Manual de Programador

Football Life



Tomás Silva

Índice

Bibliotecas e Headers 1

Titulo.h 2

Main.c 3

Login.H 4

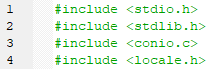
Menu\_Login.h 4

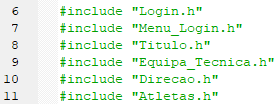
Direção.h 5

Equipa Técnica 6

Atletas 7

## Bibliotecas e Headers

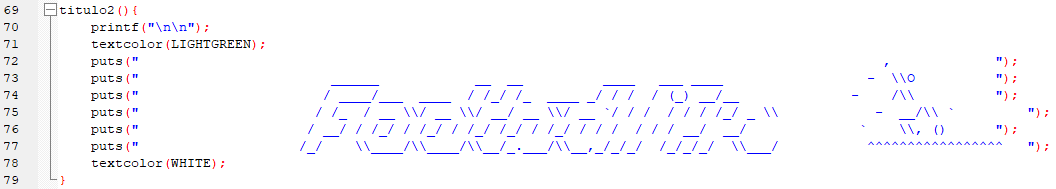
 Adicionei 4 bibliotecas ao programa, onde o <conio.c> serve para os textcolor(); e os Sleep(); e o <locale.h> para que o texto seja apresentado em português, com acentos adequados.

Adicionei também headers correspondeste ás funções que eu criei, cada uma delas têm o nome da sua função no programa.

 E ainda alguns defines para corresponder   
um número a uma certa variável, cada uma delas correspondente ao código de acesso dos nomes correspondestes.

## Titulo.h

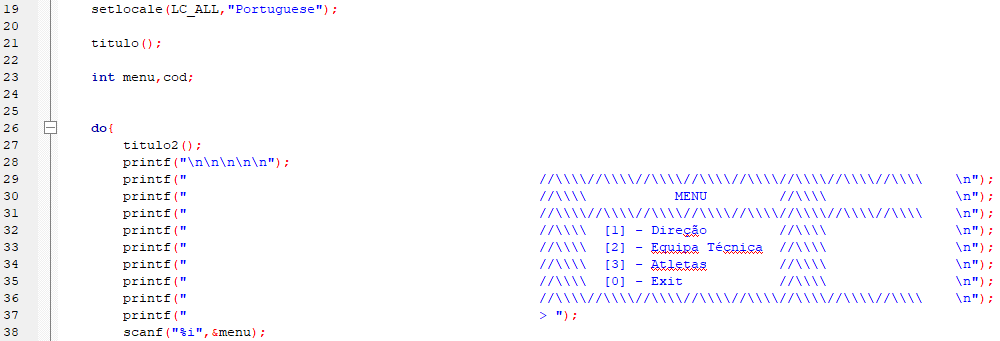
No header “Titulo.h” temos o que corresponde ao título, ou seja, existem duas funções onde a “titulo();” é usada para o início do programa apareça as várias cores na palavra FOOTBALL LIFE em art ascii, usando vários textcolor, e a função “titulo2();” serve para que em todas as páginas seja apresentado o titulo/logo (FOOTBALL LIFE em art ascii) em verde.



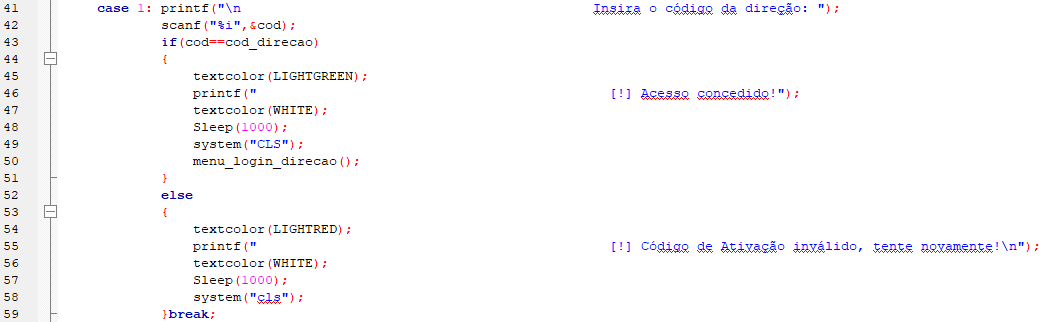
**Nota:** Este título não está muito percetível por que programação C tem de ser usar duas barras em vez de uma quando esta aparece: (\ -> \\).

## Main.c

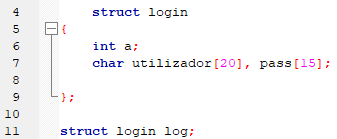
O “main.c” e a principal página do programa e, é lá, que está o menu inicial, já com os códigos de acesso.



