**Introdução da Unidade**



**Objetivos da Unidade(5M)**

Ao longo desta Unidade, você irá:

* descrever a evolução do gerenciamento de memória utilizados em computadores multiprogramados;
* explicar os conceitos do *swapping* e o funcionamento dos algoritmos de troca de processos;
* identificar as características da memória virtual e esclarecer o funcionamento dos algoritmos de substituição de páginas.

**Introdução da Unidade**

Ola estudante, boas-vindas!

Nesta unidade, veremos como é realizado o gerenciamento de memória nos sistemas operacionais e suas características, como se dá a monoprogramação sem troca de processos ou paginação, e a multiprogramação com partições fixas, além de apresentar os problemas da multiprogramação: relocação e proteção. Estudaremos a troca de processos, conhecida como *swapping*, e a multiprogramação com partições variáveis.

Ainda, aprenderemos como são realizados o gerenciamento de memória por meio de mapa de bits e os algoritmos de troca de processos, o funcionamento da memória virtual, da paginação e da tabela de páginas, assim também como são realizadas a segmentação da memória e os algoritmos de substituição de páginas.

A memória é um dos recursos computacionais mais importantes, pois é o local onde são armazenados dados e informações no computador. Logo, é necessário que seja feito um bom gerenciamento da memória nos sistemas operacionais, uma vez que a tendência dos programas é consumir cada vez mais este recurso durante sua execução.

Neste âmbito, vamos adentrar no seguinte contexto: você trabalha em uma empresa de infraestrutura como gerente de TI e foi convidado para representa-la em uma importante feira de tecnologia que acontecerá em sua cidade dentro de alguns dias. A empresa fornece serviços e soluções para problemas de recursos computacionais de acordo com a necessidade dos contratantes.

Neste evento estarão presentes as melhores empresas na área de tecnologia do país e todas as inovações de gerenciamento de memória, *swapping* e memória virtual serão apresentadas. Você será o responsável pela apresentação técnica das inovações em gerenciamento de memória, *swapping* (que é a paginação de memória em disco) e memória virtual da empresa para os clientes, apresentando os conteúdos teóricos e práticos sobre eles. Para isso será necessário dominar os seguintes conceitos:

1. saber como funcionam o controle e a alocação de memória principal para os processos, o gerenciamento de memória e mostrar como são gerenciados os problemas da multiprogramação (relocação e proteção);
2. saber como se dá o tratamento da insuficiência de memória dos processos que estão sendo executados, utilizando a técnica *swapping* (troca de processos) e como são definidas as estratégias de organização da memória através da multiprogramação com partições variáveis e do gerenciamento da memória com mapa de bits, além de apresentar os principais algoritmos de troca de processos;
3. mostrar como é tratada a insuficiência de memória dos processos que estão sendo executados, utilizando a técnica de memória virtual, e também mostrar as técnicas de paginação e segmentação e apresentar os algoritmos de substituição de páginas.

Algumas observações importantes que podem surgir durante o evento:

* qual a função do gerenciador de memória?
* quais são os esquemas de gerenciamento de memória?
* quando ocorrem os erros de páginas?
* e quais as ações que o sistema operacional toma quando ocorre este erro?

Você deverá elaborar um anteprojeto de gerenciamento de memória e de dispositivos de entrada e saída para cada cliente que procurar o seu *stand*. Após o término desta unidade você terá condições de entender como é realizado o gerenciamento de memória, o funcionamento do *swapping* e da memória virtual no computador.

Vamos juntos conhecer mais sobre o gerenciamento de memória e suas propriedades! Vamos lá!

**Introdução da aula**



**Qual é o foco da aula?**

Nesta aula, vamos apresentar a evolução do gerenciamento e os tipos de memória que auxiliam na execução das aplicações para elevar o desempenho do computador.

**Objetivos gerais de aprendizagem**

Ao longo desta aula, você irá:

* descrever o gerenciamento de memória, importante recurso para um bom funcionamento de um computador;
* distinguir os métodos de gerenciamento de memória: monoprogramação e multiprogramação;
* analisar os problemas da multiprogramação: relocação e proteção.

