



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

RELATÓRIO TÉCNICO

Programa para consultas em arquivo CSV

RESUMO

Nossa escolha do tema do projeto foi o de consultas em um arquivo CSV.

A ideia principal do projeto consiste em indexar um arquivo CSV por inteiro, de forma a separar a indexação por colunas a fim de agilizar a busca por linhas que contenham os valores desejados pelo usuário na coluna especificada. O programa também deve permitir a busca utilizando dois ou mais valores, de forma a filtrar os resultados.

Ao fim do projeto, nosso código foi capaz de realizar as buscas por linhas do arquivo utilizando a coluna e o valor desejado pelo usuário, conforme solicitado. Além disso, ele também permite encontrar linhas através da busca simultânea em duas ou mais colunas. O código foi testado com os arquivos disponibilizados pelo professor e mais alguns outros variados. Sendo assim, todos os requisitos solicitados foram cumpridos.

São José, 30 de Julho de 2021.



MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

INTRODUÇÃO

O programa deve realizar a busca das linhas que contenham os valores desejados na coluna especificada pelo usuário. Deve ser possível realizar a busca filtrando por múltiplas colunas. O diretório e nome do arquivo a ser consultado devem ser informados ao iniciar o programa, que em seguida mostrará uma interface que pede que o usuário digite a coluna e depois o valor que ele quer que a linha tenha nessa coluna.

DESENVOLVIMENTO

Foram criados dois arquivos de código-fonte para o projeto, um main.cpp, onde fica o código da interface do usuário e o sistema.cpp, onde estão as funções primordiais para o funcionamento do sistema.

No código, acabamos utilizando filas, listas, árvores e tabelas hash, de forma a abordar todo o conteúdo aprendido na disciplina.

As filas foram utilizadas na separação das colunas das linhas, além de armazenar outros dados de forma que o programa possa manuseá-los na ordem correta. As listas foram utilizadas para armazenar dados importantes, como o nome de cada coluna, pois essa informação precisaria ser iterada em outras partes do código. A tabela hash foi utilizada de forma que a chave é o nome da coluna e o conteúdo é a árvore contendo a indexação das linhas, o que simplificou demais o resto do código.

RESULTADOS

Após a indexação completa e sem erros do arquivo, a busca dos dados solicitados pelo usuário em um arquivo que continha uma grande quantidade de dados armazenados foi eficiente e rápida. O programa realizou com sucesso a busca por múltiplas colunas também.

Indexar o arquivo e armazenar essa indexação otimiza muito o processamento e o uso da memória, de forma a não precisar abrir o arquivo cada vez que for realizar uma busca, uma vez que a indexação pode ser reutilizada a cada busca.



MANUAL

Para compilar o programa, abra o terminal de comando do Linux, entre no diretório em que os arquivos do programa se encontram e digite:

```
g++ -I. -std=c++14 *.cpp libs/BasicTree.cpp -o programa
```

Para testá-lo, digite:

```
./programa
```

O usuário deve informar o nome do “arquivo.csv”, este deve estar dentro da pasta raiz do programa, além das colunas que ele deseja indexar.

O usuário deve informar a coluna ou as colunas em que deseja pesquisar, na mesma ordem em que aparecem no arquivo;

O usuário deve informar o valor ou valores correspondentes a cada coluna solicitada, na mesma ordem das colunas digitadas;

Exemplos de compilação e teste do programa:

```
fausto@fausto-A410-G-BC44P1: /media/fausto/c07542e0-bd6e-4ea0-a050-f97dbce2d07f/fausto/Documentos/CLionProjects/PROG2/projeto2$ g++ -I. -std=c++14 *.cpp libs/BasicTree.cpp -o programa
fausto@fausto-A410-G-BC44P1: /media/fausto/c07542e0-bd6e-4ea0-a050-f97dbce2d07f/fausto/Documentos/CLionProjects/PROG2/projeto2$ ./programa
teste.csv,Idade

coluna > Idade

valor > 15

123123123-00,João Vitor,15

coluna > Idade

valor > 32

234567789-98,Joaquim José Xavier,32
234567789-64,Manuela Santana,32

coluna > █
```



O programa retorna com as linhas que contém os valores solicitados.

Em caso de pesquisa simultânea em mais de uma coluna, o usuário deve digitá-las separadas por vírgula e sem espaço, além disso, as colunas e valores devem estar na mesma ordem da que aparece no arquivo.

Um exemplo de uso com duas colunas:

```
fausto@fausto-A410-G-BC44P1: /media/fausto/c07542e0-bd6...  
fausto@fausto-A410-G-BC44P1: /media/fausto/c07542e0-bd6e-4ea0-a050-f97dbce2d07f/fausto/Documentos/CLionProjects/PROG2/projeto2$ ./programa  
teste.csv, Nome, Idade  
  
coluna > Nome, Idade  
  
valor > Antonieta de Barros, 51  
  
424258583-67, Antonieta de Barros, 51  
424258583-63, Antonieta de Barros, 51  
424258583-63, Antonieta de Barros, 51  
  
coluna > 
```



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

CONCLUSÃO

Após a conclusão do projeto, realizamos múltiplos testes. A busca foi realizada com sucesso em todos os casos testados condizentes com as condições de uso descritas, seja em apenas uma coluna ou mais. Portanto, o programa foi capaz de realizar tudo o que foi solicitado com a devida rapidez e eficiência.

BIBLIOGRAFIA

Utilizamos apenas os livros de exemplos disponibilizados no moodle.