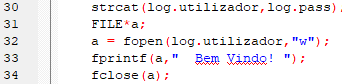
Começo por usar o “setlocale” para o texto aparecer em português e assim ter uso da biblioteca “locale.h”; declaro as variáveis inteiras do menu e do código de acesso e então segui para a construção do menu com o uso de um “do{…switch(){}…}while(menu!=0);”

De seguida, quando o utilizador digitar o número pretendido este terá de meter o código de acesso. Basicamente, o programa lê o número digitado pelo utilizador e compara-o com o “#define” correspondente, isto usando um” if() else()”. Se o utilizador acertar ele será direcionado para o “menu\_login\_(correspondente)();” que veremos mais a frente.

## Login.H



Para criar o login, no header “Login.h”, utilizei uma struct para as variáveis do utilizador e da password do mesmo.

Neste header existem duas funções: “registar();” e “login\_(correspondente)();”. A função “registar();”, tal como vemos na imagem, junta todos os caracteres introduzidos pelo utilizador com uma string e então cria um file com esse mesmo nome (utilizador e password), já na função “login();” também junta os caracteres introduzidos pelo utilizador e vê se há algum ficheiro txt com esse mesmo nome, se sim vai para o função correspondeste, seja esta a função para a direção, para a equipa técnica ou para os atletas.

## Menu\_Login.h

No header “Menu\_Login.h” simplesmente existem 3 funções, uma para cada função (direção, equipa técnica ou atletas), com um menu e dependente da sua escolha este poderá fazer o registo ou logar na sua conta já criada.



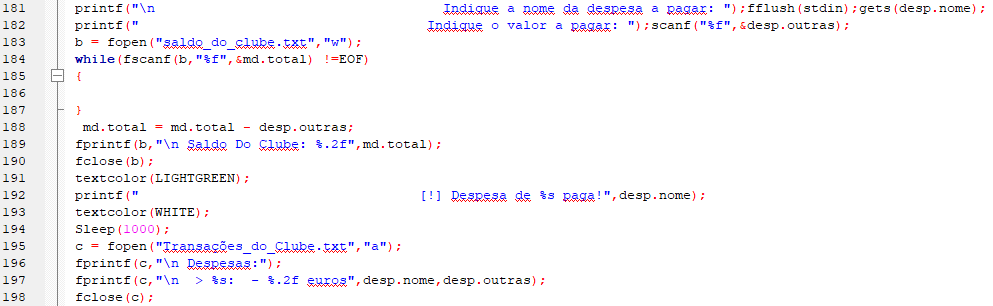
Imagem da função “menu\_login\_direcao();”.

## Direção.h

Logo depois de introduzir 4 structs, criei uma página com algumas opções, sendo algumas delas verificar o saldo do clube e as suas transações, pagar e receber quantias de dinheiro, que estão exemplificadas já a seguir.  
  
  
Ver Saldo do Clube

Quando o user escolhe a opção para ver o saldo do clube vai deparar-se com um outro menu em que poderá consultar o saldo do clube abrindo o “saldo\_do\_clube.txt” e também poderá consultar as suas ações no ficheiro “Transações\_do\_Clube.txt”. Estes ficheiros são desenvolvidos pelas outras duas funções.

#### Pagar Despesas

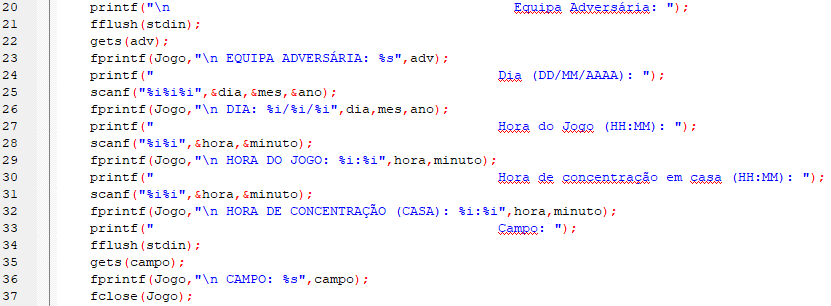
Para o membro da direção conseguir pagar as despesas, este terá um menu com as mais regulares despesas de clubes, e ainda uma outra opção para quaisquer outras despesas, como a luz, a água…  
Para isso, abri o “Saldo\_do\_Clube.txt” e li o saldo existente no clube, ou seja, o número escrito no txt e então foi só subtrair esse mesmo valor com o valor que o user tinha digitado, volto a escrever o novo saldo no txt e também abro outro (“Transações\_do\_Clube.txt”) para escrever a despesa paga e assim aparecer o que o membro da direção pagou neste mesmo txt e, assim, ele pode consultá-lo na “verificação do saldo do clube”.

#### Receber Pagamentos

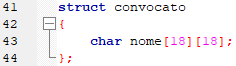
Esta opção não varia muito que a anterior, apenas o membro da direção consegue adicionar dinheiro ao clube, seja ele por cotas dos jogadores ou de qualquer outro meio. Assim, apenas mudei a subtração para uma soma e mais uns pequenos ajustes, tal como no ficheiro das transações.

