

Universidade Federal do Espírito Santo
Departamento de Engenharia da Computação
Curso de Engenharia da Computação

1º Trabalho Prático

André Luiz Piona de Araujo

Vitória–ES

Maio de 2016

1º Trabalho Prático

Trabalho apresentado na disciplina de Estrutura de dados I do curso de Engenharia da Computação na Universidade Federal do Espírito Santo, sob orientação da Profª Patrícia Dockhorn Costa

1 INTRODUÇÃO

Wiki é um conjunto de páginas interligadas, que podem ser visitadas e editadas por qualquer pessoa. Nos últimos anos tem crescido bastante a quantidade de usuários que estão utilizando esse tipo de serviço.

O objetivo desse trabalho é criar uma versão simplificada de um Wiki, utilizando listas encadeadas na linguagem C. Essa versão será chamada WikED, e usará como recurso arquivos de texto, tanto como conteúdo para as futuras páginas criadas pelo programa, como para dar comandos ao programa.

O WikED será feito utilizando uma lista de “páginas”, expandido a cada nova página adicionada. Essas páginas receberão “contribuições” que são os arquivos com conteúdo para elas, essas contribuições serão feitas por “editores” que serão adicionados pelo usuário do programa. As páginas também terão “links” interligando elas umas as outras à vontade do usuário, assim como os Wikis.

2 METODOLOGIA

O programa utiliza arquivos de texto para seu funcionamento, apenas o arquivo log.txt não é informado pelo usuário. O arquivo log.txt é o arquivo que registrará todos os erros, e resultados da função “CAMINHO”.

Logo ao ser executado, o programa pede o nome do arquivo de texto com os comandos, esse arquivo de texto tem que estar no mesmo diretório que o programa, assim como todos os arquivos com conteúdos das páginas que serão criadas ao longo do programa.

2.1 Estruturas

Foram criadas 4 estruturas para o funcionamento do programa, “Páginas”, “Arquivos”, “Links” e “Editores”.

A estrutura de “Páginas” possui as variáveis “Nome da Página”, “Arquivo da Página”, “Próxima Página”, “Página Anterior”, “Links” e “Arquivos” (que são as contribuições). Utilizei o encadeamento duplo para essa lista, pois sendo a lista mais utilizada pelo programa, optei pelo método que tenha mais modos de utilizar a lista. A lista criada por essa estrutura é inicializada no começo do programa no arquivo .c principal, pois é modificada e retornada por praticamente todas as funções, as exceções são “INSEREEDITOR”, “RETIRAEDITOR”, “IMPRIMAPAGINA”, “IMPRIMAWIKED” e “FIM”.

A estrutura de “Arquivos” possui as variáveis “Nome da Página”, “Arquivo da Contribuição”, “Próximo Arquivo”, “Arquivo Anterior” e “Retirado”. Nessa lista optei por usar o encadeamento duplo, pois será necessário adicionar as contribuições na mesma ordem em que serão adicionados. A variável “Retirado” foi criado como função booleana, para saber se o arquivo foi retirado sem a necessidade de ser realmente deletado.

A estrutura de “Links” possui as variáveis “Nome da Página Destino”, “Arquivo da Página Destino”, “Próximo Link”, “Página Destino”. A “Página Destino” é a variável com ponteiro da memória da página destino utilizada na criação do link.

A estrutura de “Editores” possui as variáveis “Nome do Editor” e “Próximo Editor”. É uma lista simples, porém independente igual a lista de páginas, e também inicializada logo no começo do programa no arquivo .c principal. Essa lista serve apenas para saber os editores que estão inseridos no WikiED. É modificada e retornada nas funções “INSEREEDITOR” e “RETIRAEDITOR”.

2.2 Funções

Apenas 2 estruturas necessitaram de uma função com o único objetivo de inicializar-las com valor “NULL”, que foram “Páginas” e “Editores”. O restante recebia o valor “NULL” ao ser inserido uma nova página na lista de páginas, logo não tinha a necessidade de utilizar uma função específica.

