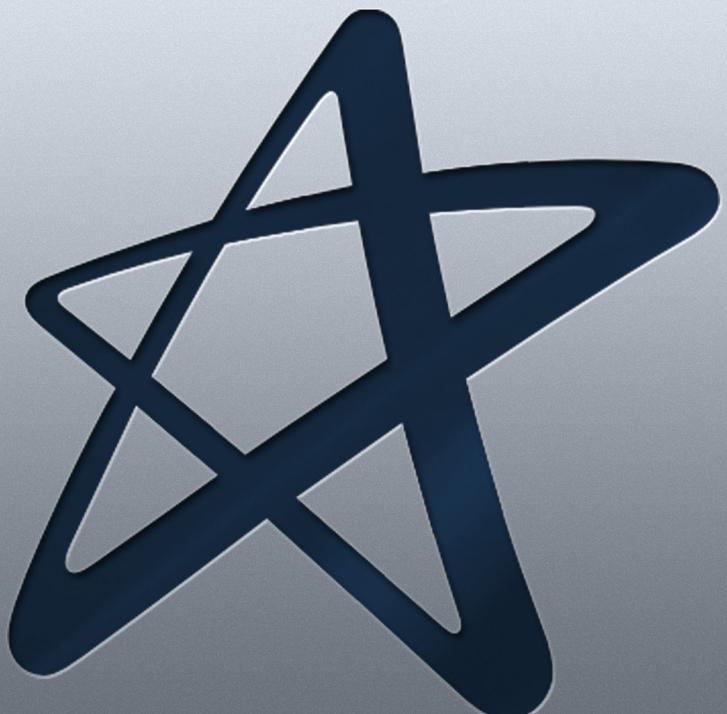


# Sistemas Operacionais



Educação a Distância  
Cruzeiro do Sul Educacional  
Campus Virtual



# Material Teórico



**Gerenciamento de Memória**

**Responsável pelo Conteúdo:**

Prof. Ms. Claudney Sanches Junior

**Revisão Textual:**

Profa. Dra. Patrícia Silvestre Leite Di Iório



# UNIDADE

## Gerenciamento de Memória



- Memória
- Gerência de Memória
- Memória Virtual
- Segmentação



O objetivo desta unidade será apresentar o gerenciamento da memória em SO e estudar um pouco mais sobre os tipos de memória, sua classificação e como o sistema gerencia a memória primária e secundária. Você será apresentado também à memória virtual, aprenderá sobre paginação e segmentação. Neste contexto, espera-se que ao final da unidade você seja capaz de entender o gerenciamento de memória dos SOs modernos.

Para que possa entender os conceitos de gerenciamento de memória do SO, esta unidade está organizada da seguinte forma:

- seção 2, apresenta o conceito e tipos de memória;
- seção 3, mostra a gerencia de memória;
- seção 4, detalha a memória virtual;
- a seção 5, apresenta as paginação e segmentação.

Ao final do estudo e das atividades desta unidade, você deve ser capaz de:

- entender a gerência da memória em SOs;
- conhecer o gerenciador de memória e seus principais algoritmos.

## Contextualização



Uma empresa comprou uma CPU com muita memória física. Faça um documento no word com os passos necessários para desabilitar no Windows a Memória Virtual. Demostre, capturando as telas, que você consegue desabilitar e utilizar o SO. Isto é, ilustre, com imagens da tela, os passos necessários para a desabilitação da Memória Virtual.

## Memória



Antes de começar a falar do gerenciamento de memória do SO, precisamos revisar os conceitos de memória, memória principal e memória secundária. O primeiro conceito a se guardar é que ao se referir à memória em computação estamos nos referindo a todos os dispositivos que permitem a um computador guardar dados temporariamente ou permanentemente. O termo memória é genérico servindo tanto para o armazenamento de dados como para o armazenamento de programas.

As memórias de alta velocidade, localizadas no processador que guardam dados para uso imediato, são as mais velozes e caras de um sistema computacional, pois operam na mesma velocidade dos processadores e recebem o nome de registradores.

Como existe uma diferença de velocidade muito grande dos registradores em relação à memória principal, surgiu a necessidade de um tipo de memória interna que intermedia o processador e o dispositivo de armazenamento, normalmente, com um serviço que antecipa a probabilidade de dados serem utilizados novamente. Esta memória mais lenta que os registradores e mais rápida que a memória principal recebeu o nome de Cache.

