

Relatório 1º Estudo Dirigido - Teoria dos Grafos

Alunos: Iuker de Souza Santos, Gabriel Bauer

Define o número máximo de livros (MAX_LIVROS) que podem ser armazenados na biblioteca, e o tamanho máximo do nome, autor e categoria de cada livro (MAX_NOME).

```
4  #define MAX_LIVROS 100
5  #define MAX_NOME 100
6
```

Armazenar as informações de cada livro, contendo três strings:

Nome, autor e categoria (gênero).

```
7  typedef struct {
8      char nome[MAX_NOME];
9      char autor[MAX_NOME];
10     char categoria[MAX_NOME];
11 } Livro;
12
```

Declara as variáveis Globais e cria um vetor biblioteca para armazenar até 100 livros. Cada posição desse vetor é do tipo (Livro). A variável contador_livros mantém o número atual de livros inseridos.

```
12
13  Livro biblioteca[MAX_LIVROS];
14  int contador_livros = 0;
15
```

Esta função insere um novo livro na biblioteca.

Verifica se ainda há espaço disponível e se houver, usa strcpy para copiar os parâmetros para o próximo índice disponível na biblioteca, incrementa o contador e imprime uma mensagem de sucesso. E se não houver espaço informa que a biblioteca está cheia.

```
15
16  void inserir_livro(char nome[], char autor[], char categoria[]) {
17      if (contador_livros < MAX_LIVROS) {
18          strcpy(biblioteca[contador_livros].nome, nome);
19          strcpy(biblioteca[contador_livros].autor, autor);
20          strcpy(biblioteca[contador_livros].categoria, categoria);
21          contador_livros++;
22          printf("\n");
23          printf("O livro '%s' foi inserido com sucesso!\n", nome);
24          printf("\n");
25      } else {
26          printf("A biblioteca está cheia! Não é possível inserir mais livros.\n");
27      }
28  }
29
```

Faz uma busca por um livro específico comparando o nome fornecido com o nome de cada livro da biblioteca. Se encontrar, imprime as informações do livro. Se não encontrar, imprime uma mensagem informando que o livro não foi encontrado.

```
29
30 void buscar_livro_por_nome(char nome[]) {
31     int encontrado = 0;
32     for (int i = 0; i < contador_livros; i++) {
33         if (strcmp(biblioteca[i].nome, nome) == 0) {
34             printf("\n");
35             printf("Livro encontrado: %s, Autor: %s, Categoria: %s\n", biblioteca[i].nome, biblioteca[i].autor, biblioteca[i].categoria);
36             printf("\n");
37             encontrado = 1;
38         }
39     }
40 }
41 if (!encontrado) {
42     printf("\n");
43     printf("Nenhum livro encontrado com o nome '%s'.\n", nome);
44     printf("\n");
45 }
46 }
```

Faz uma busca por um livro específico comparando o autor fornecido com o autor de cada livro da biblioteca. Se encontrar, imprime as informações do livro. Se não encontrar, imprime uma mensagem informando que o livro não foi encontrado. Igual a função de anterior, porém por (autor).

```
48 void buscar_livro_por_autor(char autor[]) {
49     int encontrado = 0;
50     for (int i = 0; i < contador_livros; i++) {
51         if (strcmp(biblioteca[i].autor, autor) == 0) {
52             printf("\n");
53             printf("Livro encontrado: %s, Autor: %s, Categoria: %s\n", biblioteca[i].nome, biblioteca[i].autor, biblioteca[i].categoria);
54             printf("\n");
55             encontrado = 1;
56         }
57     }
58     if (!encontrado) {
59         printf("\n");
60         printf("Nenhum livro encontrado com o autor '%s'.\n", autor);
61         printf("\n");
62     }
63 }
```

Remove um livro pelo nome. Se o livro for encontrado, ele é removido deslocando todos os livros subsequentes para a esquerda, para preencher o "buraco" que ficaria.