As funções, em geral, são as pré-determinadas no enunciado do trabalho, que são:

- a) **“INSEREPAGINA”**: A função recebe como dados do usuário o nome da página e o arquivo que terá o conteúdo da página. Logo no começo da função, é verificado se o nome da página ou o arquivo da mesma já foram utilizados anteriormente, caso não e
- b) **“RETIRAPAGINA”**: A função recebe como dado do usuário o nome da página apenas, caso a página com nome igual seja encontrada, será deletada. Nessa função foi encontrado um erro ao utilizar ponteiro apontando para a próxima página da pagina anterior, e vice-versa. Não foi encontrado o motivo desse erro, porém foi resolvido ao criar uma variável que apontava diretamente para proxima página, ou para página anterior.
- c) **“INSEREEDITOR”**: A função recebe como dado do usuário o nome do editor que deseja que seja adicionado. É verificado se existe algum editor com nome igual criado anteriormente, caso não tenha sido, é criado um editor novo de mesmo nome.
- d) **“RETIRAEDITOR”**: A função recebe como dado do usuário o nome do editor para ser retirado. É verificado se existe o editor com nome igual, caso seja encontrado, é deletado.
- e) **“INSERECONTRIBUICAO”**: Essa função recebe como dados do usuário o nome do editor, da página da contribuição e do arquivo com o conteúdo que será adicionado. A função verifica se existe a página e o editor com nomes iguais, a contribuição é adicionado caso não encontre.
- f) **“RETIRACONTRIBUICAO”**: Essa função recebe como dados do usuário o nome do editor, da página da contribuição, e o arquivo com a constribuição. A função verifica se existe a página, se possui o arquivo como contribuição, e se o editor é o mesmo que adicionou inseriu a contribuição. Caso não tenha erros, a contribuição recebe “1” na variável “Retirada”, para não ser necessário deletar.
- g) **“INSERELINK”**: A função recebe como dados do usuário a página de origem, e a página de destino. Caso ambas já tiverem sido criadas, será feito um link da página de origem para a de destino.

- h) **“RETIRALINK”**: A função recebe como dados do usuário a página de origem e destino. Caso ambas já tiverem sido criadas, e já possuem link da página de origem para a de destino, o link é deletado.
- i) **“CAMINHO”**: A função verifica se possui um link entre as páginas de origem e destino informados pelo usuário. Caso ambas as páginas já tenham sido criadas, será feito a verificação do link, e o resultado é então escrito no log.txt.
- j) **“IMPRIMEPAGINA”**: Essa função recebe do usuário a página que deseja que seja criado o arquivo de texto com as contribuições recebidas até então. A função verifica se a página existe antes de tentar gerar o arquivo. Como a lista de contribuições é utilizada no “Histórico de Contribuições” e para passar os arquivos com as contribuições, foi necessário criar um método para passar o histórico sem a variável da lista arquivos fique “NULL”, o jeito encontrado foi usar um “while” e um “if” ao invés de apenas um “while”. Além disso, como dito anteriormente, foi utilizado o encadeamento duplo para conseguir passar o conteúdo na mesma ordem em que foi adicionado.
- k) **“IMPRIMEWIKED”**: A função é apenas um “for” passando o nome de cada página da lista de páginas para a função “IMPRIMEPAGINA”.
- l) **“FIM”**: Essa função libera a memória utilizada pelo programa e finaliza o mesmo.

3 CONSIDERAÇÕES FINAIS

Ao longo do trabalho foi notado a dificuldade de se fazer um projeto maior sem o planejamento adequado.

No começo foi feito função por função e modificando as estruturas para atender as necessidades das funções, porém cada vez que avançava mais no trabalho, ficava cada vez mais complexo fazer uma modificação. Não bastava mudar apenas uma estrutura por exemplo, as vezes era necessário modificar também algumas funções ou o arquivo principal, e nesses casos, descobrir a onde, se tornava um problema cada vez maior.

4 REFERÊNCIAS

CELES, Waldemar., CERQUEIRA, Renato., RANGEL, José Lucas. **“Introdução a Estrutura de Dados”**. Campus. 2004