**Situação-problema**

Em um sistema operacional, o gerenciamento de memória é fundamental para garantir a eficiência das aplicações que rodam no computador, uma vez que a tendência é consumir cada vez mais memória durante sua execução.

Iremos apresentar a evolução do gerenciamento de memória, desde o mais simples até os sistemas de gerenciamento de memória utilizados em computadores multiprogramados. Existem diferentes tipos de memória que auxiliam na execução das aplicações.

Por exemplo, a memória cache é rápida e é utilizada juntamente com o processador, elevando assim o desempenho do computador. Quando uma página é acessada constantemente num navegador, ao invés do processador fazer uma busca das informações na memória RAM, ela busca as informações na memória cache.

Neste caso, as informações acessadas são copiadas para essa memória, agilizando a recuperação da página desejada. O controle de todas é de responsabilidade do gerenciador de memória, que realiza a gestão de quais partes estão sendo usadas e quais não estão.

Relembrando nosso contexto, você trabalha em uma empresa de infraestrutura como gerente de TI e irá representa-la em uma importante feira de tecnologia. Durante a feira, um funcionário de uma empresa do ramo de produtos de beleza, responsável pelo setor de TI, procura seu *stand* interessado em substituir os computadores nela utilizados.

Ele ressalta que precisará de máquinas potentes, com memória que suporte o processamento de um grande volume de dados, uma vez que a empresa atua em nível nacional. Ele informou que no início da empresa, não obteve êxito na gestão das informações e teve que começar do zero, devido ao fato da memória dos computadores não suportar a quantidade de dados processados como pedidos de produtos, vendas realizadas, orçamento dos revendedores, entre outros.

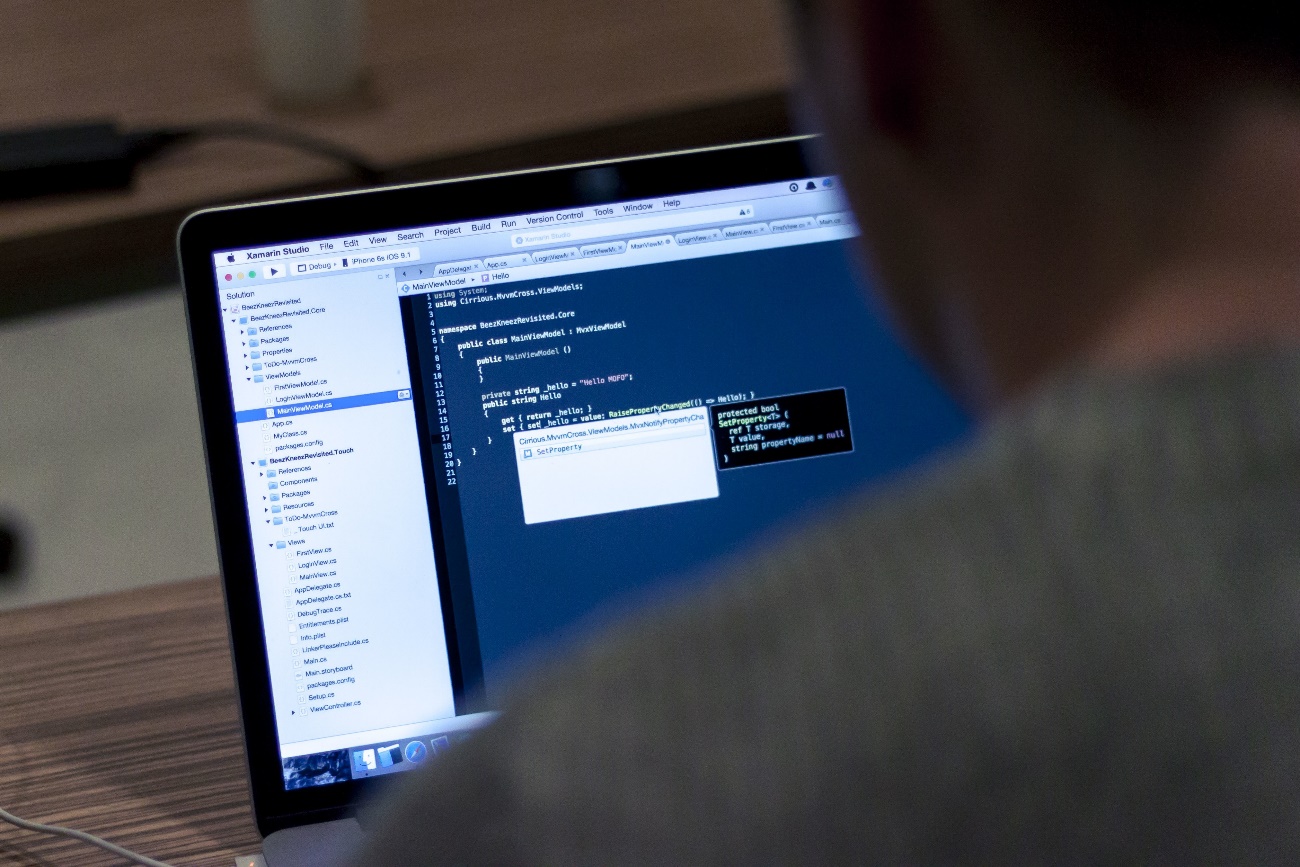
Em algumas situações os dados foram perdidos e/ou sobrescritos. Surgem os seguintes questionamentos:

* qual a solução para este cliente referente ao gerenciamento de memória dos computadores de sua empresa?
* qual estratégia poderá ser adotada para obter um bom desempenho de memória considerando a situação atual da empresa?

Para que você consiga responder a esses e outros questionamentos sobre o gerenciamento de memória e os problemas da multiprogramação, vamos conhecer mais sobre eles. Não se esqueça que, ao fim desta unidade, você deverá entregar um relatório para o seu gestor, com os anteprojetos elaborados durante a feira.

Bons estudos!

**Gerenciamento de memória**



Segundo Machado e Maia (2007), desde os primeiros sistemas operacionais a memória do computador era considerada como um recurso caro e escasso. Mesmo com a redução de custo e aumento da capacidade de armazenamento das memórias, gerenciar o seu uso é fundamental e complexo.

Memória é o local de armazenamento de informações no computador e o seu gerenciamento em um sistema operacional é importante para garantir a eficiência das aplicações que rodam no computador, uma vez que a tendência das aplicações dos usuários é consumir cada vez mais esse recurso durante sua execução.

Além disso, o gerenciamento de memória permite que vários processos sejam executados ao mesmo tempo, mantendo um bom desempenho do sistema (TANENBAUM, 2003). Machado e Maia (2007) afirmam que o sistema operacional deve proteger as áreas de memória utilizadas pelos processos de forma que, se um programa tente acessá-la indevidamente, o sistema seja impedido.

\_\_\_\_\_\_

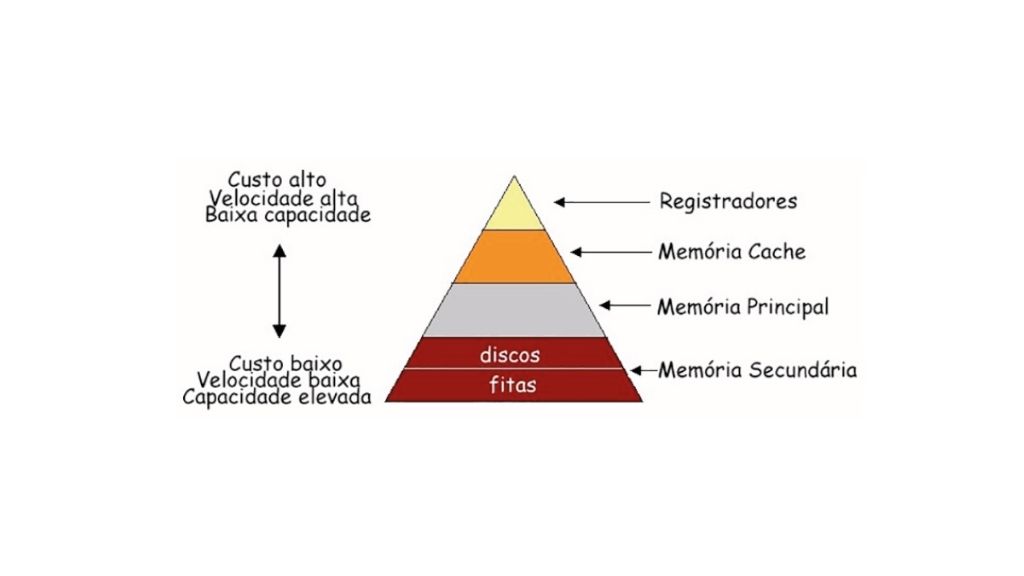
**💭 Reflita**

A memória é um recurso importantíssimo para um bom funcionamento de um computador. Quais são os desafios do gerenciamento de memória? Quais seriam os efeitos se o controle da memória fosse mal gerenciado?