Com o avanço tecnológico, várias caches foram desenvolvidas, sendo, algumas conhecidas e outras não. As que daremos destaque são as caches L1, L2 e L3. As memórias caches são medidas por sua capacidade de armazenamento e sua latência. Latência é o tempo decorrido entre um ciclo de *clock* da máquina e o tempo de transferência de dados. A latência é mediada em nanosegundos ou em ciclos de processador, ou seja, ciclos de *clock* que o processador tem que rodar sem executar nenhuma operação, pois fica aguardando a memória. Assim, a Cache L1 demora 2 ou 3 ciclos para responder a uma solicitação, enquanto que se um dado for solicitado a cache L2 demorará 10 ciclos. Assim, os algoritmos de previsão de uso são também de extrema importância.

A memória principal consiste em memória volátil de acesso aleatório (*Random Access Memory* – RAM). O termo aleatório vem do sentido de que os processos podem acessar dados em qualquer ordem. O que diferenciava os sistemas de armazenamento em fita é que os dados somente eram lidos em uma determinada sequência, mas foram os meios de armazenamento populares dos primeiros computadores.

### Você Sabia ?



Todos usamos *caching*. Guardamos coisas próximas em lugares estratégicos para ter acesso fácil e rápido. Por exemplo, deixar um lápis ou caneta e papel sobre a mesa do escritório é uma forma de *caching*. Contudo, os projetistas de SO tem que tomar muito cuidado ao utilizar *caching*. A cache em computação é uma cópia do dado armazenado sendo que está cópia sofrerá constante alteração e o original não. Portanto, o SO tem que frequentemente copiar os dados da memória cache para o original – esse processo é denominado esvaziamento de cache.

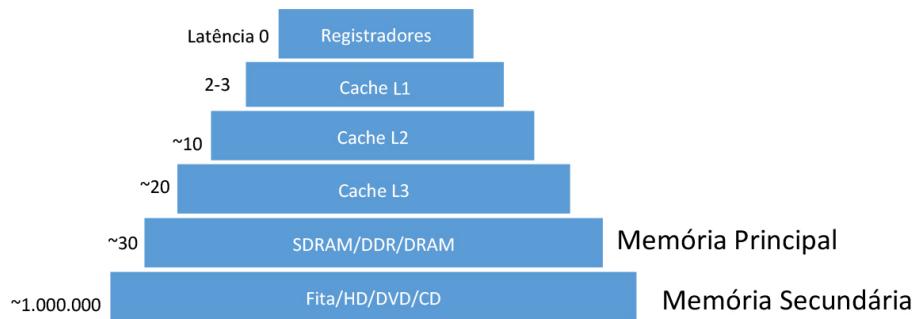
Toda memória principal ou RAM inicia com uma classificação que pode ser SIMM (*Single In-line Memory Module*) ou DIMM (*Double In-line Memory Module*), dependerá do pente de memória, ou seja, a base em que os chips são soldados. Para saber de qual se tratava basta olhar o pente que pode ter chips de memórias RAM soldadas de apenas um lado (*Single*) ou de ambos os lados (*Double*) (DEITEL, 2005).

Ainda, a memória principal pode ser classificada quando a frequência e sincronização com o barramento. Se a memória trabalha na mesma frequência do processador obedecendo ao mesmo ciclo de *clock*, sincronizando a saída de dados com os demais componentes do computador, recebe o nome de SDRAM (*Synchronous Dynamic Random Access Memory*). As memórias assíncronas ou simplesmente DRAM (*Dynamic Random Access Memory*) tem um custo mais baixo e conseguem armazenar mais dados no mesmo espaço, mas devido ao suporte a múltiplos pentes das SDRAM suplantou a DRAM.

O suporte a múltiplos pentes propicia que enquanto um atenda uma solicitação de leitura de um dado outro pode já enviar uma resposta a outra solicitação, de forma que o barramento tem uma alimentação continua. Outra característica que diferencia a DRAM é que a mesma necessita de um circuito de *refresh* ou renovação do sinal algumas vezes em um milissegundo para não perder os dados enquanto que a SDRAM não necessita de tal recurso (DEITEL, 2005).

Para finalizar a classificação de memória principal, devemos pensar na entrega dos pentes de memória. A memória poderia entregar o dobro de dados em uma só transmissão. Assim, surgiu o padrão DDR (*Double Date Rate*) que dobrou a taxa de transferência de dados. Depois surgiram a DDR2 e DDR3 que aumentaram este fator para 4x e 8x respectivamente. E já foi lançada a DDR4 que deverá chegar à produção em massa após 2.014.

