Universidade Federal do Rio Grande do Sul – Instituto de Informática
INF 01203 – Estruturas de Dados – Renata Galante e Viviane Orengo
Semestre: 2008/2 – Turmas A, B e C

Trabalho Final – Busca de Metadados na Biblioteca Digital Brasileira de Computação (BDBComp)

Introdução

Este trabalho tem por objetivo proporcionar aos alunos a oportunidade de aplicar os conhecimentos adquiridos e as estruturas de dados desenvolvidas em aula na solução de um problema real de busca.

Problema

Bibliotecas digitais são compostas por coleções de objetos digitais, como, por exemplo, documentos, imagens, vídeos, mapas, etc. que oferecem serviços aos seus usuários como pesquisa e publicação desses objetos. Além dos objetos digitais, as bibliotecas digitais são compostas por um catálogo de metadados cuja função é descrever, organizar e especificar a forma como os objetos digitais podem ser manipulados e recuperados.

O catálogo da Biblioteca Digital Brasileira de Computação (BDBComp) é composto por diversas coleções de metadados que referenciam artigos científicos da área da computação publicados em conferências nacionais. Esses metadados são representados através de documentos XML com a estrutura hierárquica apresentada na Figura 1.

Cada elemento possui um rótulo representado pela palavra entre os caracteres < e >. O início de um elemento é caracterizado pela seqüência de caracteres `<rótulo>` e o final do elemento por `</rótulo>`. O valor de um elemento é determinado pela palavra ou estrutura entre rótulos de mesmo nome. Por exemplo, o primeiro elemento de rótulo `setSpec` possui valor `109`. O primeiro elemento de rótulo `header` possui valor igual a outros três elementos de rótulos `identifier`, `datestamp` e `setSpec`.

Os registros referentes aos objetos digitais possuem um cabeçalho (`header`) com informações que o identificam de forma única (`identifier`), uma data de criação (`datestamp`) e um número correspondente ao conjunto ou coleção a que pertence (`setSpec`). Além do cabeçalho, cada registro possui uma série de metadados (`metadata`) que referenciam o objeto digital. Entre os metadados estão incluídos, por exemplo, o título (`title`), o ano de publicação (`date`) e o idioma (`language`) do objeto digital. Note que alguns metadados podem possuir cardinalidade maior que um, ou seja, podem aparecer mais de uma vez. Por exemplo, um objeto digital pode possuir vários autores (`creator`). Alguns metadados são opcionais e não aparecerão em todos os registros (`description`).

Existem dois tipos básicos de registros: conferências e artigos científicos. As conferências são caracterizadas pelo metadado `<type>Collection</type>` e os artigos científicos pelo metadado `<type>Text</type>`. Os artigos possuem no cabeçalho o mesmo identificador da conferência onde foram apresentados, seguido do número do artigo. Por exemplo, no cabeçalho da conferência de título `<title>XX Simpósio Brasileiro de Banco de Dados</title>` existe o elemento `<identifier>sbbd2005meta</identifier>`. O registro com o elemento `<identifier>sbbd2005meta001</identifier>` presente no cabeçalho refere-se ao primeiro artigo científico publicado nesta conferência.

```
<ListRecords>
  <record>
    <header>
      <identifier>sbbd2005meta</identifier>
      <datestamp>2005-09-08</datestamp>
      <setSpec>109</setSpec>
    </header>
    <metadata>
      <title>XX Simpósio Brasileiro de Banco de Dados</title>
      <creator>Carlos A. Heuser</creator>
      <date>2005</date>
      <type>Collection</type>
      <identifier>sbbd2005</identifier>
      <language>por</language>
      <coverage>Uberlândia, MG, Brasil</coverage>
      <rights>Sociedade Brasileira de Computação</rights>
    </metadata>
  </record>
  <record>
    <header>
      <identifier>sbbd2005meta001</identifier>
      <datestamp>2005-09-08</datestamp>
      <setSpec>109</setSpec>
    </header>
    <metadata>
      <title>Self Describing Components: Searching for Digital Artifacts on the Web</title>
      <creator>André Santanchè</creator>
      <creator>Claudia Bauzer Medeiros</creator>
      <description>The Semantic Web has opened new horizons in exploring Web functionality.
            One of the many challenges is to proactively support the reuse of digital
            artifacts stored in repositories all over the world. Our goal is to contribute
            towards this issue, proposing a mechanism for describing and discovering artifacts
            called Digital Content Components (DCCs). DCCs are self-contained stored entities
            that may comprise any digital content, such as pieces of software, multimedia or
            text. Their specification takes advantage of Semantic Web standards and
            ontologies, both of which are used in the discovery process. DCC construction and
            composition procedures naturally lend themselves to patternmatching and
            subsumption-based search. Thus, many existing methods for Web searching can be
            extended to look for reusable artifacts. We validate the proposal discussing its
            implementation for agro-environmental planning.</description>
      <date>2005</date>
      <type>Text</type>
      <identifier>sbbd2005article001</identifier>
      <identifier>http://www.sbbd-sbes2005.ufu.br/arquivos/artigo-01-
            BauzerSantache.pdf</identifier>
      <source>sbbd2005</source>
      <language>eng</language>
      <coverage>Uberlândia, MG, Brasil</coverage>
      <rights>Sociedade Brasileira de Computação</rights>
    </metadata>
  </record>
  ...
</ListRecords>
```