\_\_\_\_\_\_

Segundo Tanenbaum (2003), na maioria dos computadores existe o conceito de hierarquia de memória (figura “Hierarquia de memórias”), combinando uma pequena quantidade de memória cache (rápida, alto custo e volátil – guardam a informação temporariamente), uma grande quantidade de memória principal (RAM – volátil, possui um médio custo) e uma memória secundária (baixo custo, alto armazenamento das informações, não volátil, com as informações armazenadas em disco).

Quanto mais alto as memórias estiverem na pirâmide, mais caras serão, possuirão alta velocidade de processamento e baixa capacidade de armazenamento. E quanto mais baixo e mais baratas são, possuirão uma grande capacidade de armazenamento e baixa velocidade de processamento.

Hierarquia de memórias. Fonte: Universidade Federal da Paraíba.

Segundo Tanenbaum (2003), a hierarquia de memória é controlada pelo gerenciador de memória, responsável por gerenciar quais partes estão em uso e quais não estão, alocando-a quando os processos precisarem, liberando-a após o término dos processos e controlando a troca de processos entre a memória e o disco quando a memória principal não é suficiente para manter todos os processos em execução.

\_\_\_\_\_\_

**➕ Pesquise mais**

A hierarquia de memória fornece para as aplicações dos usuários e do computador uma grande quantidade de memória barata, rápida e não volátil. É necessário que estas memórias estejam em sintonia enquanto o computador esteja ligado, ou seja, que elas se interliguem para garantir o bom funcionamento do sistema operacional.

Para saber mais sobre este assunto acesse indicamos o artigo [**Como funcionam as diferentes memórias quando o computador está em uso**](https://www.tecmundo.com.br/armazenamento/9415-como-funcionam-as-diferentes-memorias-quando-o-computador-esta-em-uso.htm).

\_\_\_\_\_\_

Segundo Tanenbaum (2003), o gerenciamento de memória pode ser dividido em duas classes:

**1)** sistemas que durante o processamento levam e trazem a informação da memória para o disco (troca de processos e paginação);

**2)** 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 de forma eficiente.

A seguir serão apresentados os mecanismos de gerenciamento de memória que não fazem troca de processos ou paginação.

