

Trabalho Prático

Laboratórios de Informática

Grupo 6

Guilherme Viveiros , Mateus Silva e
Angelo André

Contents

1	Motivação:	1
2	Processamento de dados:	1
3	Estruturas principais	2
4	Gestão de memória:	2
5	Implementações:	3
6	Auxiliares:	3
7	Modularização:	4
8	Convenções para entradas inválidas:	4
9	Referências:	5

1 Motivação:

No âmbito da Unidade Curricular Sistemas Operativos, foi proposto pelo corpo docente a implementação de um processador de notebooks.

Processador de NoteBooks tem como objetivo misturar fragmentos de código, resultados da execução, e documentação. Neste contexto, um notebook é um ficheiro de texto que depois de processado é modificado de modo a incorporar resultados da execução de código ou comandos nele embebidos.

POR IMAGEM DEMOSNTRATIVA AQUI

2 Processamento de dados:

Os dados são processados através de ficheiros de leitura que são fornecidos.

3 Estruturas principais

Decidimos que seria uma boa implementacao usarmos um Buffer com um growing array nele contido

4 Gesto de memria:

Para assegurar que no h perdas de memria nas diferentes partes do projeto usamos o software Valgrind¹, que indica os blocos de memria que foram reservados pela aplicaco mas que no contm nenhuma varivel com o endereo desta.

¹<http://valgrind.org/>

5 Implementações:

Dado a nossa estrutura não foi muito complicado devolver o ficheiro final. O array é sequencialmente percorrido ,através dos métodos disponibilizados, ocorrendo uma verificação para ver se é comando ou apenas uma documentação , caso fosse comando o resultado desse comando (disponibilizado no array) era escrito num ficheiro , limitado entre o nome do comando , »» resultado do comando ««.

Caso não fosse comando era copiada a linha do array para o o ficheiro.

6 Auxiliares:

FALAR NAS FUNCOES USADAS PARA CONSEGUIR EXECUTAR COMANDOS MAIS COMPLEXOS

7 Modularização:

Todas as componentes do nosso trabalho, foram criadas em ficheiros fonte separados e com atributos privados, para que fosse impossível aceder a estes sem usar os métodos diretamente associados a cada uma das estruturas de dados criadas.

A legibilidade e a flexibilidade do nosso projeto permitiu testar todas os diferentes componentes dos módulos criados sem que fosse necessário alterar as centenas de linhas externas que usufruem destes.

8 Convenções para entradas inválidas:

Dado que o utilizador pode usar comandos inválidos ou até parar o programa com CTRL C as convenções para entradas inválidas foram simplesmente parar o programa e devolver o ficheiro recebido intacto.

9 Referências:

[1] - J.W. J. Williams, Algorithm 232: Growing-Array, Communications of the ACM 7 (1964), 347-348.

[2] - R.W. Floyd, Algorithm 245: Treesort 3, Communications of the ACM 7 (1964), 701 NAO ESTOU A USAR AQUI

[3] - D. B. Johnson, Priority queues with update and finding minimum spanning trees, Information Processing Letters 4 (1975), 53-57. NAO ESTOU A USAR AQUI

[4] - Willams, Jr., Louis F. , A modification to the half-interval search (binary search) method , Proceedings of the 14th ACM Southeast Conference (1975) , 95-101.NAO ESTOU A USAR AQUI