

Universidade do Minho
Licenciatura em Engenharia Informática



Universidade do Minho

Laboratórios de Informática II

Ano Lectivo de 2009/2010

Etapa 1

54738 **João Gomes**
54745 **André Pimenta**
54764 **Nelson Vieira**

7 de Março de 2010

Resumo

Neste relatório está exposta a realização da 1ª Etapa do Projecto da Unidade Curricular de Laboratórios de Informática II, do primeiro ano da LEI.

Esta consiste no desenvolvimento de uma interface por linha de comandos utilizando a linguagem de programação C e, durante a sua realização, foram testados os nossos conhecimentos básicos sobre referida linguagem, tais como a compilação de programas utilizando o gcc, a utilização de instruções condicionais, ciclos e funções, a manipulação de informação em matrizes e a leitura/ escrita em ficheiros.

Conteúdo

Conteúdo	1
1 Introdução	2
1.1 Contextualização e apresentação do Caso de Estudo	2
1.2 Motivação e objectivos	2
1.3 Estrutura	2
2 Desenvolvimento	3
2.1 Análise do Problema	3
2.2 Solução do Problema	4
2.3 Ferramentas utilizadas	10
3 Conclusão	11
4 Bibliografia e referências WWW	12

Capítulo 1

Introdução

1.1 Contextualização e apresentação do Caso de Estudo

Este documento descreve os vários passos dados pelo grupo durante a realização da primeira etapa do projecto da disciplina Laboratórios de Informática II, na qual era objectivo desenvolver-se uma interface por linha de comando utilizando a linguagem C.

1.2 Motivação e objectivos

Todos os elementos do grupo partiram para a realização desta etapa motivados pelo trabalho a desenvolver. Como todos tínhamos já alguns conhecimentos sobre a linguagem de programação C, a auto-exigência a que nos propusemos foi elevada. Além disso, o facto de a tarefa consistir no desenvolvimento de um interface gráfico torna-a um problema algo interessante de se resolver.

Esta etapa tem como objectivo mais importante introduzir nos alunos os conhecimentos básicos sobre a linguagem de programação C, qe já foram referidos no resumo.

1.3 Estrutura

Este relatório é constituído por três capítulos: Introdução; Desenvolvimento (dividido em duas secções, “Análise do Problema”, “Solução do Problema” e “Ferramentas Utilizadas”), onde constam os passos mais importantes que foram dados pelo grupo durante a realização da tarefa; Conclusão; e por fim, a Bibliografia.

Capítulo 2

Desenvolvimento

2.1 Análise do Problema

Uma interface por linha de comandos é algo do tipo DOS que escreve um prompt, espera por um comando inserido pelo utilizador, interpreta-o, e executa-o, exibindo o resultado.

No caso concreto do nosso projecto, o objectivo era desenvolver uma espécie de jogo, que, lendo um tabuleiro no formato LTR, recebesse um comando com uma cor e duas coordenadas, e “pintasse” o local do tabuleiro definido da cor escolhida.

Os comandos que o interface deve interpretar são os seguintes:

gr “nome do ficheiro”, que grava o estado actual do tabuleiro no formato LTR (i. e., se o ficheiro do tabuleiro se chamar brd3, grava-o como brd3.ltr);

cr “nome do ficheiro”, que carrega um tabuleiro de acordo com o nome introduzido;

b x y, que transforma em branca a posição dada pela linha x e pela coluna y;

p x y, que transforma em preta a posição dada pela linha x e pela coluna y;

i x y, que deve tornar “indefinida” a cor da posição dada pela linha x e pela coluna y;

q, comando que faz o programa terminar;

2.2 Solução do Problema

Descreveremos nesta parte a forma em como ultrapassámos as dificuldades que foram surgindo ao longo do desenvolvimento do projecto. As partes do código desenvolvido que julgamos mais importantes para o funcionamento do interface serão também aqui explicadas detalhadamente.

Em primeiro lugar, apresentamos o tipo de estruturas usado para o projecto:

```
typedef struct elemento{  
  
    char letra;  
    char cor;  
  
} Elemento,*Elem;
```

Optámos por este tipo de estrutura pois achamos que para o problema em causa seria a mais adequada, visto tratar-se de um tabuleiro caracterizado por uma letra e por a cor associada à célula dessa mesma letra.