**Métodos de gerenciamento de memória: monoprogramação e multiprogramação**

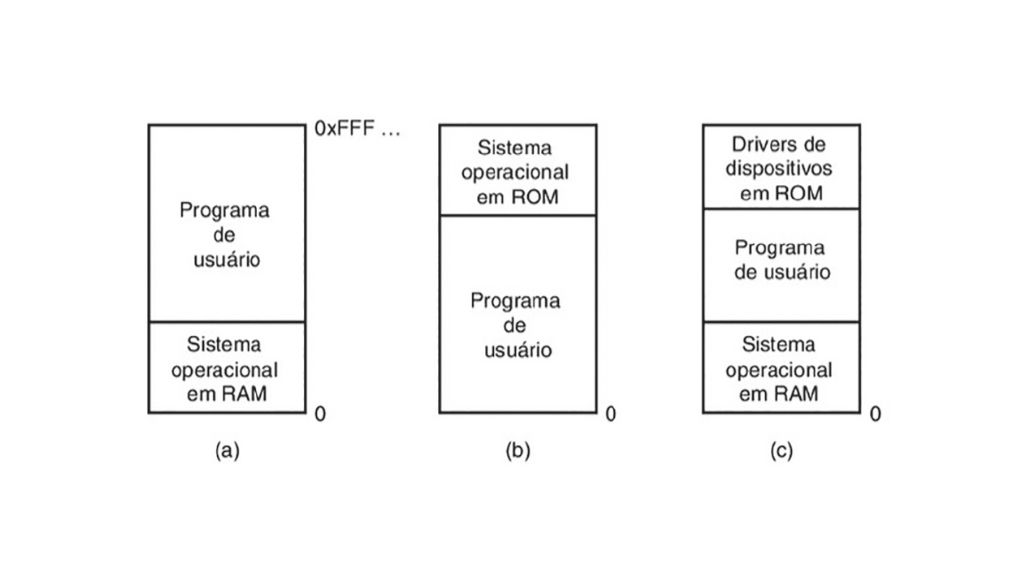


**Monoprogramação sem troca de processos ou paginação**

Segundo Tanenbaum (2003), este método de gerenciamento de memória é o mais simples, pois somente um programa é executado por vez, e a memória é compartilhada entre o sistema operacional e o programa.

A figura “Monoprogramação sem troca de processos ou paginação” apresenta três formas que podem ocorrer neste esquema:

* o sistema operacional está utilizando o espaço de endereçamento em RAM (figura (a)), modelo aplicado aos *mainframes* e minicomputadores.
* o sistema operacional está utilizando o espaço de endereçamento em ROM somente para a leitura (figura (b)), usado em alguns computadores de mão e em sistemas embarcados.
* os drives de dispositivos estão em ROM e os programas do usuário e o sistema operacional está em RAM (figura (c)) – modelo utilizado nos primeiros computadores pessoais (MS-DOS).

Monoprogramação sem troca de processos ou paginação. Fonte: Tanenbaum (2003, p. 140)

Segundo Machado e Maia (2007), a alocação contigua simples (outro termo usado para Monoprogramação sem Troca de Processos ou Paginação) foi utilizada nos primeiros sistemas operacionais e ainda está presente em alguns sistemas monoprogramáveis. O usuário tem o controle sobre toda a memória, podendo acessar qualquer posição dela e, até mesmo, a área do sistema operacional. Neste método, a utilização de recursos computacionais não é eficiente, pois somente um usuário tem acesso aos recursos.

Ainda segundo Machado e Maia (2007), todos os programas são limitados à área de memória principal disponível para o usuário. Para resolver essa limitação, utiliza-se uma técnica chamada *overlay* (sobreposição), que consiste em dividir o programa em módulos, sendo cada um executado independentemente, utilizando a mesma área de memória. Cada módulo de código ou dados só é sobreposto quando não for mais necessário utilizar o que estava naquele local na memória anteriormente. O *overlay* era definido pelo programador e foi utilizado no MS-DOS.

\_\_\_\_\_\_

**📝 Exemplificando**

Por exemplo, imagine que um programa tenha três módulos: um principal, um de cadastramento e outro de impressão. Os módulos de cadastramento e impressão são independentes.

Assim, quando o módulo de cadastramento estiver em execução na memória, o módulo de impressão não precisa estar presente. O modulo principal é comum aos módulos de cadastramento e impressão, e deve estar na memória durante o tempo de execução do programa (MACHADO; MAIA, 2007).

\_\_\_\_\_\_

**Multiprogramação com Partições Fixas**

Segundo Tanenbaum (2003), a maioria dos sistemas operacionais modernos permite que vários processos executem ao mesmo tempo (multiprogramação) e quando um processo é bloqueado aguardando uma informação de entrada/ saída, outro processo poderá utilizar a CPU, aumentando a sua utilização.