A memória secundária é uma expansão da memória principal menos dispendiosa e mais lenta. A latência do armazenamento em disco (HD) é, normalmente, medida em milissegundos, em geral um milhão de vezes mais lento do que as memórias cache que são memórias colocadas próximas aos processadores com latência de 10 a 20 ciclos de *clock*. A figura 1 apresenta a latência das memórias (DEITEL, 2005).



Uma vantagem dos dispositivos de armazenamento secundário é que eles tem grande capacidade de armazenamento e os dados são guardados em meio persistente, portanto preservados quando se retira a fonte de energia do dispositivo (TANENBAUM, 2009).

## Gerência de Memória



O que todo programa deseja é dispor de maneira infinita, rápida, não volátil e a um baixo custo de uma memória que pudesse conter todo seu conteúdo. A função do SO é abstrair a hierarquia e latência das memórias existentes em um modelo útil e então gerenciar essa abstração. (TANENBAUM, 2009).

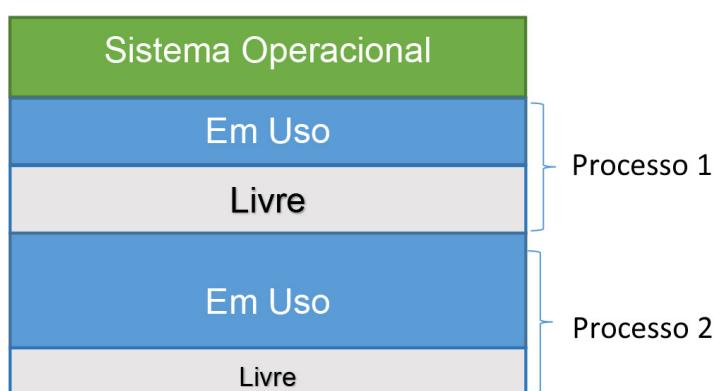
A parte do SO que gerencia parcialmente a memória do computador é denominada Gerenciador de Memória. Entre suas tarefas, podemos citar a tarefa de alocar memória para os processos quando estes precisam, a tarefa de liberar memória quando um processo termina e tratar a tarefa de tratar do problema de swapping. O gerenciador de memória é o componente do SO que se preocupa com o esquema de organização da memória e com a estratégia do gerenciamento.

Programas  
tendem a se  
expandir e ocupar  
toda a memória  
disponível. Lei de  
Parkinson.

O Gerenciador de Memória tem como primícias otimizar a utilização da memória principal. Isso era fácil nos primeiros SO que simplesmente utilizavam a memória física disponível (1960-1970). Neste período não era possível rodar mais de um programa por vez, pois o SO entregava a gerência dos endereços ao programa que estava em execução e o programa do usuário podia acessar os endereços que estavam alocados para o próprio SO.

Para permitir que mais de um programa execute simultaneamente, o SO deve resolver dois problemas: o de segurança e o de realocação dos endereços atribuídos ao programa. A solução é criar um espaço de endereçamento para cada processo. Assim, o endereço 28 na memória de um processo é diferente do endereço 28 em outro processo. Uma forma simples de criar os espaços é, via hardware, criar dois registradores um registrador-base e outro registrador-límite e cada processo deverá somar o registrador-base em seus endereços e comparar o resultado com o registrador-límite para ver se está dentro do espaço de endereçamento reservado para ele. Apenas o SO poderá atribuir valores para esses registradores.

O primeiro PC 8088 utilizava parte desta solução denominada de realocação dinâmica, pois tinha apenas o registrador-base (TANENBAUM, 2009). O maior problema desta solução é a fragmentação da memória conforme ilustrado na figura 2 que acaba desperdiçando memória, por isso surgiu a necessidade de tentar outra solução.



O grande problema desta solução que adotava limites fixos de memória é a fragmentação e a limitação de processos ativos, pois o SO dividia toda a memória quando era carregado e alocava partes iguais aos processos. Adotando-se um número variável para o espaço da memória um número maior de processos podem ser alocados. Essa solução demanda do Gerenciador de Memória um algoritmo de alocação da memória livres. Assim, os novos processos sempre poderão ser alocados.

Elaboraram-se três métodos de seleção de uma região livre: Melhor Escolha (*best fit*), Pior Escolha (*worst fit*) e Primeira Escolha (*first fit*). O método melhor escolha coloca o processo na menor região livre, o pior escolha coloca o processo com a maior região livre e o primeira escolha aloca a primeira região livre para o processo. Veja a figura 3 que apresenta um processo que demanda 14 kbytes em uma memória com 5 processos em uso.

