

Etapa 1 de Laboratórios de Informática II

Licenciatura de Engenharia Informática

Ano Lectivo 09/10

Conteúdo

1 Critérios de Aprendizagem	1
2 Enunciado	1
3 Interface por Linha de Comando	2
4 Avaliação	3
4.1 Material a entregar para cada uma das etapas	3
4.2 Relatório	3

1 Critérios de Aprendizagem

No final desta etapa os alunos deverão ser capazes de:

1. Compilar programas em código C utilizando o `gcc` e correr o executável gerado pelo compilador;
2. Utilizar intruções condicionais, ciclos e funções;
3. Armazenar e percorrer matrizes;
4. Utilizar a biblioteca de input/output do C para ler e escrever ficheiros de texto;
5. Fazer algum processamento básico sobre *strings*.

2 Enunciado

1. Desenvolver um programa que compila em `gcc` que implementa uma interface por linha de comando;
2. Implementar os seguintes comandos **sem qualquer tipo de variação na sintaxe**:

<code>gr nome_do_ficheiro</code>	Gravar o estado do tabuleiro num ficheiro com o formato LTR (ao escrever <code>gr xpto</code> deverá guardar o estado no ficheiro <code>xpto.ltr</code>);
<code>cr nome_do_ficheiro</code>	Carregar o estado do tabuleiro a partir dum ficheiro com o formato LTR (ao escrever <code>cr xpto</code> deverá ir buscar o estado ao ficheiro <code>xpto.ltr</code>);
<code>b x y</code>	Colocar a célula com as coordenadas (x,y) com a cor <i>branca</i> ;
<code>p x y</code>	Colocar a célula com as coordenadas (x,y) com a cor <i>preta</i> ;
<code>i x y</code>	Colocar a célula com as coordenadas (x,y) com a cor <i>indefinida</i> ;
<code>q</code>	Sai do programa.

3 Interface por Linha de Comando

Uma interface por linha de comando é algo tipo DOS ou shell que escreve um prompt, espera o comando do utilizador, interpreta o comando, executa-o e mostra o resultado. Seguidamente, volta ao início. No nosso caso concreto, após a invocação de cada comando (excepto o comando `q`), o programa deve mostrar o estado do tabuleiro.

Segue-se um exemplo que assume que o prompt é `'Letrorium> '`:

```
Letrorium> cr brd1
A (B) A B
(C) [B] (A) D
D (A) C D
(A) [B] (D) C
Letrorium> b 1 3
A (B) A B
(C) [B] (A) D
(D) (A) C D
(A) [B] (D) C
Letrorium> p 4 3
A (B) A B
(C) [B] (A) D
(D) (A) C [D]
(A) [B] (D) C
Letrorium> gr brd1sol
A (B) A B
(C) [B] (A) D
(D) (A) C [D]
(A) [B] (D) C
Letrorium> q
```

Repare que o primeiro comando carregou o estado a partir do ficheiro `brd1.ltr`; o segundo e o terceiro modificaram o estado do tabuleiro; o quarto gravou o estado actual no ficheiro `brd1sol.ltr` e o último fez o programa terminal e por isso não voltou a imprimir o estado do tabuleiro.

4 Avaliação

Esta etapa vale no máximo 3 valores e deve ser entregue até ao fim do dia 7 de Março no site da disciplina. Para não reprovar à disciplina os alunos precisam de tirar um mínimo de 50% da classificação desta etapa (1.5 valores). A entrega atrasada incorre numa penalização de 10% ao dia.

A avaliação é feita durante as aulas da semana seguinte. Caso um aluno não apareça à avaliação sem apresentar motivos de força maior devidamente justificados por documentos este reprova automaticamente à disciplina por faltar à avaliação.

4.1 Material a entregar para cada uma das etapas

O grupo deverá entregar em cada etapa um arquivo criado com o comando `tar` e comprimido com o algoritmo de compactação `bzip2` contendo a seguinte informação:

code	Uma pasta com todo o código fonte do trabalho que deverá incluir a <code>makefile</code> utilizada para compilar todo o código e gerar a documentação;
doc	Uma pasta com a documentação em formato <code>html</code> gerada a partir do código utilizando a ferramenta <code>Doxygen</code> ;
relatorio.pdf	Um ficheiro gerado utilizando o comando <code>L^AT_EX</code> no formato <code>pdf</code> com o relatório;
doc.pdf	Um ficheiro gerado utilizando o comando <code>L^AT_EX</code> a partir dos ficheiros gerados pela ferramenta <code>Doxygen</code> .

O arquivo deverá utilizar o seguinte arquétipo para o nome:

`g<número do grupo><turno>-et<número da etapa>.tar.bz2`

Por exemplo, o ficheiro entregue na segunda etapa pelo grupo número 3 do turno PL1 teria como nome `g3PL1-et2.tar.bz2`.

4.2 Relatório

Cada etapa deve vir acompanhada de um relatório escrito em `LATEX` e do código documentado. O relatório deve necessariamente ter as seguintes secções:

Resumo	Onde se apresenta um breve resumo do relatório que não deverá ultrapassar as 250 palavras;
---------------	--

Introdução	Onde se apresenta a introdução do relatório, tipicamente, deverá ter subsecções como motivação, objectivos e estrutura do relatório;
Desenvolvimento	O desenvolvimento poderá ser mais do que uma secção, no caso deste relatório deverá explicar as opções tomadas e apresentar breves algoritmos que ajudem a compreender as partes críticas do código;
Conclusão	Onde se apresentam as conclusões objectivas do trabalho efectuado;
Bibliografia	Onde se citam as referências bibliográficas utilizadas no trabalho.

Lembre-se que o relatório pretende explicar quais foram as dificuldades encontradas e como estas foram resolvidas. A apresentação deverá ser técnica de forma a ajudar o avaliador a perceber o que foi feito e como foi feito.

É obrigatório apresentar a documentação gerada automaticamente através do código comentado utilizando o **Doxygen** <http://www.stack.nl/~dimitri/doxygen>. Todas as funções deverão ser documentadas com uma pequena descrição que ilustre o seu funcionamento. Assim, não caia na tentação de descrever as várias funções no relatório já que o resultado do **Doxygen** produzirá um documento com essa informação. Deverá configurar o **Doxygen** para gerar documentação para todas as entidades para garantir que não se esquece de nenhuma.