Figura 1: Exemplo de metadados da BDBComp

Para que um usuário da BDBComp possa consultar o conteúdo de um artigo científico, é necessário que ele busque no catálogo de metadados as informações referentes ao artigo em questão. Para realizar esta consulta deve existir algum software que busque por determinados metadados em documentos XML.

Tarefa

O(s) aluno(s) deve(m) desenvolver uma biblioteca contendo todas as funções de manipulação das estruturas e as funções de aplicação necessárias para a solução do problema apresentado neste documento. O protótipo a ser construído corresponde à execução de consultas em documentos XML que contém metadados da BDBComp.

São fornecidos alguns documentos XML para teste da aplicação, os quais estão disponíveis em (http://moodle.inf.ufrgs.br/file.php/23/trabalhoed.zip). Para cada documento XML, existem duas versões de arquivos. Uma composta de caracteres acentuados e outra composta de entidades HTML correspondentes aos caracteres acentuados. A segunda versão possui sempre a seqüência de caracteres "HtmlCodes" no nome do arquivo. O(s) aluno(s) pode(m) usar qualquer versão para teste da aplicação construída.

Devem ser desenvolvidos dois componentes do protótipo:

1. uma biblioteca de funções que leia um documento XML (arquivo texto) com os metadados da BDBComp e armazene os dados em alguma estrutura de dados na memória;
2. uma aplicação que peça para o usuário digitar palavras, faça a busca na estrutura armazenada, e liste os artigos científicos que contenham o termo consultado. **A busca deve ser feita a cada caractere digitado**.

A cada busca devem ser listadas as seguintes informações dos artigos: o título (`title`); todos os autores (`creator`); o ano de publicação (`date`); o identificador da conferência onde foi publicado (`source`); a **quantidade de registros encontrados**; e o **tempo gasto na consulta**.

A aplicação deve possui, no mínimo, as seguintes funções de busca:

1. Busca por **título** de artigo
   Classificação obrigatória dos resultados por ordem alfabética de título do artigo
2. Busca por **autor** de artigo
   Classificação obrigatória dos resultados por ordem alfabética de título do artigo
3. Busca por **intervalo de anos de publicação** de artigo
   Classificação obrigatória dos resultados por ordem descendente de ano de publicação do artigo
4. Busca por **título de conferência** (artigos que foram publicados nesta conferencia)
   Classificação obrigatória dos resultados por ordem do identificador do artigo

A aplicação desenvolvida deve ser capaz de ler os registros do arquivo XML para uma única estrutura de dados na memória. Esta estrutura pode ser complexa e construída a partir da combinação de outras estruturas mais simples (como registros, listas, pilhas, árvores, tabelas, grafos). **Mas não é permitido criar mais de uma estrutura!** Também não é permitido o uso de arquivos auxiliares. **O único arquivo que deve ser utilizado é o de entrada** que contém os metadados.

O programa deve medir e mostrar na tela os tempos gastos tanto na **operação de carga** da base de metadados (arquivo XML), quanto nas **operações de busca** efetuadas. Os tempos gastos podem ser verificados utilizando as seguintes bibliotecas:

- time.h (linguagem de programação C). Documentação disponível em
  http://www.cplusplus.com/reference/clibrary/ctime/
- Sysutils (linguagem de programação Pascal). Documentação disponível em
  http://www.freepascal.org/docs-html/rtl/sysutils.