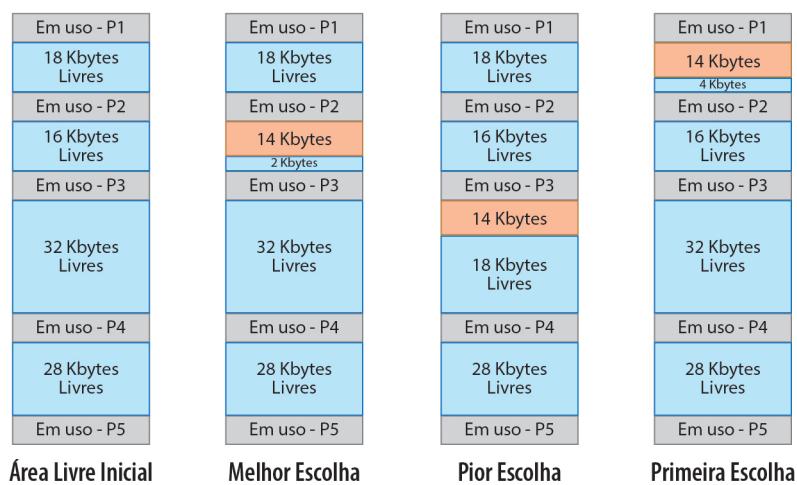


Figura 3. Gerencia de Memória – Alocação de Região Livre

O uso do algoritmo de seleção de memória livre melhor escolha deixa o menor resto, porém após um longo período de processamento deixará espaços muito pequenos na memória para serem úteis a algum processo. O algoritmo pior escolha deixa o maior espaço após cada locação, mas tende a espalhar as porções não utilizadas sobre área não contínua de memória tornando difícil alocar grandes processos. O algoritmo primeira escolha tende a ser um meio termo com característica adicional de fazer com que os espaços vazios migrem para o final da memória. O gerenciador de memória deverá com qualquer uma dessas abordagem manter uma lista de chamada “Lista Livre” dos blocos disponíveis com informações sobre o seu tamanho. Antes de retornar um bloco à lista livre, o gerenciador de memória deve verificar se o bloco liberado está próximo a outros blocos de forma que possam ser combinados, formando um bloco maior.

## Memória Virtual



A maioria do SO enfrenta a dificuldade da falta de memória física para a demanda dos processos ativos. Por exemplo, facilmente ao ligar o seu computador com o SO Windows ou SO Linux, uns 50 – 200 MB serão alocados. A estratégia de manter um processo na memória principal até sua finalização recebe o nome de swapping que pode ser traduzida por troca ou permuta. O Gerenciador de Memória deverá trocar os dados que estão na memória principal (RAM) por dados que estão na memória secundária (HD) e utilizar a memória secundária como uma extensão da memória principal.

Com o tempo, o Gerenciador de Memória foi aperfeiçoado para antecipar a necessidade das trocas, permitindo que vários processos permanecessem na memória principal ao mesmo tempo. Os novos algoritmos de swapping trocam um processo somente quando outro precisar daquele espaço de memória, visto que a memória secundária é mais lenta. Com uma quantidade de memória principal suficiente, esses sistemas reduzem muito o tempo gasto nas trocas. Os sistemas de swapping do início da década de 1960 levaram a uma nova estratégia – a memória virtual com paginação.

A estratégia da Memória Virtual é permitir que a memória física e memória secundária sejam combinadas numa nova e única memória. A ideia básica é que cada programa tenha seu espaço de endereçamento dividido em blocos chamados de páginas. Cada página é uma série continua de endereços. Estas páginas têm seus endereços mapeados principalmente na memória física, de forma que ao acessar um endereço que tenha o equivalente em memória física o sistema apenas passa o endereço físico correspondente. Caso o endereço seja da parte gravada na memória secundária, o sistema irá solicitar a troca de forma que os dados que estão na memória física serão gravados na memória secundária e os dados da memória secundária serão carregados para a memória física. Quando o processo finalizar as trocas, o sistema informa o endereço físico correspondente (DEITEL, 2005).

