



Programação Procedimental

GBC014 – 2015/1

Prof. Renan Cattelan – www.facom.ufu.br/~renan

Prática 11

Arquivos

Arquivos em C

- ❑ A linguagem C utiliza o conceito de streams e arquivos para manipular entradas e saídas de dados
- ❑ Um stream é um dispositivo lógico representando um arquivo e pode ser de dois tipos:
 - ❑ Stream de texto: sequência de caracteres dividida em linhas (terminadas em `\n`)
 - ❑ Stream binário: sequência de valores inteiros, reais, ou complexos usando suas representações de memória

Biblioteca stdio.h

- Contém funções de manipulação de arquivos

```
1 /* fopen example */
2 #include <stdio.h>
3 int main ()
4 {
5     FILE * pFile;
6     pFile = fopen ("myfile.txt","w");
7     if (pFile!=NULL)
8     {
9         fputs ("fopen example",pFile);
10        fclose (pFile);
11    }
12    return 0;
13 }
```

Biblioteca stdio.h

- Contém funções de manipulação de arquivos

- Objeto FILE
- Usado para identificar uma stream
- pFile é uma variável do tipo ponteiro para arquivo/stream

```
1 /* f
2 #inc
3 int main
4 {
5     FILE * pFile;
6     pFile = fopen ("myfile.txt","w");
7     if (pFile!=NULL)
8     {
9         fputs ("fopen example",pFile);
10        fclose (pFile);
11    }
12    return 0;
13 }
```

□ Con arqu

- Função que cria/abre um arquivo para manipulação
- 2 parâmetros:
 - nome do arquivo (filename)
 - modo de acesso (mode)
 - “r”: leitura (read)
 - “w”: escrita (write)
 - “a”: anexo (append)
- Retorna um ponteiro para o arquivo

```
4 {  
5     FILE * pFile;  
6     pFile = fopen ("myfile.txt", "w");  
7     if (pFile != NULL)  
8     {  
9         fputs ("fopen example", pFile);  
10        fclose (pFile);  
11    }  
12    return 0;  
13 }
```

Biblioteca stdio.h

- Contém funções de manipulação de arquivos

```
1 /* fopen example */
2 #include <stdio.h>
3 int main ()
4 {
5     FILE * pFile;
6     pFile = fopen ("
7     if (pFile!=NULL)
8     {
9         fputs ("fopen example",pFile);
10        fclose (pFile);
11    }
12    return 0;
13 }
```

- Escreve uma string no arquivo
- 2 parâmetros:
 - A string
 - O ponteiro de arquivo
- Outras funções de escrita:
 - fprintf(), fputc(), ...