Medir o tempo de cada sub-busca (para cada caractere digitado) e exibir o somatório dos tempos no final da consulta. Por exemplo, a busca por autor de artigo "Eduardo" consiste em 7 sub-buscas ("E", "Ed", "Edu", "Edua", "Eduar", "Eduard" e "Eduardo"), uma para cada caractere digitado pelo usuário. Após o usuário digitar <ENTER>, exibir o somatório dos tempos das 7 sub-buscas. Perceba que a estrutura de dados utilizada será determinante nos tempos das buscas.

A cada sub-busca também devem aparecer os primeiros resultados parciais (os que couberem na tela) e após o usuário digitar <ENTER> deve aparecer o resultado final.

Critérios de Avaliação

O software desenvolvido deve funcionar com todos os documentos XML disponibilizados para teste. Será utilizado um documento diferente dos arquivos de teste durante a avaliação do trabalho. Portanto, é necessário construir a aplicação genérica, ou seja, que funcione com qualquer documento XML com a estrutura apresentada na seção "problema" deste documento.

Serão adotados diversos critérios de avaliação, incluindo:

- quantidade de registros processados e uso da memória;
- tempo gasto na criação da estrutura de dados;
- tempo médio gasto nas buscas;
- escolha da estrutura de dados;
- organização e documentação da biblioteca;
- organização, documentação e interface do programa;
- capacidade de defender o código-fonte durante a apresentação e explicar com detalhes o funcionamento do software.

O programa deve **indicar os tempos gastos na criação da estrutura de dados e o tempo gasto em cada busca**. Quanto mais registros o programa processar e quanto menor o gasto de memória, maior será a nota atribuída a esse critério.

A escolha da estrutura de dados deve demonstrar conhecimento teórico e prático buscando a melhor combinação que atinja os resultados satisfatoriamente. Esse trabalho não avalia apenas o desempenho, mas a capacidade do aluno de criar estruturas elegantes e fáceis de serem mantidas. Para avaliar esse critério, é muito importante que o aluno **DESCREVA COM RIQUEZA DE DETALHES** as estruturas utilizadas no programa. Não esqueça(m) que **APENAS uma estrutura de dados deve armazenar todos os metadados do documento XML e essa mesma estrutura deve ser usada para responder a todas as consultas**. O trabalho consiste justamente em definir a estrutura de dados mais apropriada e que seja eficiente para a maioria das operações. É óbvio que a estrutura escolhida não será eficiente para todas as consultas.

Por fim, serão avaliadas a organização da biblioteca e a organização da aplicação. A interface do usuário também será avaliada. Fica a critério do(s) aluno(s) desenvolver uma interface do usuário gráfica (GUI) ou texto (linha de comando). O importante é que a interface seja atraente, fácil de usar e bem documentada.

Datas Importantes

03/11/2008: Entrega da especificação da estrutura de dados e da justificativa detalhada das escolhas envolvidas na construção desta estrutura. O(s) aluno(s) também deve(m) entregar o código-fonte parcialmente implementado, mesmo que não funcional. A submissão dos arquivos deve ser realizada através do sistema Moodle.

01/12/2008 e **02/12/2008**: Apresentação da aplicação construída e defesa do código-fonte durante o horário de aula. O dia vai depender da turma em que o(s) aluno(s) estiver(em) matriculado(s). A submissão dos arquivos deve ser realizada através do sistema Moodle logo após a aula de apresentação.

Observações

O trabalho deve ser realizado individualmente ou em duplas. Junto do código-fonte deverá ser entregue uma documentação completa incluindo as argumentações que justifiquem a escolha das estruturas de dados utilizadas. O trabalho deve ser apresentado por todos os autores no final do semestre.

Qualquer Ambiente de Desenvolvimento Integrado (IDE) ou compilador pode ser utilizado para implementar o trabalho, **desde que o código-fonte seja escrito em Pascal ou C** (NÃO em Delphi, C++ ou C#). Além disso, **o trabalho não deve utilizar nenhuma estrutura de dados ou artifícios prontos que essas IDE ofereçam**, com exceção de componentes visuais (para aqueles que implementarem a interface do usuário gráfica)

O plágio é terminantemente proibido e a sua detecção incorrerá na divisão da nota obtida pelo número de alunos envolvidos! Para detectar o plágio, além da análise detalhada do código-fonte de todos os trabalhos, será utilizado o software MOSS (http://theory.stanford.edu/~aiken/moss).