

ANÁLISE DE DESEMPENHO DE TABELAS DE SÍMBOLOS

Y. KOHAYAKAWA

Este desafio é mais um “exercício”, isto é, embora haja espaço para várias idéias criativas, o básico não envolve nenhuma dificuldade maior (apenas um pouco de trabalho). Entretanto, acredito que este desafio será muito instrutivo para aqueles que o atacarem. Neste desafio, você deve elaborar um *esquema de teste comparativo de desempenho para as várias implementações de tabelas de símbolos que estudaremos ao longo do semestre*.

Por exemplo, uma base para tal teste seria o seu ep1. Naquele EP, você implementou a tabela de símbolos de forma modular. Portanto, cada vez que estudarmos uma nova implementação de tabelas de símbolos, você pode substituir o seu ST.c de seu ep1 para obter uma nova versão do seu EP.

Um outro exemplo instrutivo é o programa de indexação de textos, disponível em http://www.ime.usp.br/~yoshi/2001i/mac323/exx/Text_index.

A sua tarefa. Elabore um documento que descreva (i) seu procedimento para analisar o desempenho de seus programas e (ii) seus resultados com várias implementações de tabelas de símbolos.

Metodologia e literatura de suporte. Você certamente achará útil estudar com cuidado as análises de desempenho apresentadas no livro do Sedgewick [*Algorithms in C*, 3a. edição, vol. 1, Addison Wesley Longman, 1998].

Outros livros e material de suporte muito importantes são os seguintes:

1. Um esquema refinado de teste de desempenho proposto por D.E. Knuth envolve a contagem do número de acessos à memória. Para este tópico, você pode consultar algum colega que já tenha feito a disciplina MAC328, *Algoritmos em Grafos*. Esta disciplina, nos últimos 5 anos, tem sido baseada em parte no *Stanford GraphBase*. (Veja também a página do professor Paulo Feofiloff sobre o SGB.) Seria excelente você estudar algumas implementações de tabelas de símbolos usando a metodologia de Knuth.
2. Um livro muito instrutivo sobre programação em geral e eficiência em particular é o *Programming Pearls*, de Jon Bentley, da Bell Labs. Em particular, na Coluna 9 há uma discussão sobre *code tuning* (ajustes finos de programas) e a técnica de *profiling*.
3. Finalmente, você também pode consultar outro excelente livro, *The Practice of Programming*, de Brian W. Kernighan e Rob Pike. Os Capítulos 2 e 7 deste livro são de particular interesse para nós.

Observação. Você pode entregar seu relatório em partes, pois estudaremos implementações novas até o fim do semestre. Entretanto, você deve de qualquer forma produzir um relatório final, contendo uma análise comparativa geral.

Avaliação. Na avaliação de seu relatório, estarei particularmente interessado na metodologia que você adotou para medir e comparar o desempenho de seus programas. Ademais, também estarei muito interessado em seus resultados e conclusões.