Laboratório de programação II

Manipulação de arquivos

Edson Moreno

edson.moreno@pucrs.br

http://www.inf.pucrs.br/~emoreno

Relembrar é viver

- Manipulação em C
  - Criação de Arquivos
    - Manipulação de arquivo requer a biblioteca STDIO.H
    - Para usar um arquivo em C é necessário abrí-lo. Para tanto, a linguagem C possui o comando ***fopen***.

      arq = fopen("ArqGrav.txt", "rt");

      - O primeiro parâmetro é o nome do arquivo, o segundo a forma de abertura:

        "wt": abertura para gravação, arquivo texto

        "rt": abertura para leitura, arquivo texto
    - A função fopen retorna um "apontador" para o arquivo caso consiga abrí-lo, caso contrário, retorna a constate NULL.

Relembrar é viver

- Manipulação em C
  - Código exemplo

```
FILE *arq;
int result;
char Str[50];

arq = fopen("ArqGrav.txt", "rt");

if (arq == NULL){
    printf("Problemas na CRIACAO do arquivo\n");
    return;
}
```

Relembrar é viver

- Manipulação de arquivos texto em C
  - Leitura
    - Para leitura de arquivos texto usar a função fgets ou fscanf.
  - Leitura com FGETS
    - A função FGETS lê uma linha inteira de uma vez.

```
// o 'fgets' lê até 99 caracteres ou até o '\n'
result = fgets(Linha, 100, arq);
```

    - Se a função for executada com sucesso então
      - fgets retorna o endereço da string lida
    - Senão retorna NULL.

Relembrar é viver

- Manipulação de arquivos em C (Leitura)

```
void main()
{
  FILE *arq;
  char Linha[100];
  char *result;
  int i;
  clrscr();
  // Abre um arquivo TEXTO para LEITURA
  arq = fopen("ArqTeste.txt", "rt");
  if (arq == NULL){  // Se houve erro na abertura
printf("Problemas na abertura do arquivo\n");
     return;
  }
  i = 1;
  while (!feof(arq)){
      // Lê uma linha (inclusive com o '\n')
      // o 'fgets' lê até 99 caracteres ou até o '\n'
      result = fgets(Linha, 100, arq);
      if (result) printf("Linha %d : %s",i,Linha);
      i++;
  }
  fclose(arq);
}
```

Relembrar é viver

- Manipulação de arquivos em C (Leitura)
  - Leitura com FSCANF
    - Funcionamento similar a função SCANF
      - Diferença está na origem do dado lido (arquivo x teclado)
    - Exemplo:

```
int i, result;
float x;
result = fscanf(arq, "%d%f", &i, &x);
```

    - Se result for igual à constante EOF, não há mais dados para serem lidos.

Relembrar é viver

- Manipulação de arquivos em C (escrita)
  - Gravação
    - Usa-se as funções fputs e fprintf
  - Exemplo de fputs:

    result = fputs(Str, arq);

  - Se a função NÃO for executada com sucesso então
    - fputs retorna a constante EOF.
  - Senão retorna um valor não negativo

Relembrar é viver

- Manipulação em C (escrita)

```
char Str[100];
FILE *arq;

// Cria um arquivo texto para gravação
arq = fopen("ArqGrav.txt", "wt");

if (arq == NULL){// Se não conseguiu criar
   printf("Problemas na CRIACAO do arquivo\n");
   return;
}

strcpy(Str,"Linha de teste");
result = fputs(Str, arq);

if (result == EOF) printf("Erro na Gravacao\n");

fclose(arq);
```

Relembra é viver

- Manipulação em C (escrita)
  - Exemplo de fprintf:
    - result = fprintf(arq,"Linha %d\n",i);
  - Se a função fprintf for executada com sucesso então
    - Devolve o número de caracteres gravados.
  - Senão retorna a constante EOF.

Relembrar é viver

- Manipulação em C (escrita)

```
#include <stdio.h>
#include <conio.h>
void main()
{
  FILE *arq;
  int i;
  int result;
  clrscr();
  arq = fopen("ArqGrav.txt", "wt");  // Cria um arquivo texto para gravação
  if (arq == NULL){ // Se nào conseguiu criar
     printf("Problemas na CRIACAO do arquivo\n");
     return;
  }
  for (i = 0; i<10;i++) {
    // A funcao 'fprintf' devolve o número de bytes gravados
    // ou EOF se houve erro na gravação
    result = fprintf(arq,"Linha %d\n",i);
    if (result == EOF) printf("Erro na Gravacao\n");
  }
  fclose(arq);
}
```

