Organizador de programação de seminários.

Daniel Fernandes Campos

Engenharia de computação – Universidade Estadual de Feira de Santana (UEFS)

CEP: 44036-900 – Feira de Santana – BA – Brasil

dfc152@gmail.com

Resumo. *Devido a necessidade de organizar as apresentações que acontecem na Feira de Graduação foi solicitado que desenvolvesse um software para tal. Durante a criação deste algoritmo foram realizadas sessões de discussão, bem como pesquisa sobre certos temas, como arquivo binário e algoritmos de ordenação. Por fim no código criado é capaz de realizar o cadastro de uma palestra a ser apresentada, receber o status de aprovado ou negado para as palestras e suas notas e por fim gerar e mostra a programação.*

Palavras chaves: algoritmo, semana do mestrado, organização.

1. Introdução

Devido a necessidade de organizar as palestras da Feira de Graduação foi solicitado que fizesse um algoritmo capaz de receber as palestras e suas avaliações (aprovados ou negados) e suas notas, e gerarmos uma programação baseada nas 64 melhores palestras sendo divididas por áreas que serão apresentadas em 2 dias, guardando as demais para eventuais reposições se necessário.

Os conhecimentos que foram necessários de serem aprendidos para resolução desses problemas foram manipulação de arquivos binários, algoritmos de ordenação (como eles funcionam), como utilizar memória alocada dinamicamente e um maior estudo sobre structs.

Durante a criação deste algoritmo foi necessário a realização de pesquisas sobre os temas acima citados, discussões em grupo sobre partes do código, como seria feito ou o que poderia ser melhorado, com colegas e algumas horas de codificação.

2. Desenvolvimento

O desenvolvimento do programa foi feito seguindo os passos 1 a 5 respectivamente conforme o papel base que nos foi dado. Assim sendo para início foi feito a possibilidade de cadastro de novas palestras, sendo também possível fazer cadastro pelo .txt, com os dados conforme a primeira struct da Figura 1. Para fazer isso foi necessário entender o que são structs segundo Cassavella(2014) structs são variáveis que são feitas de outras (chamadas de membros). Também é necessário conhecer como manusear arquivos .txt sendo usadas as funções de arquivos conforme é apresentado por Sonoda.

No .txt a primeira linha fica em branca e posteriormente é escrito as palestras seguindo um padrão: 1º é escrito o título, posteriormente é escrito o status (será falado mais a frente), então sua nota, seu autor, se existe coautor, caso exista o nome, o nome do

orientador, se existe coorientador, caso exista seu nome, sua área de conhecimento e por fim seu código. Assim sendo cada palestra cadastrada pode tomar de 11 linhas a 9, sendo colocada outra palestra logo abaixo caso exista.

```
// Structs
struct dados_cadastro{ // dados completos da palestra
        int codigo;
        char titulo[200];
        int area_do_conhecimento;
        char autor[200];
        char orientador[200];
        int status; // confirmado ou negado
        int verificacao_coautor;
        char coautor[200];
        int verificacao_coorientador;
        char coorientador[200];
        float nota;
    };
struct organizador_notas{ // dados usados para organizar as palestras
    int posicao_da_palestra;
    float nota;
    char titulo_trabalho[200];
    int area_do_conhecimento;
    int codigo;
};
```

Figura 1.Structs usadas.

Após a implementação desta parte foi implantada a avaliação sendo necessário a realizar uma leitura do arquivos onde existe os cadastros das palestras para realizar a avaliação das que ainda não foram, para realizar esta avaliação foi escolhido fazer uma variável extra na struct que guarda o estado (aprovado(1), negado(0) ou não avaliado(-1)), sendo esta uma escolha pessoal outro métodos que podiam ser usadas seriam: zerar o código, zerar a nota, salvar outra palestra por cima das negadas e salvar em outro arquivo... Quando uma palestra recebe o status de aprovada é solicitado que entre uma nota de 0 a 10, para notas decimais deve usar ‘,’ ao invés de ponto exemplo: “ 6,7 ”. Após o término da avaliação as palestras que não foram negadas são salvas no arquivo .txt. Então é chamado automaticamente um outra opção do menu a de gerar programação.

