

A escrita de registros em um arquivo segue caminho contrário. À medida que registros são escritos no arquivo, eles vão sendo colocados em posições contíguas de memória na área de armazenamento. Quando a área de armazenamento não possui espaço suficiente para armazenar mais um registro, o bloco é copiado para a memória secundária, deixando a área de armazenamento vazia e pronta para receber novos registros.

A técnica de utilização de áreas de armazenamento evita que um processo que esteja realizando múltiplas transferências de dados de forma sequencial tenha de ficar esperando que as transferências se realizem para prosseguir o processamento. As transferências são realizadas em blocos pelo sistema operacional diretamente para uma área de armazenamento. O processo usuário pega o dado nessa área e somente é obrigado a esperar quando a área se esvazia. Quando isso ocorre, o sistema operacional enche novamente a área e o processo continua. Essa técnica pode ser aprimorada com o uso de duas ou mais áreas de armazenamento. Nesse caso, enquanto um processo está operando em uma área, o sistema operacional enche a outra.

3. Para desenvolver um método de pesquisa eficiente, o aspecto sistema de computação é da maior importância. As características da arquitetura e do sistema operacional da máquina tornam os métodos de pesquisa dependentes de parâmetros que afetam seus desempenhos. Assim, a transferência de blocos entre as memórias primária e secundária deve ser tão eficiente quanto as características dos equipamentos disponíveis o permitem. Tipicamente, a transferência torna-se mais eficiente quando o tamanho dos blocos é de 512 *bytes* ou múltiplos deste valor, até 4.096 *bytes*.

Na próxima seção, apresentamos um modelo de computação para memória secundária que transforma o endereço usado pelo programador no endereço físico alocado para o dado a ser acessado. Esse mecanismo é utilizado pela maioria dos sistemas atuais para controlar o trânsito de dados entre o disco e a memória principal. A seguir, apresentamos o método de acesso sequencial indexado e mostramos sua utilização para manipular grandes arquivos em discos ópticos e apenas leitura. Finalmente, apresentamos um método eficiente para manipular grandes arquivos em discos magnéticos: a árvore *n*-ária de pesquisa.

## 6.1 Modelo de Computação para Memória Secundária

Esta seção apresenta um modelo de computação para memória secundária conhecido como **memória virtual**. Esse modelo é normalmente implementado como uma função do sistema operacional. Uma exceção é o sistema operacional DOS para microcomputadores do tipo IBM-PC, que, apesar de muito vendido no mundo inteiro, não oferece um sistema de memória virtual. Por essa razão, vamos apresentar o conceito e mostrar uma das formas possíveis de se implementar um

sistema de memória virtual. Além disso, o conhecimento de seu funcionamento facilita a implementação eficiente dos algoritmos para pesquisa em memória secundária também em ambientes que já ofereçam essa facilidade. Mais detalhes sobre este tópico podem ser obtidos em livros da área de sistemas operacionais, tais como Lister (1975), Peterson e Silberschatz (1983) e Tanenbaum (1987).

### 6.1.1 Memória Virtual

A necessidade de grandes quantidades de memória e o alto custo da memória principal têm levado ao modelo de sistemas de armazenamento em dois níveis. O compromisso entre velocidade e custo é encontrado com o uso de uma pequena quantidade de memória principal (até 640 *kilobytes* em microcomputadores do tipo IBM-PC usando sistema operacional DOS) e de uma memória secundária muito maior (vários milhões de *bytes*).

Como apenas a informação que está na memória principal pode ser acessada diretamente, a organização do fluxo de informação entre as memórias primária e secundária é extremamente importante. A organização desse fluxo pode ser realizada utilizando-se um mecanismo simples e elegante para transformar o endereço usado pelo programador na correspondente localização física de memória. O ponto crucial é a distinção entre *espaço de endereçamento* (endereços usados pelo programador) e *espaço de memória* (localizações de memória no computador). O espaço de endereçamento *N* e o espaço de memória *M* podem ser vistos como um mapeamento de endereços do tipo

$$f: N \rightarrow M.$$

O mapeamento de endereços permite ao programador usar um espaço de endereçamento que pode ser maior que o espaço de memória primária disponível. Em outras palavras, o programador enxerga uma memória virtual cujas características diferem das características da memória primária.

Existem várias formas de implementar sistemas de memória virtual. Um dos meios mais utilizados é o sistema de paginação, no qual o espaço de endereçamento é dividido em **páginas** de igual tamanho, em geral múltiplos de 512 *bytes*, e a memória principal é dividida de forma semelhante em **Molduras de Páginas** de igual tamanho. As Molduras de Páginas contêm algumas páginas ativas enquanto o restante das páginas estão residentes em memória secundária (páginas inativas). O mecanismo de paginação possui duas funções, a saber:

- a) realizar o mapeamento de endereços, isto é, determinar qual página um programa está endereçando, e encontrar a moldura, se existir, que contenha a página;
- b) transferir páginas da memória secundária para a memória primária quando do necessário, e transferi-las de volta para a memória secundária quando não são mais utilizadas.