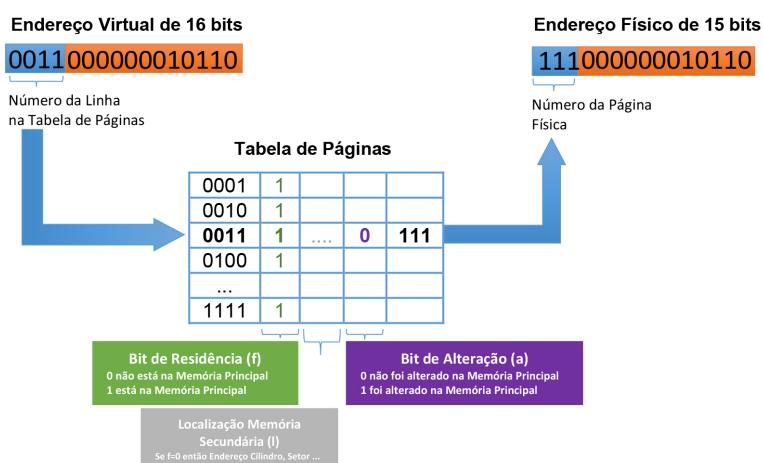
## Glossário



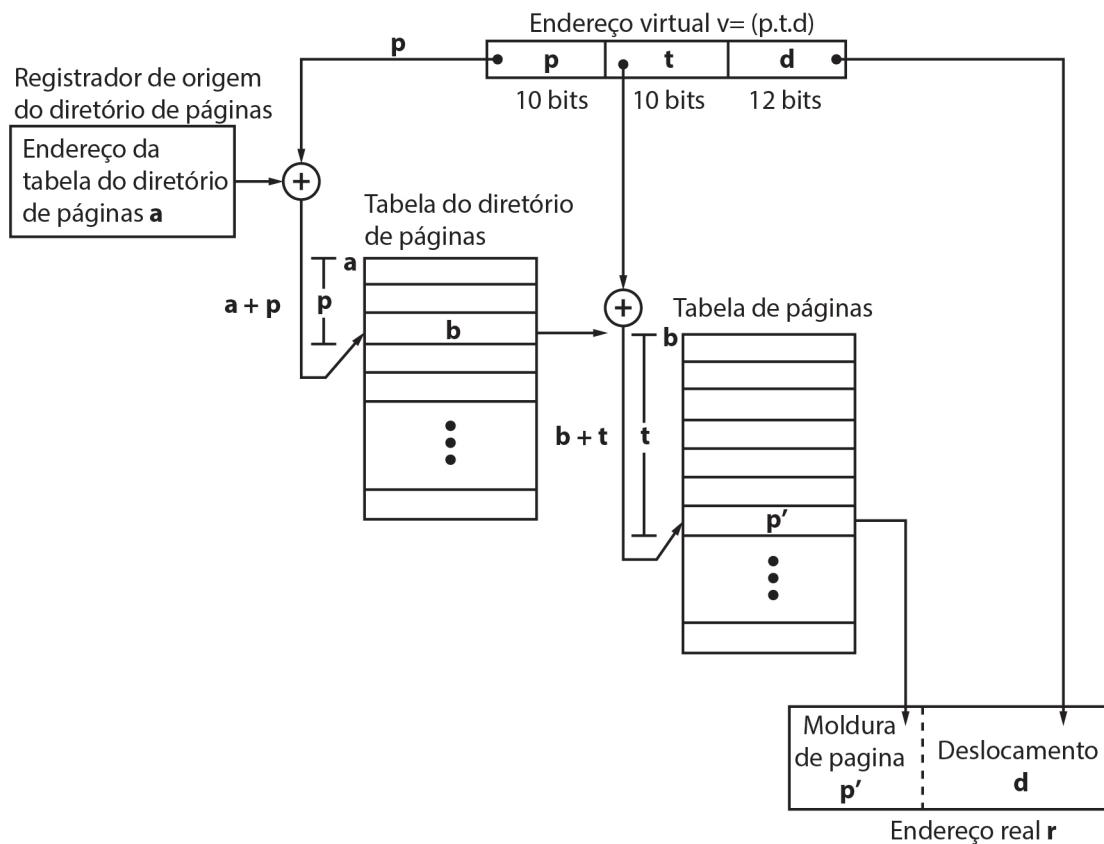
- Espaço de endereçamento Virtual: são os endereços que o programa pode referenciar;
- Espaço de endereçamento Físico: são os endereços reais de memória;
- Tabela de Páginas: relaciona os endereços virtuais com os endereços físicos.