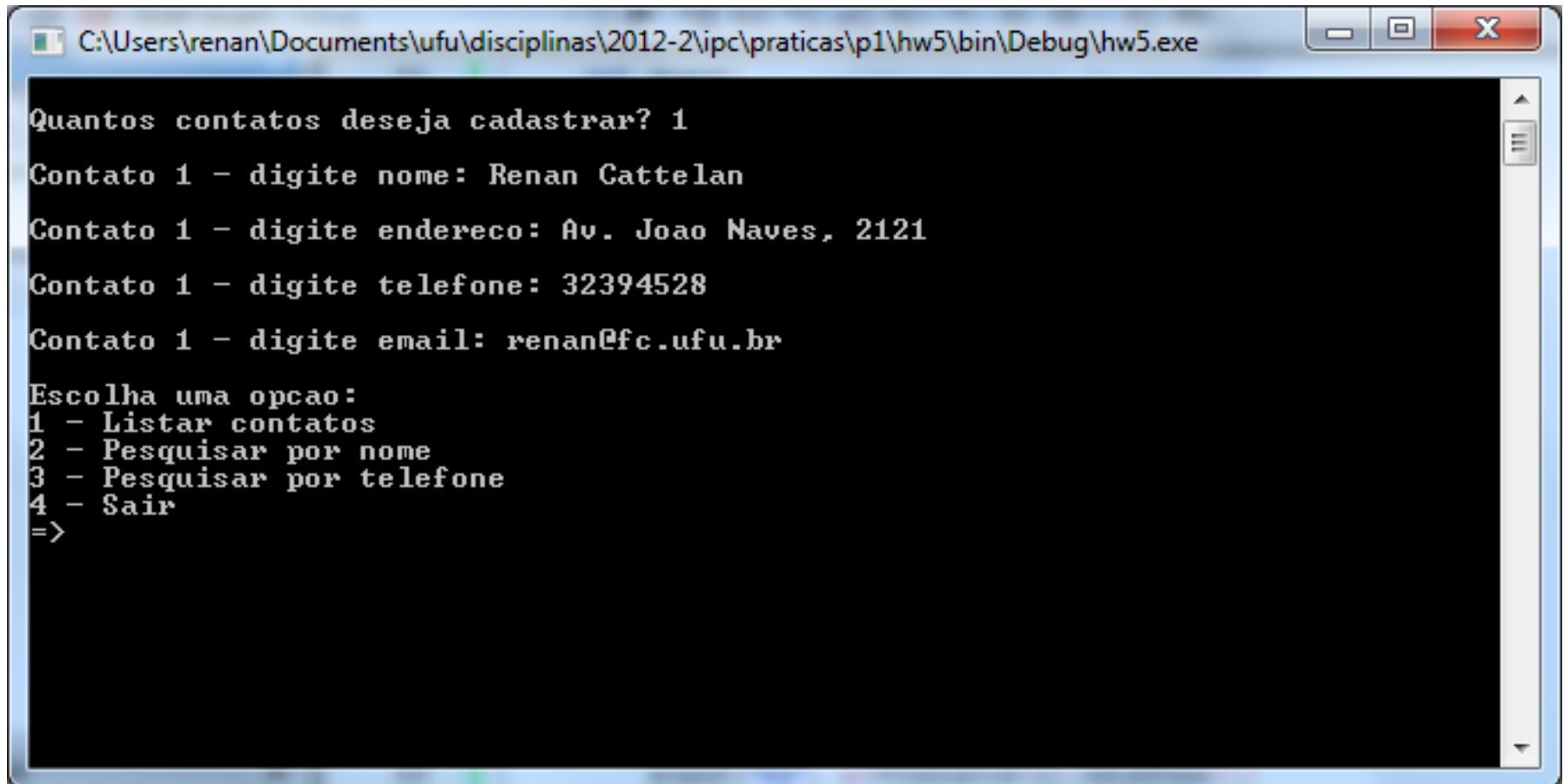
Biblioteca stdio.h

- Contém funções de manipulação de arquivos

```
1 /* fopen example */
2 #include <stdio.h>
3 int main ()
4 {
5     FILE * pFile;
6     pFile = fopen ("myfile.txt","w");
7     if (pFile!=NULL)
8     {
9         fputs ("fopen example",pFile);
10        fclose (pFile);
11    }
12    return 0;
13 }
```

•Fecha o arquivo

Prática 11: agenda de contatos



```
C:\Users\renan\Documents\ufu\disciplinas\2012-2\ipc\praticas\p1\hw5\bin\Debug\hw5.exe

Quantos contatos deseja cadastrar? 1
Contato 1 - digite nome: Renan Cattelan
Contato 1 - digite endereco: Av. Joao Naves, 2121
Contato 1 - digite telefone: 32394528
Contato 1 - digite email: renan@fc.ufu.br

Escolha uma opcao:
1 - Listar contatos
2 - Pesquisar por nome
3 - Pesquisar por telefone
4 - Sair
=>
```


Exercício

- ❑ Complemente o exercício feito na Prática 11 de modo a criar novas funções para:
 - ❑ Carregar a lista de contatos a partir de um arquivo cujo nome é fornecido pelo usuário (a função deve retornar o número de contatos carregados – dica: crie uma função auxiliar para contar as linhas do arquivo)
 - ❑ Adicionar um novo contato
 - ❑ Remover um contato existente (a partir do nome do contato – a função deve retornar o número de contatos que sobraram no vetor)
 - ❑ Permitir salvar a lista de contatos em um arquivo informado pelo usuário (um campo de contato por linha do arquivo)
- ❑ Reimplemente o vetor de contatos de modo estático, com tamanho máximo de 50 contatos.