Manipulando streams com C++

- Diversos recursos arquivos (*streams)*.
  - S*tream* representa um fluxo de entrada ou de saída
    - Exemplo
      - *cout* é um fluxo de saída
      - *cin* é um fluxo de entrada
  - Pode-se usar fluxos de saida para *strings*
    - Permitem compor *strings* mais facilmente

Manipulando streams com C++

- String stream
  - Permite explorar funcionalidades da *iostream*
    - Formatos
    - Alinhamento
    - precisão numérica
    - tamanho de campo
    - etc

Manipulando streams com C++

- String stream

```
#include <iostream>
#include <iomanip>
#include <cmath>
#include <sstream>

using namespace std;

string converte(float valor)
{
        ostringstream aux;  // Declara o string stream de saída chamado aux
        aux << "Exemplo de saída em string ";  // Escreve em aux
        aux << fixed << setprecision(2) << sqrt(valor); // idem
        return aux.str(); // Retorna a string resultante a partir do stream aux
}

int main()
{
        float v;
        cout << "Digite o valor: ";
        cin >> v;
        cout << converte(v);
}
```

Manipulando streams com C++

- Manipulação de arquivos – Stream de dados
  - Trata-se streams como arquivos físicos
  - Explora duas ações
    - *Gravação de dados ou criação de arquivo*
    - *Leitura de dados*

Manipulando streams com C++

- Gravando em um stream
  - Bibliotecas básicas/úteis

```
#include <fstream>      // para usar file streams (ifstream,ofstream)
#include <iostream>     // para usar cin,cout
#include <string>       // para usar string
#include <iomanip>      // para usar manipuladores (setw, right, ...)
#include <cstdlib>      // para usar srand(), rand() e exit()

using namespace std;
```

Manipulando streams com C++

- Escrita de arquivo

```
int main() { // Cria output file stream (ofstream)
  ofstream arqsaida;
  arqsaida.open( "teste.txt" , ios::out ); // Cria e abre arquivo

  if (!arqsaida.is_open()) return 0; // Se houver erro, sai do programa

  srand(time(0)); // Gera a semente aleatória

  cout << "Gerando dados..." << endl;

  arqsaida << "Cabecalho do arquivo" << endl; // Grava o cabeçalho
```

Manipulando streams com C++

- Escrita de arquivo

```
for (int i = 0; i < 10000; i++) { // grava os 10000 registros numéricos
    int num = rand() % 10000;
    arqsaida << i << setw(10) << num << endl;
    if(arqsaida.fail()) {
      cout << "Erro fatal!" << endl;
      exit(1); // Aborta programa
    }
  }
  cout << "Fechando o arquivo..." << endl;
  arqsaida.close();
  return 0;
}
```

Manipulando streams com C++

- Leitura de arquivo

```
int main() {
        // Cria input file stream (ifstream)
        ifstream arq;

        cout << "Abrindo arquivo texto..." << endl;

        // Abre arquivo
        arq.open( "teste.txt" , ios::in );

        // Se houver erro, sai do programa
        if (!arq.is_open())
                return (0);
        // Lê cabeçalho
        string cabecalho;
        getline(arq,cabecalho);
```

Manipulando streams com C++

- Leitura de arquivos

```
        // Exibe cabeçalho na tela
        cout << cabecalho << endl;
        // Agora, lê n registros numéricos
        do
        {
                int num, valor;
                arq >> num >> valor;
                if(!arq.fail()) {
                        cout << num << "\t" << valor << endl;
                }
        } while(arq.good());
        if(arq.bad() || !arq.eof()) {
                cout << "Erro fatal!" << endl;
                exit(1);          // Aborta programa
        }
        cout << "Fechando o arquivo..." << endl;
        arq.close();

        return 0;
}
```

Exercício

- Implemente um sistema para contabilização de votos:
  - Ler os dados dos candidatos do arquivo candidatos.txt
  - Ler os dados da votação de cada urna (arquivos urna[1-4].txt)
    - acumular os votos de cada candidato;
  - Exibir na tela
    - o relatório da votação:
    - Candidato mais votado;
    - Candidato menos votado;
    - Percentual de votos do candidato mais votado em relação ao total;
  - Baixar os arquivos que contem os dados

Exercício

- Detalhes dos arquivos
  - O formato dos arquivos (por linha) é o seguinte:
    - Arquivo candidatos.txt: número_do_candidato nome_do_candidato nome_partido. Ex:
      - 1 Roubalino PDR
      - 2 Estelionatino PDR
      - 3 Robaldo PD
    - Arquivo urna[1-4].txt: número do candidato para este voto.
    - Exemplo
      - 5
      - 10
      - 6
    - Leia números e strings com >>.