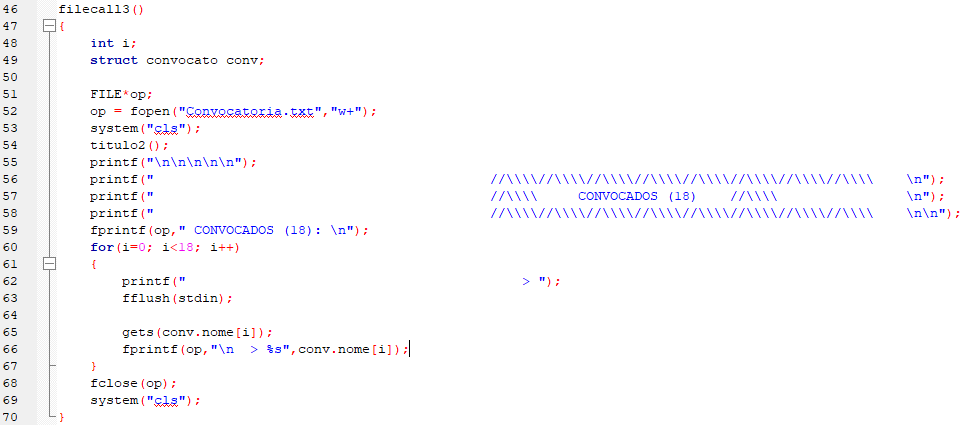
## Equipa Técnica

Normalmente, os treinadores e os atletas precisam de um meio de comunicação para saberem se são convocados e a que horas e a concentração antes do jogo e, para isso, utilizar o mesmo programa que a direção era a melhor opção.  
Para não desviar da estrutura do programa, continuei com um menu com algumas opções, sendo algumas delas inserir informações sobre o próximo jogo e fazer a convocatória, que estão exemplificadas já a seguir.  
  
Inserir informações sobre o próximo jogo

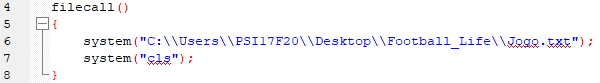
Consegui que o user da equipa técnica inserisse algumas informações para o próximo jogo e para fazê-lo bastou pedir a este que digitasse o pretendido e, a medida que este ia escrevendo, digitava a mesma coisa no “Jogo.txt” para que os atletas pudessem visualizá-lo sempre que quisessem.

#### Fazer a Convocatória

Consegui que o user da equipa técnica inserisse o nome dos 18 jogadores que iam jogar no próximo jogo e para fazê-lo bastou usar uma struct com vetores. Tal como na opção anterior, os nomes aparecem na “Convocatória.txt” para que os atletas pudessem visualizá-la sempre que quisessem.

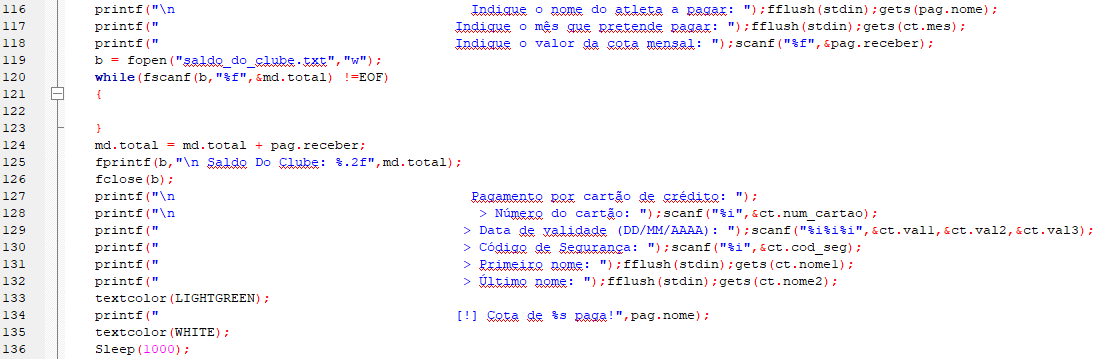


## Atletas

A parte mais simples é a dos atletas, pois apenas tem a abertura de dois filles. Apesar de ser muitos simples tem de se mudar de computador para computador, pois diferentes computadores têm diferentes users.

#### Pagar Cota

No case 3, o atleta consegue pagar a cota mensal e, para isso, pedi várias informações como as informações do cartão de crédito e então foi só utilizar o código da direção quando recebem pagamentos, ou seja, este lê o saldo do clube no ficheiro “saldo\_do\_clube.txt”, adiciona o montante da cota e volta a escrever o novo saldo no txt, como também escreve a transação no ficheiro das transações.



Neste mesmo caso existe a opção de o utilizador ver a fatura ou não e então decidi criar a função “fatura\_peg();” onde comparo a letra introduzida pelo utilizador com uma variável definida, com ajuda de uma string (“strcmpi”). A seguir, simplesmente crio uma fatura no txt com as informações do cartão de credito e abro-a com um “system();”.