Exercício

- ❑ Complemente o exercício feito na Prática 11 de modo a criar novas funções para:
 - ❑ Carregar a lista de contatos a partir de um arquivo cujo nome é fornecido pelo usuário (a função deve retornar o número de contatos carregados – dica: crie uma função auxiliar para contar as linhas do arquivo)
 - ❑ Adicionar um novo contato
 - ❑ Remover um contato existente (a partir do nome do contato – a função deve retornar o número de contatos que sobraram no vetor)
 - ❑ Permitir salvar a lista de contatos em um arquivo informado pelo usuário (um campo de contato por linha do arquivo)
- ❑ Reimplemente o vetor de contatos de modo estático, com tamanho máximo de 50 contatos.

```
62  int carregar_do_arquivo(Contato *v, char *nomeArq) {
63      FILE *fp = fopen(nomeArq, "r");
64      if (fp == NULL) return 0;
65      int q = conta_linhas(nomeArq);
66      printf("\nArquivo tem %d linhas.", q);
67      int i;
68      for (i=0; i<(q-1)/4; i++) {
69          v[i].nome = (char *) malloc(sizeof(char)*50);
70          fscanf(fp, " %50[^\n] %*c", v[i].nome);
71
72          fscanf(fp, " %50[^\n] %*c", v[i].endereco);
73
74          fscanf(fp, "%d\n", &(v[i].telefone));
75
76          fscanf(fp, " %30[^\n] %*c", v[i].email);
77      }
78
79      fclose(fp);
80      return ((q-1)/4);
81 }
```

```

62 int carregar_do_arquivo (Contato *v, char *nomeArq) {
63     FILE *fp = fopen(nomeArq, "r");
64     if (fp == NULL) return 0;
65     int q = conta_linhas(nomeArq);
66     printf("\nArquivo tem %d linhas.", q);
67     int i;
68     for (i=0; i<(q-1)/4; i++) {
69         v[i].nome = (char *) malloc(sizeof(char)*50);
70         fscanf(fp, " %50[^\n]%*c", v[i].nome);
71
72         fscanf(fp, " %50[^\n]%*c", v[i].endereco);
73
74         fscanf(fp, "%d\n", &v[i].telefone);
75
76         fscanf(fp, "%d\n", &v[i].idade);
77     }
78
79     fclose(fp);
80     return ((q-1)/4);
81 }

```

- Lê do arquivo

- Parâmetros:

- fp: ponteiro do arquivo

- Formato (ex. "%s\n")

- Outros: variáveis a serem lidas, de acordo com o formato dado

```
47  int conta_linhas(char *filename) {
48      FILE *fp = fopen(filename, "rt");
49      int ch = 0, nLinhas = 0;
50
51      if (fp == NULL) return 0;
52
53      nLinhas++;
54      while ((ch = fgetc(fp)) != EOF) {
55          if (ch == '\n')
56              nLinhas++;
57      }
58      fclose(fp);
59      return nLinhas;
60 }
```

```

47  int conta_linhas(char *filename) {
48      FILE *fp = fopen(filename, "rt");
49      int ch = 0, nLinhas = 0;
50
51      if (fp == NULL) return 0;
52
53      nLinhas++;
54      while ((ch = fgetc(fp)) != EOF) {
55          if (ch == '\n')
56              nLinhas++;
57      }
58      fclose(fp);
59      return nLinhas;
60  }

```

- Lê (caracter) do arquivo
- Parâmetros:
 - fp: ponteiro do arquivo
- Retorna caracter lido (posição continuamente atualizada)

Exercício

- ❑ Complemente o exercício feito na Prática 11 de modo a criar novas funções para:
 - ❑ Carregar a lista de contatos a partir de um arquivo cujo nome é fornecido pelo usuário (a função deve retornar o número de contatos carregados – dica: crie uma função auxiliar para contar as linhas do arquivo)
 - ❑ **Adicionar um novo contato**
 - ❑ Remover um contato existente (a partir do nome do contato – a função deve retornar o número de contatos que sobraram no vetor)
 - ❑ Permitir salvar a lista de contatos em um arquivo informado pelo usuário (um campo de contato por linha do arquivo)
- ❑ **Reimplemente o vetor de contatos de modo estático, com tamanho máximo de 50 contatos.**

```
83 void cadastrar_novo_contato(Contato *v, int *q) {
84     if (*q >= 50)
85         printf("\nNao eh possivel cadastrar novo contato. Lista ja atingiu limite.\n");
86     else {
87         printf("\nNovo contato - digite nome: ");
88         // alocao dinamica do nome
89         v[*q].nome = (char *) malloc(sizeof(char)*50);
90         scanf(" %50[^\n]c", v[*q].nome);
91
92         printf("\nNovo contato - digite endereco: ");
93         scanf(" %50[^\n]c", v[*q].endereco);
94
95         printf("\nNovo contato - digite telefone: ");
96         scanf("%d", &(v[*q].telefone));
97
98         printf("\nNovo contato - digite email: ");
99         scanf(" %30[^\n]c", v[*q].email);
100
101         (*q)++;
102     }
103 }
```

```
138 int main(void) {
139     Contato vContatos[50]; // vetor de contatos
140     int quant = 0; // tamanho do vetor de contatos (inicialmente 0)
```