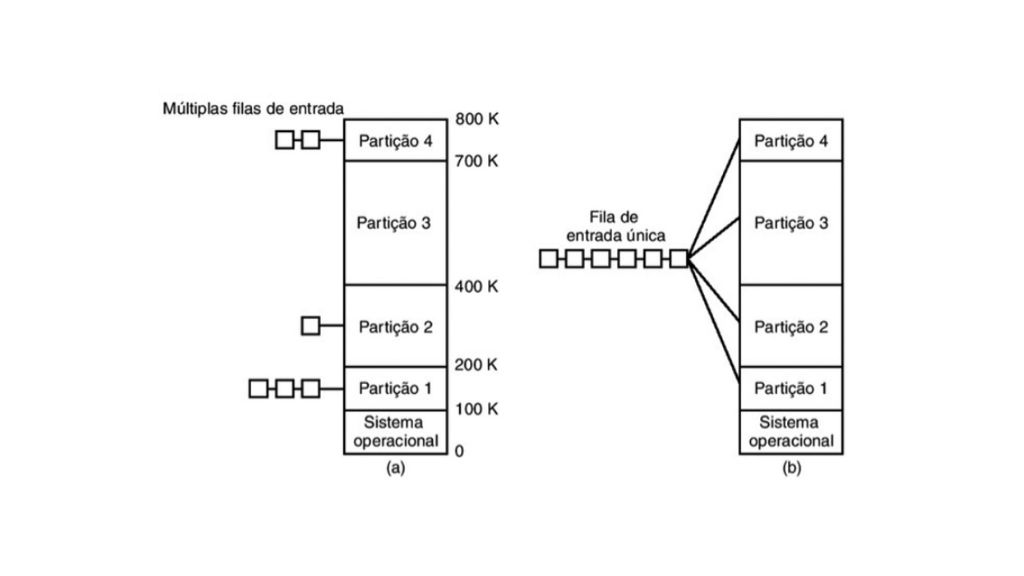
Nesse método, a memória é dividida em *n* partições de tamanhos diferentes, podendo ser definida quando o sistema for iniciado. Segundo Machado e Maia (2007), sempre que o tamanho de uma partição for alterado, o sistema deverá ser reiniciado com a nova configuração.

Segundo Tanenbaum (2003), quando um processo chega para ser executado, ele é inserido em uma fila associada à menor partição suficiente para armazená-lo. O espaço não utilizado pelo processo é perdido, uma vez que o tamanho das partições é fixo.

A figura “Partições fixas com (a) múltiplas filas de entrada e (b) com uma única fila” apresenta filas de entrada separadas para cada partição. Por exemplo, na partição 4 existem dois processos na fila, sendo que estes podem ter um tamanho de até 100 k para utilizar. A desvantagem deste método é que podem existir muitos processos aguardando para serem executados em algumas partições (como nas partições 1 e 4) e em outras filas não exista nenhum processo (como na partição 3).

Uma solução é implementar uma fila única, assim, gerando um processo mais próximo do início da fila e que caiba na partição para ser carregado e executado (figura “Partições fixas com (a) múltiplas filas de entrada e (b) com uma única fila”)). Para evitar desperdício de espaço não utilizado em uma partição, poder-se- ia percorrer a fila a procura do maior processo que comporte na partição.

Porém, processos pequenos não seriam selecionados para executar. Uma solução seria implementar uma partição pequena e definir que um processo não poderia aguardar para receber uma partiçã o mais do que ***n*** vezes.

Partições fixas com (a) múltiplas filas de entrada e (b) com uma única fila. Fonte: Tanenbaum (2003, p. 141).

**🔁 Assimile**

Vimos nesta aula como é importante gerenciar o uso da memória do computador. Nas linguagens de programação, o gerenciamento da memória é fundamental, pois a tendência das aplicações dos usuários é consumir cada vez mais este recurso. Em muitas linguagens de programação não é necessário se preocupar com o gerenciamento, porém é importante que sejam conhecidas as restrições e capacidades do gerenciador de memória para uma programação eficaz.

Por exemplo, nas linguagens de programação C/C++ é preciso realizar o gerenciamento de memória.

**Problemas da Multiprogramação: relocação e proteção**



Segundo Tanenbaum (2003), a multiprogramação apresenta dois problemas essenciais: **relocação** e **proteção**.

A **relocação**é a transferência de um código de um local para outro de forma que um programa não escreva na área de outro programa. A relocação é necessária uma vez que processos executam em diferentes endereços na memória física.

Como exemplo, podemos relatar o problema de relocação entre dois programas: quando o programa 1 está executando no endereço de memória entre 0 e 20. Após a finalização do programa 1, o programa 2 também foi alocado no endereço de memória entre 0 e 20, ambos com as suas instruções (ADD, MOV, JMP, dentre outras).

