

ii) A probabilidade de que a página segura mais profunda esteja no primeiro nível da árvore.

iii) O valor da taxa de utilização de memória.

6. Neste trabalho você deve apresentar uma implementação do conjunto de procedimentos para criação de um ambiente de memória virtual paginada em Pascal, conforme descrito na Seção 6.1.2.

O conjunto de procedimentos deverá permitir ao usuário incorporar facilmente um ambiente de memória virtual ao seu programa para poder organizar o fluxo de dados entre a memória primária e a memória secundária. Para isso, procure colocar todos os procedimentos e declarações de tipos em um único arquivo chamado SMV.PAS. Esse arquivo poderá ser incorporado a qualquer programa escrito em Pascal, o qual deverá aparecer antes das declarações de variáveis do usuário.

O tamanho máximo de cada estrutura de dados utilizada pelo sistema deverá ser definido por meio de constantes que poderão facilmente ser ajustadas pelos usuários diretamente no arquivo SMV.PAS, de acordo com suas conveniências.

O que cada aluno deve fornecer:

- Uma listagem completa do conjunto de procedimentos precedida de documentação pertinente. A descrição de cada procedimento deverá conter pelo menos a sua função e a de seus parâmetros. Dependendo da complexidade de cada procedimento, pode ser interessante descrever sucintamente a lógica do módulo obtido (evite descrever o que é óbvio).
- Uma listagem de um programa de demonstração (DEMO) que mostre claramente ao usuário como utilizar o pacote SMV.PAS. O programa DEMO deve servir também para mostrar toda a flexibilidade e potencial do SMV.PAS.
- Teste do Sistema.

Para testar os vários módulos do sistema de paginação, você deve gerar um arquivo em disco contendo algumas páginas (para fins de teste você pode utilizar uma página de tamanho pequeno, digamos 32 *bytes*). O arquivo de teste em disco deverá conter as páginas de uma árvore binária de pesquisa sem balanceamento, conforme mostrado no Programa 6.1.

É importante criar procedimentos para mostrar o conteúdo de todas as páginas em disco, da fila de Moldura de Páginas e da Tabela de Páginas. Esses procedimentos devem ser chamados pelo programa de teste nos momentos mais interessantes para se verificar o comportamento do sistema (talvez seja interessante realizar uma adaptação do programa DEMO para fins de testar o SMV conforme descrito anteriormente). A impressão de todos esses momentos deve ser fornecida junto à listagem do programa de teste.

7. O objetivo deste trabalho é projetar e implementar um sistema de programas para recuperar, inserir e retirar registros de um arquivo que pode conter milhões de registros. A aplicação que utiliza o arquivo é bastante dinâmica, existindo um

grande número de consultas e atualizações (inserções, retiradas e alterações de registros). Além do mais, a aplicação requer periodicamente a recuperação total ou parcial dos registros na ordem lexicográfica das chaves. Essas características sugerem fortemente a utilização de **árvore B** como estrutura de dados. O que fazer:

a) Para implementar os algoritmos de pesquisa, inserção e retirada em uma árvore B de ordem m , utilize o pacote SMV.PAS proposto em exercício anterior, para criar um ambiente de memória virtual e resolver o problema de fluxo de dados entre as memórias primária e secundária.

b) Para testar o seu sistema de programas para uma árvore B de ordem $m = 2$, utilize a seguinte sequência de chaves:

Inserção:

20; 10 40 50 30; 55 3 11 4 28; 36; 33 52 17 25 13; 45 9 43 8 48;

A cada ponto e vírgula você deverá imprimir a árvore.

Retirada:

45 30 28; 50 8; 10 4 20 40 55; 17 33 11 36; 3 11 52;

A cada ponto e vírgula você deverá imprimir a árvore.

c) Para ter uma ideia da eficiência do método de acesso construído, faça a medida do tempo de execução para:

i) Construir árvores B de tamanhos 1.000, 10.000 e 50.000 chaves geradas randomicamente por meio de um gerador de números aleatórios. A medida de tempo deve ser tomada de forma independente para cada uma das três árvores. A ordem m das árvores deve ser tal que o tamanho da página seja igual a ou menor do que 512 *bytes*.

ii) Para a maior das três árvores que você conseguiu construir, gere aleatoriamente um conjunto de 200 chaves e realize uma pesquisa na árvore para cada chave. Caso haja chaves que não estejam presentes no arquivo, informe quantas pesquisas foram com sucesso e quantas foram sem sucesso. Com essa medida podemos ter uma ideia do tempo aproximado para pesquisar um registro no arquivo.

Atenção: procure interpretar os resultados obtidos. Por exemplo, você deve informar qual foi o número de Molduras de Páginas utilizadas para os experimentos acima (com 256 *Kbytes* de memória real disponível é possível manter cerca de 120 molduras na memória principal). Quaisquer outras observações relevantes devem ser relatadas.

Observações:

a) A pesquisa da chave de um registro dentro de uma página da árvore B pode ser feita por meio de uma pesquisa sequencial.

b) A decisão sobre a documentação a ser apresentada fica por conta do aluno.