Explicuemos agora a principal função do nosso programa, a função **"main"**.

```
int main(){  
  
    char *line;  
  
    system("clear");  
    while(1){  
  
        line = readline ("Letrorium> ");  
        add_history (line);  
        system("clear");  
        executa_comando(line);  
        if(Tab!=NULL) printTabuleiro();  
        printf("\n\n");  
    }  
  
    return 0;  
}
```

Esta função é um ciclo que termina apenas quando o utilizador insere o comando "q", que, como já foi dito, faz sair do jogo.

A função é então responsável por receber os outros comandos continuamente e imprimir o tabuleiro enquanto o jogo decorrer.

Explicaremos agora outra das mais importantes funções do programa, a "executa_comando". Esta recebe como argumento a "linha" introduzida na shell.

```
int executa_comando(char *linha) {
    char cmd[1025];
    char args[1025];

    sscanf(linha, "%s %[^\\n]", cmd, args);

    if(strcmp(cmd, "b") == 0 || strcmp(cmd, "p") == 0 || strcmp(cmd, "i") == 0)
        return executaJogada(args, cmd);
    if(strcmp(cmd, "cr") == 0)
        return executaCriaJogo(args);
    if(strcmp(cmd, "gr") == 0)
        return executaGravaJogo(args);
    if(strcmp(cmd, "?") == 0)
        return ajuda();
    if(strcmp(cmd, "q") == 0)
        exit(1); // perguntar ao prof como é a melhor forma para exit

    fprintf(stderr, "Comando invalido, passara saber quais os comandos suportados digite: ? \\n");
    return -1;
}
```

É esta função qe reage aos comandos inseridos pelo utilizador.

Como já foi referido atrás, os nossos conhecimentos sobre a linguagem C era relativamente bons á partida para esta etapa. No entanto, esta função causou-nos algumas dificuldades, nomeadamente, a separar correctamente as palavras introduzidas na linha da shell.

Para ultrapassarmos esta questão, contámos com a ajuda do docente responsável, que nos forneceu o código contendo o "sscanf", o que facilitou sobremaneira a separação das palavras, permitindo assim que os comandos fossem executados com sucesso.

De seguida, explicamos a função ”**executaCriaJogo**“.

```
int executaCriaJogo(char *args) {
    char buf[1024];
    char nome[55];

    if(strlen(args)>50){
        fprintf(stderr,"Ficheiro pretendido demasiado grande,só pode carregar ficheiros até 50 chars\n");
        return 0;
    }

    int nargs = sscanf(args, "%s %[^\\n]", nome, buf);

    if(nargs != 1) {
        fprintf(stderr, "Argumentos passados inválidos. para mais informações consulte a ajuda.\\n");
        return -1;
    }

    strcat(nome, ".ltr");
    criarTabuleiro(nome);

    return 0;
}
```

A ”**executaCriaJogo**“ recebe os argumentos da linha de comandos, verifica se os mesmos estão certos e, se tal ocorrer, executa a ”**criarTabuleiro**“.

O parâmetro ”**args**“ recebe uma string com os argumentos associados ao comando em questão (”cr“). A função devolve um inteiro conforme o sucesso da operação.

Explicaremos agora a função **"jogada"**.

—

```
int jogada(char cor,int x,int y){  
  
    if((((x-1)<0) || ((x-1)>=DIM)) || (((y-1)<0) || ((y-1)>=DIM))) {  
        printf("\n jogou fora dos lmites do tabuleiro ou ainda não abriu o jogo \n\n");  
        return 0;  
    }  
  
    if(cor=='i' || cor=='b' || cor=='p'){  
        Tab[x-1][y-1].cor=cor;}  
    else {  
        printf(" Escolheu a cor errada \n");return 0;}  
    return 1;  
}
```

A função **"jogada"** é responsável por alterar os dados do tabuleiro (pode-se considerar que é ela que dá sentido ao "jogo", até). Esta recebe a cor pretendida e as coordenadas onde a mesma vai ser "pintada".

O parâmetro "x" corresponde á linha onde a cor vai ser substituída; o "y" á coluna; e,por fim, o parâmetro "cor" recebe a cor com que vamos pintar a posição na linha x e na coluna y.

