



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE - CTS
DEPARTAMENTO DE COMPUTAÇÃO – DEC

DISCIPLINA: LINGUAGENS DE PROGRAMAÇÃO 2
PROFESSOR ANTONIO CARLOS SOBIERANSKI
a.sobieranski@ufsc.br

ENUNCIADO TRABALHO T1 – “FILE MANIPULATOR HACK”

A ser desenvolvido individualmente ou em duplas

Elementos utilizados: C++, entrada e saída em tela, vetores e strings, leitura e gravação de arquivos. Os arquivos possuem palavras (quantidade variada), sem espaços, separadas por quebra de linha, e podem ter a mesma palavra no mesmo arquivo e/ou em arquivos distintos.

O nosso primeiro programa entregável corresponderá a um gerenciador de arquivos em disco que possui o seguinte menu abaixo:

FILE MANIPULATOR HACK

Select an option:

1. Open a File...
 2. Search for Substrings...
 3. Remove Words Containing a Substring
 4. Remove all Repeated Words
 5. Show Statistics
 6. Exit
-

Option:

1. A opção 1 abre um arquivo. Deve imprimir em uma nova linha o texto abaixo (*cout*), e em seguida abrir o arquivo, carregando-o em memória. Vários arquivos podem ser carregados em sequência na mesma execução do programa (inclusive o mesmo anteriormente carregado).

cout << “Enter with a file to open:”

2. A opção 2 localiza a ocorrência de determinada substring em todos os arquivos que foram carregados na opção 1, conforme abaixo:

cout << “Enter a substring to search:”

...e imprime em tela as ocorrências da substring. O *output* em tela deve apresentar em qual arquivo determinada substring (o nome do arquivo) existe, e a palavra completa.

Exemplo de busca da substring “izz”

File1.txt has substring “izz”: pizza

File1.txt has substring “izz”: frizzy

File1.txt has substring “izz”: pizzicato

File2.txt has substring “izz”: blizzard

File3.txt has substring “izz”: frizzy

File3.txt has substring “izz”: frizz

File3.txt has substring “izz”: frizzy

3. A opção 3 remove todas as substrings contendo determinada palavra dos arquivos que foram carregados (não remover dos arquivos originais), conforme *output* abaixo.

cout << "Enter a substring to remove all occurrences:"

4. Eliminar a ocorrência de todas as palavras **REPETIDAS** que possam por ventura ocorrer no mesmo arquivo, e em outros arquivos. Somente a primeira que surgir em ordem de arquivo e *index* do vetor de palavras deve ser mantida, e todas as repetições eliminadas.

5. Apresenta estatísticas de cada fonte (arquivo) em tela. Se palavras foram eliminadas, estas estatísticas devem ser atualizadas quando chamada a opção.

Exemplo:

File Manipulator Hack Statistics:

File1.txt has 350000 words

File2.txt has 1500 words

File3.txt has 600 words

6. Sair do sistema e gravar um único arquivo contendo TODAS as palavras correntes (após as manipulações dos menus), chamado "**concatenation.txt**", por ordem de arquivo inserido e depois pelo *index* do vetor, em sequencia separadas por quebra de linha.

Requisitos:

- Usar `std::vector<string>` para armazenar os arquivos, e `ifstream` e `ofstream` para acesso aos arquivos;
- compilar com GCC padrão (GNU-gcc, mingw). Vide *VirtualMachines* no Moodle, ou instalação própria. **Compilar com "g++ *.cpp -o exec".**
- Não compilou e não foi possível executar, nota zero.
- Nota proporcional aos itens implementados.
- Entregar somente *código-fonte* (*h*, *hpp*, *cpp*) em um arquivo compactado em formato "zip".

Dica: o armazenamento dos arquivos de palavras pode ser realizado com vetores multidimensionais, conforme slides no Moodle.