Os programas 1 e 2, ao serem relocados, foram alterados os seus endereços iniciais de memória. Ao iniciarem a execução de suas instruções com novos endereços, o programa 2, que possui uma instrução de saltar para o endereço 20, invade a área de memória do programa 1. Esta é uma operação ilegal, uma vez que os processos não podem ler ou escrever endereços que pertençam a outros processos/usuários.

Segundo Tanenbaum (2003), quando um programa é ligado (combinação do programa principal e procedimentos executados pelo usuário e procedimentos de biblioteca), o *linker*– responsável por gerar o código executável – tem que saber o endereço inicial do programa para relocar todas as instruções que fazem uso de um endereço relativo de memória quando o programa for executado.

Uma solução para o problema de relocação é alterar as instruções do programa segundo a partição de memória em que ele está carregado. No caso da relocação dos programas 1 e 2, a instrução contida no endereço 20, seria direcionada para um endereço de memória dentro da área dos endereços do programa para não comprometer sua execução.

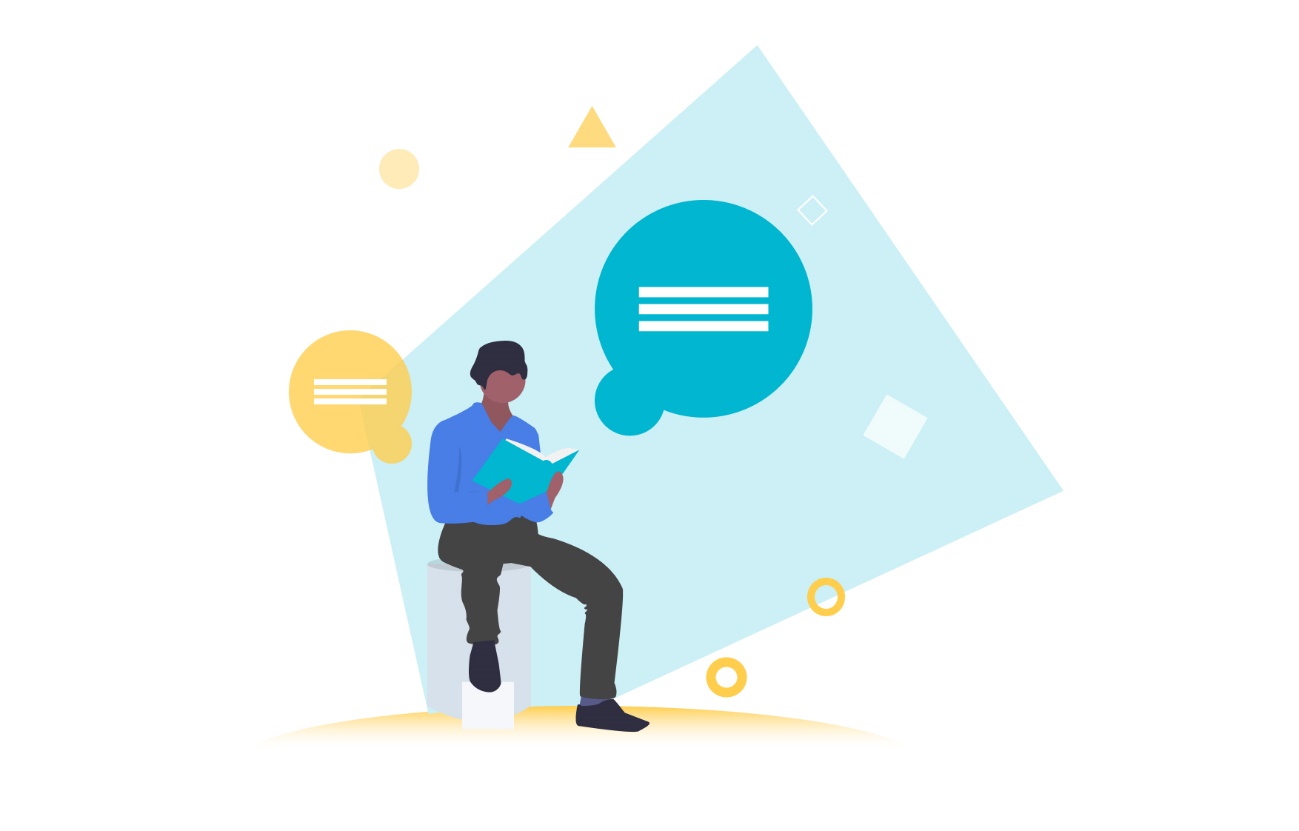
Assim, o *linker* deve inserir no código binário uma lista ou um mapa informando quais são os códigos de operação precisam ser relocados e quais não precisam.