A função "gravaJogo" é a última que achamos necessária ser explicada.

```

void gravaJogo (char *nome) {

    int i,j;

    FILE *f = fopen(nome, "w+");

    if(f == NULL){
        printf(stderr, "Erro ao criar o ficheiro '%s'\n", nome);
        exit(0);
    }
    else {
        fprintf( f,"%d\n",DIM);
        for(i=0;i<DIM;i++){
            for(j=0;j<DIM;j++){
                switch (Tab[i][j].cor) {
                    case ('i'):
                        fprintf(f, " ");
                        break;
                    case ('b'):
                        fprintf(f, "(");
                        break;
                    case ('p'):
                        fprintf(f, "[");
                        break;
                }
                fprintf(f,"%c",Tab[i][j].letra);
                switch (Tab[i][j].cor) {
                    case ('i'):
                        fprintf(f, " ");
                        break;
                    case ('b'):
                        fprintf(f, ")");
                        break;
                    case ('p'):
                        fprintf(f, "]");
                        break;
                }
            }
            fprintf(f, "\n");
        }
        fclose (f);
    }
}

```

A função "gravaJogo" é responsável por gravar o estado actual do tabuleiro com o nome que é dado como argumento, e no formato ".ltr".

O tabuleiro será imprimido para o ficheiro seguindo as regras de representação deste, para que possa ser lido de novo e seja considerado válido.

Exibiremos aqui duas imagens que mostram um pouco do funcionamento do interface.

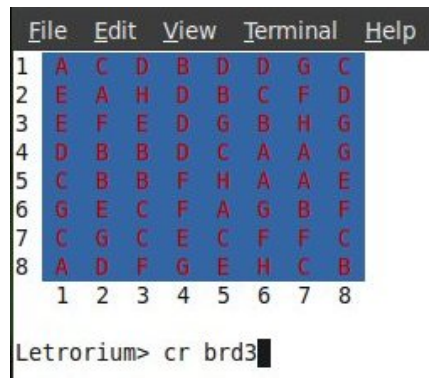


Imagem1

Introduzindo no interface o comando “cr brd3”, foi carregado o respectivo tabuleiro, ainda “vazio”, ou seja, com todas as posições sem cor definida.

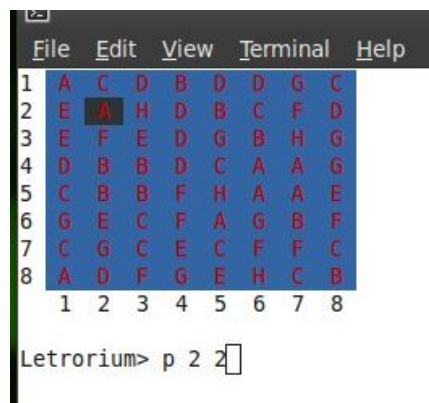


Imagem2

Introduzindo no interface o comando “p 2 2”, este “pintou” de cor preta a posição dada pela linha 2 e pela coluna 2.

Utilizar cores no interface era uma tarefa opcional, no entanto, fizemo-lo por acharmos que isso torna o aspecto do trabalho mais agradável.

2.3 Ferramentas utilizadas

Para o desenvolvimento do interface e do seu relatório, foram utilizadas as seguintes ferramentas:

Kile - Editor gráfico para LATEX;

NetBeans - IDE de código aberto para desenvolvedores de software na linguagem Java, C/C++, PHP, e muito mais;

gedit - editor oficial de texto plano para o Gnome.

De referir que todos os elementos do grupo utilizaram o Sistema Operativo **Ubuntu**.

Capítulo 3

Conclusão

Terminada a etapa, o grupo considera que o trabalho realizado foi positivo. Como já dissemos, os nossos conhecimentos sobre a linguagem de programação em questão pareciam suficientes para resolver o problema proposto. Assim, podemos considerar que esta etapa serviu mais para recordar os mecanismos básicos acerca da linguagem que poderiam estar esquecidos, mecanismos esses que nos serão certamente muito úteis nas próximas etapas do projecto da disciplina.

Pensamos que todos os objectivos propostos foram alcançados, visto que, além de o interface conter todas as exigências pretendidas, foram-lhe acrescentadas mais algumas características, como as cores do tabuleiro.

Capítulo 4

Bibliografia e referências WWW

Damas, Luís - “Linguagem C” FCA - Editora Informática, 1999

Simões, Alberto - “Uma não tão pequena introdução ao LATEX” Versão 4.14, 10/04/2004

Ritchie, Dennis e Kernighan, Brian - “The C Programming Language” 1978

www.wikipedia.org