

Instituto de Computação da UNICAMP

Disciplina MC202: Segundo Semestre de 2014

Laboratório Nº 08

Prof. Tomasz Kowaltowski

O objetivo deste laboratório é a implementação e teste de uma base de dados semelhante à proposta no Laboratório 07. Desta vez, a base de dados será implementada através de uma **árvore AVL**. A implementação seguirá novamente a idéia de um tipo abstrato de dados realizado através de um par de arquivos `base.h` e `base.c` e testado por um programa `principal.c`. O programa principal aceita os seguintes comandos:

i	<i>ra</i>	insere novo registro com <i>ra</i> e <i>nome</i> na base;
c	<i>ra</i>	consulta registro com <i>ra</i> na base;
d		imprime a base sob forma gráfica;
r	<i>ra</i>	remove registro com <i>ra</i> da base;
l		libera espaço ocupado pela base;
n		imprime o número de elementos na base;
a		imprime a altura da base (árvore binária);
p		executa um percurso inordem na base, imprimindo os elementos;
h		imprime resumo dos comandos;
x ou q		encerra interpretação;
#		linha de comentário.

Observações:

1. Um **RA** sempre será **representado por um número inteiro** e **não** é precedido das letras RA.
2. A representação da base está escondida do usuário que deve enxergá-la apenas como um *tipo abstrato de dados*. Note no arquivo `base.h` que isso é feito em C definindo o tipo `Base` como um apontador do tipo `void`. A declaração do tipo *árvore AVL* ficará escondida no arquivo `base.c` onde as conversões de tipo apropriadas (*casting*) deverão ser efetuadas.
3. A operação de remoção é **opcional** e vale até 2 pontos extra na **versão A** desta tarefa (até 12 pontos), sendo que a média final máxima de laboratório (todas as tarefas) é 10.
4. A versão B não testa a função de remoção e vale no máximo 10.
5. Se for feita submissão nas duas versões, valerá aquela que produzir resultados corretos. Se ambas submissões produzirem resultados corretos, valerá a versão A.
6. Para as operações de inserção e remoção da base de dados é importante que sejam seguidas as idéias explicadas em aula e expostas nas transparências e na apostila. Para a remoção, em caso de ajuste na árvore deverá ser usado o sucessor no percurso em inordem como substituto do nó removido (mínimo da subárvore direita).
7. Devem ser utilizadas as macros `MALLOC` e `FREE` fornecidas com o pacote *balloc*. A sua implementação não pode utilizar as operações habituais de alocação.
8. A função *VerificaBase* declarada em `principal.c` é invocada pelo programa principal imediatamente após uma inserção ou remoção e percorre a base conferindo as restrições de árvores AVL. Algumas funções (*SubEsq*, *SubDir*, *FatorBal*) foram incluídas na especificação somente para permitir esta verificação.
9. A função *ImprimeBase* exibe a árvore binária "deitada", com a raiz do lado esquerdo. A função é fornecida pronta no arquivo incompleto `base.c` e utiliza o pacote *imprimearvore* também fornecido na íntegra. A função é chamada pelo comando **d** explicado acima. Inserção deste comando em mais pontos dos arquivos de teste pode facilitar a depuração.
10. O número máximo de submissões é 10.