Exercício

- ❑ Complemente o exercício feito na Prática 11 de modo a criar novas funções para:
 - ❑ Carregar a lista de contatos a partir de um arquivo cujo nome é fornecido pelo usuário (a função deve retornar o número de contatos carregados – dica: crie uma função auxiliar para contar as linhas do arquivo)
 - ❑ Adicionar um novo contato
 - ❑ **Remover um contato existente (a partir do nome do contato – a função deve retornar o número de contatos que sobraram no vetor)**
 - ❑ Permitir salvar a lista de contatos em um arquivo informado pelo usuário (um campo de contato por linha do arquivo)
- ❑ Reimplemente o vetor de contatos de modo estático, com tamanho máximo de 50 contatos.

```
105 void remover_contato(Contato *v, int *q, char *nome) {
106     int i, achou = 0;
107     for (i=0; i<(*q); i++) {
108         if (strcmp(v[i].nome,nome) == 0 && achou == 0) {
109             achou = 1;
110         } else if (achou == 1 && i > 0) { // move todos que vem depois uma posicao antes
111             v[i-1].nome = v[i].nome;
112             strcpy(v[i-1].endereco, v[i].endereco);
113             v[i-1].telefone = v[i].telefone;
114             strcpy(v[i-1].email, v[i].email);
115         }
116     }
117     if (achou == 1) {
118         printf("\nContato removido com sucesso!\n");
119         (*q)--;
120     } else {
121         printf("\nContato nao encontrado!\n");
122     }
123 }
```

Exercício

- ❑ Complemente o exercício feito na Prática 11 de modo a criar novas funções para:
 - ❑ Carregar a lista de contatos a partir de um arquivo cujo nome é fornecido pelo usuário (a função deve retornar o número de contatos carregados – dica: crie uma função auxiliar para contar as linhas do arquivo)
 - ❑ Adicionar um novo contato
 - ❑ Remover um contato existente (a partir do nome do contato – a função deve retornar o número de contatos que sobraram no vetor)
 - ❑ Permitir salvar a lista de contatos em um arquivo informado pelo usuário (um campo de contato por linha do arquivo)
- ❑ Reimplemente o vetor de contatos de modo estático, com tamanho máximo de 50 contatos.

```
125 void salvar_em_arquivo(Contato *v, int q, char *nomeArq) {
126     FILE *fp = fopen(nomeArq, "w");
127     if (fp == NULL) return;
128     int i;
129     for (i=0; i<q; i++) {
130         fprintf(fp, "%s\n", v[i].nome);
131         fprintf(fp, "%s\n", v[i].endereco);
132         fprintf(fp, "%d\n", v[i].telefone);
133         fprintf(fp, "%s\n", v[i].email);
134     }
135     fclose(fp);
136 }
```

```
125 void salvar_em_arquivo(Contato *v, int q, char *nomeArq) {
126     FILE *fp = fopen(nomeArq, "w");
127     if (fp == NULL) return;
128     int i;
129     for (i=0; i<q; i++) {
130         fprintf(fp, "%s\n", v[i].nome);
131         fprintf(fp, "%s\n", v[i].endereco);
132         fprintf(fp, "%d\n", v[i].telefone);
133         fprintf(fp, "%s\n", v[i].email);
134     }
135     fclose(fp);
136 }
```

- Escreve no arquivo

- Parâmetros:

- fp: ponteiro do arquivo

- Formato (ex. "%s\n")

- Outros: variáveis a serem escritas, de acordo com o formato dado

```
171     case 4:
172         printf("\nNome do arquivo: ");
173         scanf(" %50[^\n]%"c", nomeArq);
174         quant = carregar_do_arquivo(vContatos, nomeArq);
175         break;
176     case 5:
177         cadastrar_novo_contato(vContatos, &quant);
178         break;
179     case 6:
180         printf("\nDigite o nome desejado: ");
181         scanf(" %50[^\n]%"c", nome);
182         remover_contato(vContatos, &quant, nome);
183         break;
184     case 7:
185         printf("\nNome do arquivo: ");
186         scanf(" %50[^\n]%"c", nomeArq);
187         salvar_em_arquivo(vContatos, quant, nomeArq);
188         break;
189     case 8:
190         printf("\nTchau!\n");
191         break;
```

Referências

- ❑ <http://www.cplusplus.com/reference/cstdio/FILE/>
- ❑ <http://www.cplusplus.com/reference/cstdio/fopen/>
- ❑ <http://www.cplusplus.com/reference/cstdio/fputs/>
- ❑ <http://www.cplusplus.com/reference/cstdio/fgets/>
- ❑ <http://www.cplusplus.com/reference/cstdio/fprintf/>
- ❑ <http://www.cplusplus.com/reference/cstdio/fscanf/>
- ❑ <http://www.cplusplus.com/reference/cstdio/fputc/>
- ❑ <http://www.cplusplus.com/reference/cstdio/fgetc/>
- ❑ <http://www.cplusplus.com/reference/cstdio/fclose/>

- ❑ Em português: <http://pt.cppreference.com/w/c/io>