O contador de livros é acionado e se não encontrar o livro, exibe uma mensagem informando que o livro não foi encontrado.

```
65 void remover_livro(char nome[]) {
66     int encontrado = 0;
67     for (int i = 0; i < contador_livros; i++) {
68         if (strcmp(biblioteca[i].nome, nome) == 0) {
69             for (int j = i; j < contador_livros - 1; j++) {
70                 biblioteca[j] = biblioteca[j + 1];
71             }
72             contador_livros--;
73             printf("\n");
74             printf("O livro '%s' foi removido com sucesso!\n", nome);
75             printf("\n");
76             encontrado = 1;
77             break;
78         }
79     }
80     if (!encontrado) {
81         printf("\n");
82         printf("Nenhum livro encontrado com o nome '%s'.\n", nome);
83         printf("\n");
84     }
85 }
```

Faz o menu interativo. Aparece as opções:

Inserir um livro, Buscar livro por nome, buscar livro por autor, Remover livro, Sair do programa.

A função é “puxada” / “chamada” de acordo com o número que o usuário digitou.

O loop mantém o menu sendo exibido até o usuário escolher a opção de sair, (opção 5).

```
87 void menu() {
88     int opcao;
89     char nome[MAX_NOME];
90     char autor[MAX_NOME];
91     char categoria[MAX_NOME];
92
93     do {
94         printf("—————\n");
95         printf("— Biblioteca Imaginária Digital —\n");
96         printf("—————\n");
97         printf("\n");
98         printf("1. Inserir um livro\n");
99         printf("\n");
100        printf("2. Buscar livro por nome\n");
101        printf("\n");
102        printf("3. Buscar livro por autor\n");
103        printf("\n");
104        printf("4. Remover livro\n");
105        printf("\n");
106        printf("5. Sair\n");
107        printf("\n");
108        printf("Escolha uma opção: ");
109        printf("\n");
110        scanf("%d", &opcao);
111        getchar();
112
113        switch (opcao) {
114            case 1:
115                printf("\n");
116                printf("Digite o nome do livro: ");
117                printf("\n");
118                fgets(nome, MAX_NOME, stdin);
119                nome[strcspn(nome, "\n")] = '\0';
120                printf("\n");
121                printf("Digite o nome do autor: ");
122                printf("\n");
123                fgets(autor, MAX_NOME, stdin);
124                autor[strcspn(autor, "\n")] = '\0';
125                printf("\n");
126                printf("Digite a categoria do livro: ");
127                printf("\n");
128                fgets(categoria, MAX_NOME, stdin);
129                categoria[strcspn(categoria, "\n")] = '\0';
130                inserir_livro(nome, autor, categoria);
131                break;
132            case 2:
133                printf("\n");
134                printf("Digite o nome do livro para buscar: ");
135                printf("\n");
136                fgets(nome, MAX_NOME, stdin);
137                nome[strcspn(nome, "\n")] = '\0';
138                buscar_livro_por_nome(nome);
139                break;
```

```

139         break;
140     case 3:
141         printf("\n");
142         printf("Digite o nome do autor para buscar: ");
143         printf("\n");
144         fgets(autor, MAX_NOME, stdin);
145         autor[strcspn(autor, "\n")] = '\0';
146         buscar_livro_por_autor(autor);
147         break;
148     case 4:
149         printf("\n");
150         printf("Digite o nome do livro para remover: ");
151         printf("\n");
152         fgets(nome, MAX_NOME, stdin);
153         nome[strcspn(nome, "\n")] = '\0';
154         remover_livro(nome);
155         break;
156     case 5:
157         printf("\n");
158         printf("Saindo ... \n");
159         printf("\n");
160         break;
161     default:
162         printf("\n");
163         printf("Opção inválida. Tente novamente.\n");
164         printf("\n");
165         break;
166     }
167 } while (opcao != 5);
168 }

```

Chama a função menu para iniciar a biblioteca virtual.

```

170 int main() {
171     menu();
172     return 0;
173 }

```