Gerenciamento de Arquivos

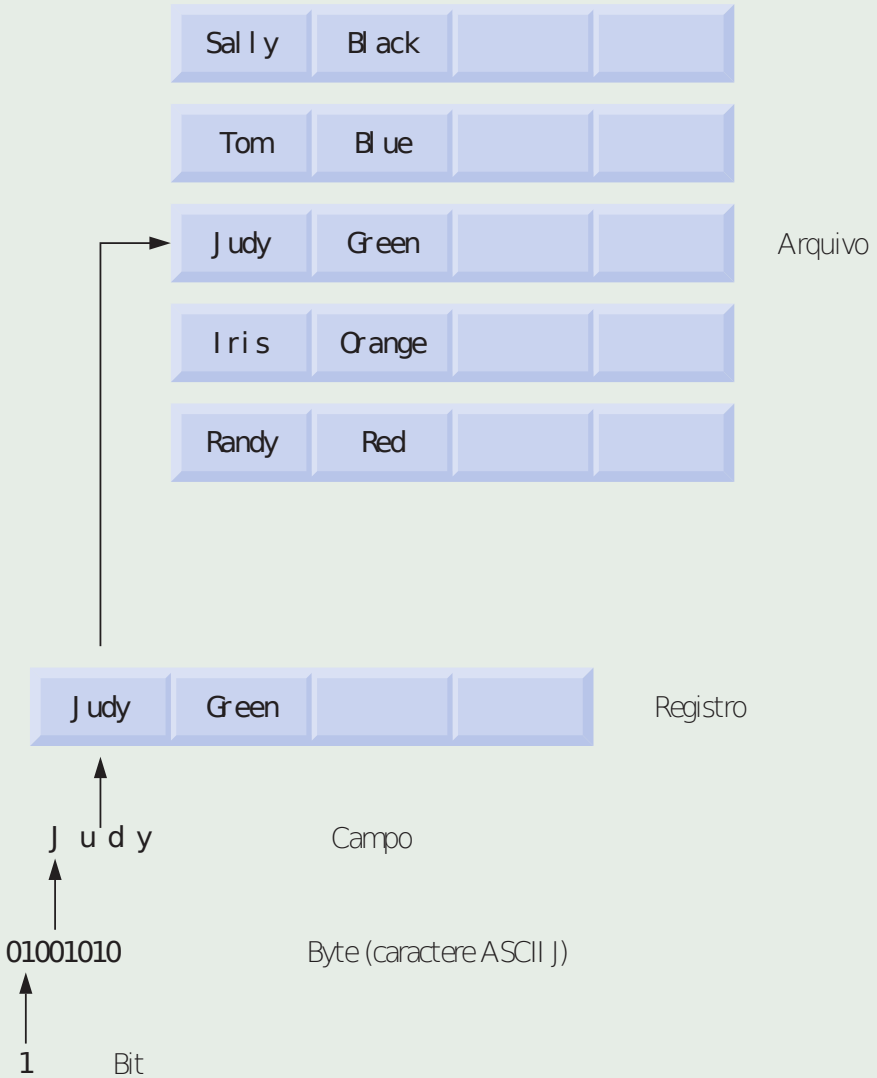
Prof. Valdir Pedrinho de Tomin Junior, Eng.

Universidade Federal de Santa Catarina
Centro Tecnológico de Joinville
EMB5631 – Programação III

e-mail: valdir.pedrinho@ufsc.br

6 de novembro de 2018

Hierarquia de dados



Introdução

Arquivos e fluxos

- Em C++ os arquivos não são estruturados;
  - O programador deve saber como o arquivo está organizado. Você deve organizar o arquivo, sabendo onde estão os dados, como ler/escrever.
- Cada arquivo é visto como uma sequência de bytes;
- Todo arquivo termina com um marcador de fim;
- Fluxo para uso de arquivos
  - Abertura $\rightarrow$ manipulação $\rightarrow$ fechamento
- Para realizar processamento de arquivo no C++, os arquivos de cabeçalho <**iostream**> e <**fstream**> devem ser incluídos. Templates de classe de fluxo:
  - **basic_ifstream**;
  - **basic_ofstream**;
  - **basic_fstream**.