Segundo Tanenbaum (2003), a relocação de um programa durante a carga não resolve a proteção. Diferentes aplicações não podem acessar dados que não estejam dentro de sua partição. Se não houvesse esta proteção, caso um programa mal-intencionado insira uma nova instrução, ele poderá desviar o controle para ela, ocasionando uma falha de proteção.

Tanto os problemas de relocação quanto de proteção podem ser resolvidos fornecendo o processador com dois registradores (memória localizada no processador): base e limite. O registrador é uma memória localizada no processador do computador e acelera a execução das aplicações. Ele está no topo da hierarquia de memória, possuem baixa capacidade de armazenamento, alta velocidade de processamento, porém são caros.

Quando um processo é escalonado, o registrador-base é carregado com o endereço de início da partição e o registrador-limite com o tamanho da partição. Os endereços gerados são verificados no registador-limite garantindo que não seja possível endereçar memória fora da região alocada ao processo durante a execução.

**Conclusão**



Agora que você já aprendeu sobre como é realizado o gerenciamento de memória nos sistemas operacionais e suas características e sobre os problemas da multiprogramação, vamos relembrar o nosso contexto. Durante a feira um funcionário de uma empresa do ramo de produtos de beleza, responsável pelo setor de T.I., procura seu *stand* interessado em substituir os computadores utilizados na empresa.

Assim, surgem os seguintes questionamentos:

* qual a solução para este cliente referente ao gerenciamento de memória dos computadores de sua empresa?
* qual estratégia poderá ser adotada para obter um bom desempenho de memória considerando a situação atual da empresa?

No caso deste cliente é necessário fazer um estudo em sua empresa para definir qual será a melhor estratégia para o seu negócio. O cliente relatou que processa um grande volume de informações, e a memória dos computadores não consegue processar a quantidade de dados geridos na empresa. Neste caso, é preciso ver quais programas e aplicações são realmente necessários nas máquinas da empresa para não sobrecarregar o processamento dos dados principais do negócio do cliente. Estas aplicações extras instaladas nas máquinas podem fazer com que o computador fique lento ou pare de executar.

Uma solução inicial, é a aquisição de mais memória RAM, pois a maior quantidade permitirá que mais programas e aplicações sejam executados. A memória RAM auxilia o processador na transferência dos dados entre o disco e a memória. Porém o aumento da memória nem sempre acelerará o desempenho da máquina.

Pode ser que o processador dos computadores não seja adequado para a quantidade de memória existente, não conseguindo endereçar um volume tão grande de memória RAM. Os processadores mais antigos não suportam uma memória maior que 3 GB. Talvez seja necessário realizar a troca do sistema operacional que contenha um processador mais eficiente que consiga atender a uma maior quantidade de memória.

Ainda segundo o relato do cliente, em algumas situações os dados foram sobrescritos. Pode ter acontecido um atraso na atualização dos dados pela rede e, assim, o último arquivo salvo sobrescreveu os arquivos existentes. Outra análise a ser feita é verificar como é realizada a relocação dos arquivos do computador.

O problema de relocação altera a partição (endereços) de memória em que ele está carregado. Mesmo os processos não podendo ler ou escrever endereços que pertençam a outros processos/usuários se tiver ocorrido algum problema na rede, por exemplo, o último processo que acessou a memória, invadiu o endereço de memória de outro processo apagando os dados. Verifique se os registradores do computador estão inibindo que outros processos aloquem memória em regiões que estejam sendo utilizadas.

As soluções apresentadas auxiliam no desempenho de memória das máquinas e, consequentemente, o cliente não terá problemas. Porém, mesmo após todas as soluções apresentadas os computadores continuarem não suportarem a quantidade de dados processados, será necessário a troca das máquinas para outras mais potentes, para que não ocorra prejuízos nos negócios da empresa do cliente.