A paginação tem como premissa que existe memória secundária disponível e suficiente para manter o programa completo com seus dados. A cópia do programa na memória secundária pode ser considerada o original enquanto que as partes trazidas da memória principal de vez em quando podem ser consideradas copias. Quando modificações são feitas na memória principal, essas modificações devem ser refletidas na memória secundária. O espaço de endereçamento virtual é dividido em páginas de tamanhos iguais. Sempre que um processo solicitar o acesso a um endereço, ele está solicitando através de um endereço virtual, que deve ser traduzido em um endereço físico.

Isso acontece com tanta frequência que solicitar ao processador fazer a tradução custaria na performance, de modo que o sistema de memória virtual tem que ter um hardware para este propósito especial. Existe a unidade de gerenciamento de memória (Memory Management Unit – MMU) que mapeia rapidamente endereços virtuais para endereços reais e o mecanismo de tradução dinâmica de endereços (Dynamic Address Translation – DAT) que convertem endereços virtuais em físicos durante a execução. A tradução de um endereço de memória virtual para um endereço físico é feita como na Figura 4. O sistema mantém uma tabela de mapas de blocos para cada processo.



O uso dessa solução pode levar a tabelas muito extensas e tornar a MMU muito lenta. Para ilustrar: de 32 bits de endereços e páginas de 4K tem-se 1 milhão de entradas na tabela. Para solucionar este problema, a ideia básica é manter todas as tabelas na memória optando pelo uso de dois apontadores e um deslocamento. Assim, o primeiro apontador é do Diretório (p) e o segundo (t) aponta para os quadros e o terceiro para o deslocamento (d) (DEITEL, 2005).



## Segmentação



A segmentação parte do princípio que um programa pode ser dividido em dados e instruções e que estes podem ser armazenados em blocos chamados segmentos. Os segmentos não precisam ser do mesmo tamanho, nem ocupar posições adjacentes na memória principal. Um segmento que corresponda a um array é tão grande quanto o array. O segmento gerado para conter um código é do tamanho do código. O sistema gerenciador de memória segmentado mantém na memória principal apenas os segmentos necessários para a execução em um determinado instante. Um processo pode executar enquanto suas instruções e dados estiverem localizados na memória principal. Se o processo referenciar um segmento que não está na memória principal, o gerenciador deverá recuperar o segmento solicitado. Um segmento que chegar poderá ser alocado para qualquer área disponível na memória principal que for grande suficiente para contê-lo (DEITEL, 2005).

Há muitas estratégias para implementar a tradução de endereços de segmentação, pois um sistema pode empregar mapeamento direto, associativo ou mapeamento combinado/associativo. Algumas vantagens da segmentação são:

- facilidade de compilação e ligações entre os procedimentos separados em segmentos;
- a mudança do tamanho de um segmento não afetará os demais;
- a segmentação facilita partilhar dados entre vários processos;
- o programador tem ciência do conteúdo do segmento e pode protegê-lo.

A desvantagem é que os segmentos dependem dos processos, mas os blocos de memória são recursos de hardware e seu tamanho é dependente do sistema. Uma solução para este problema é a combinação de segmentação com paginação.

## Material Complementar

Com a Máquinas Virtuais (VM) que você instalou na Unidade I, inicie no SO Windows a gerência de memória virtual. O SO Windows apresenta uma interface amigável para gerenciar Memória Virtual. Pesquise os comandos, teste e experimente modificar e desabilitar a Memória Virtual.



### Memória Virtual

- <http://canaltech.com.br/o-que-e/windows/O-que-e-e-como-gerenciar-a-memoria-virtual-do-Windows/>

### Comandos para alterar o tamanho

- <http://windows.microsoft.com/pt-br/windows-vista/change-the-size-of-virtual-memory>

## Referências

TANENBAUM, A.S. **Sistemas Operacionais Modernos**. 3 ed. São Paulo; Pearson Prentice Hall, 2009

DEITEL, H.M. **Sistemas Operacionais**. 3 ed. São Paulo; Pearson Prentice Hall, 2005

# Anotações





**Educação a Distância**  
Cruzeiro do Sul Educacional  
*Campus Virtual*

www.cruzeirodosulvirtual.com.br  
Campus Liberdade  
Rua Galvão Bueno, 868  
CEP 01506-000  
São Paulo SP Brasil  
Tel: (55 11) 3385-3000



Universidade  
**Cruzeiro do Sul**



**UNICID**  
Universidade  
Cidade de S. Paulo



**UNIFRAN**  
Universidade  
de Franca



**UDF**  
Centro  
Universitário



**Módulo**  
Centro  
Universitário