Introdução

Manipulação de arquivos

- *ios::in*
  - Abre arquivo para entrada;
- *ios::out*
  - Abre o arquivo para saída (elimina conteúdo pré-existente)
- *ios::trunc*
  - Trunca o arquivo (elimina conteúdo pré-existente)
- *ios::app*
  - Grava no fim do arquivo (continua de onde parou);
- *ios::ate*
  - Abre um arquivo para saída e move-se para o fim do arquivo (normalmente utilizado para acrescentar dados a um arquivo);
- *ios::binary*
  - Abre um arquivo para entrada ou saída binária (isto é, não-texto).

Manipulação de arquivos

- Manipulação do cursor do arquivo
  - Ponteiro para próxima posição de leitura: arquivo.seekg(posicao)
  - Ponteiro para próxima posição de escrita: arquivo.seekp(posicao)
- Posição do cursor do arquivo
  - Ponteiro para próxima posição de leitura: arquivo.tellg(posicao)
  - Ponteiro para próxima posição de escrita: arquivo.tellp(posicao)

Arquivos Sequenciais

Formato de Arquivo

- Os registros são normalmente armazenados em ordem por um campo de chave de registro;
- Dados em formato legível (visível em editor).

Manipulação

Arquivos texto podem ser manipulados:

- Somente para leitura;
- Somente para escrita;
- De forma combinada: leitura e escrita
  - Na verdade usa-se um arquivo temporário

Arquivos Sequenciais

Arquivos de Entrada

Input File Stream:
ifstream nome("nomearquivo.extensão")

Arquivos de Saída

Output File Stream:
ofstream nome("nomearquivo.extensão")

Exemplos

17.4, 17.7, 17.8 do *DEITEL. C++, Como programar. 5ª ed.*

Arquivos Sequenciais

Riscos relacionados às atualizações de arquivos sequencias

- Os dados formatados e gravados em um arquivo sequencial não podem ser modificados sem o risco de destruir outros dados no arquivo;
- Um registro com mais caracteres que o registro original sobrescreveria o registro na sequência;
- O problema é que, no modelo de entrada/saída formatada utilizando o operador de inserção de fluxo « e o operador de extração de fluxo » , os campos – e, portanto, os registros – podem variar de tamanho;
- Por exemplo, os valores $7, 14, -117, 2.074$ e 27.383 são todos **int**, que armazenam o mesmo número de bytes. Entretanto, esses inteiros tornam-se campos de diferentes tamanhos quando sua saída é gerada como texto formatado.

Arquivos de Acesso Aleatório

Atualizando arquivos sequenciais

- Copiar dados para um novo arquivo;
- Requer processar cada registro no arquivo para atualizar um registro.

Características

- Os arquivos sequenciais são inadequados aos aplicativos de acesso instantâneo, em que um registro particular deve ser imediatamente localizado;
- Os dados podem ser inseridos em um arquivo de acesso aleatório sem destruir outros dados no arquivo;
- Os dados previamente armazenados também podem ser atualizados ou excluídos sem regravar o arquivo inteiro;
- O método mais fácil seja impor que todos os registros em um arquivo tenham o mesmo comprimento fixo.

Arquivos de Acesso Aleatório

Criando um arquivo de acesso aleatório

- Quando o fluxo é associado com um arquivo, a função **write** grava os dados na localização do arquivo especificada pelo ponteiro de posição de arquivo **put**;
- A **istream read** transfere um número fixo de bytes a partir do fluxo especificado para uma área do início da memória em um endereço especificado;
- Se o fluxo estiver associado com um arquivo, a função **read** insere os bytes na localização do arquivo especificada pelo ponteiro de posição de arquivo **get**

Usando ostream write

```
outFile.write( reinterpret_cast <const char * >
 (&number), sizeof (number) );
```

Arquivos de Acesso Aleatório

Exemplos do *DEITEL. C++, Como programar. 5ª ed.*

- Criando um arquivo de acesso aleatório:
  17.10-12
- Gravando em um arquivo de acesso aleatório:
  17.13
- Lendo um arquivo de acesso aleatório sequencialmente:
  17.14
- Estudo de caso: um programa de processamento de transação:
  17.15

Arquivos de Acesso Aleatório

Exercício Acesso Aleatório

Crie um TDA para modelar uma pessoa, com os seguintes atributos: nome, idade e altura. Crie dois programas. O primeiro programa deverá receber entradas do usuário, cadastrando novas pessoas em um *array*. O *array* pode ser estático ou dinâmico. Grave as informações em um arquivo de acesso aleatório (***.bin**). O segundo programa deverá ler o conteúdo do arquivo binário e imprimir as informações cadastradas. A informação do número de objetos do tipo pessoa salvo no arquivo deverá ser extraída do arquivo.