A função de gerar programação realiza a leitura do arquivo .txt e pega as palestras aprovadas colocando elas em um vetor de ordem decrescente de nota. Posteriormente é escrito em um arquivo .bin a quantidade de palestras disponíveis para apresentá e as palestras em ordem de apresentação, usando a função de arquivo binário conforme Fei(2018) fwrite(*ponteiro_para_o_que_quer_escrever, tamanho, quantidade, arquivo), também é escrito em outro .txt esta mesma programação. A escrita das palestras no arquivo binário e no arquivo .txt acontecem em sequência, assim sendo a ordenação do vetor de palestras em ordem crescente de posição de apresentação (de 1 até o final) acontece no momento da escrita do binário.

Com o arquivo .bin, que é lido com fread que recebe os mesmos argumentos do fwrite conforme Fei(2018), é possível ver a programação através do código, este realiza leitura e

mostra na tela a programação para os 2 dias com uma palestra a cada 15 minutos tendo início às 7:30 e término às 17:30 com 2h de almoço das 11:30 às 13:30.

Também é possível marcar uma palestra como desistente após ela ter sido cadastrada. Quando essa opção é selecionada é mostrado na tela todos os projetos aprovados e então é pedido para digitar o código do desistente, posteriormente todas as outras palestras salvas, sendo chamado a função que gera programação logo em seguida.

Para realização da ordenação é criado um vetor de struct auxiliar que recebe a primeira palestra cadastrada na primeira posição e coloca as demais na última posição, então comessa a comparar a nota desta (da última posição) com todas as outras que já foram colocadas no vetor se a nota dela (da última posição) for maior soma 1 na posição da outra palestra senão (se for menor ou igual) soma 1 na posição desta e por fim ela é salva ao lados das outras já salvas no vetor para ser considerada nas próximas comparações. Essa foi uma escolha pessoal podendo também ser usado um algoritmo de ordenação qualquer mostrado por wikibooks(2017).

Um problema que havia era na leitura dos títulos, nome de autores, coautores , entre outros vinha com um ‘\n’ do arquivo e caso fosse salvo desse jeito atrapalharia a leitura futuras leituras do arquivo, portanto foi criada uma função para remover-lo. esta função procurava o ‘\n’ e trocava por um ‘\0’, sendo isso uma outra escolha pessoal também sendo possível pega o tamanho da string e salvar naquela posição o ‘\0’.

Outra escolha pessoal para resolver este problema foi a utilização de vetores alocados dinamicamente em vez de lista encadeada devido ao maior conhecimento do assunto. Conforme Feofiloff (2018) lista encadeada é uma sequência de objetos de um mesmo tipo que estão guardados na memória RAM e que guardam alguma informação (ou informações) e um ponteiro para a localização do próximo elemento. A lista encadeada poderia ter sido usada nas opções de apagar um candidato ou na ordenação das palestras por nota.

Um dos problemas enfrentados na criação deste algoritmo foi quando a função de tirar enter era chamada e a string nao apresentava um ‘\n’ acontecendo do programa crasha. Suposição devido ao fato de não haver um ‘\n’ era passado por todo a string e continuava acessando outros espaços de memória, para consertar isso foi necessário apenas acrescentar uma condição de parada no loop, quando achar um ‘\0’.

3. Conclusão.

Na produção deste código todos os objetivos foram completos, sendo criado um software capaz de receber palestras, suas avaliações e notas, gerar e mostra a programação.

Porém uma das sugestões de melhoria que podem ser adotadas são: modificar o código aceito para as palestras para ser float ou double (assim aumentando a quantidade de vezes que esse código pode ser utilizado), apesar de serem possíveis fazer palestras até o ano de 9999 seguindo o código modelo apresentado (sendo ele feito da seguinte forma: 4 dígitos do ano, seguido por 2 dígitos de apresentações da área, seguido por 2 dígitos da área).

Outra sugestão de melhoria seria a modificar o código para ao invés de usar tabela ASCII utilize caracteres unicode, permitindo assim a utilização de uma gama maior de caracteres.

Referências

Casavella, Eduardo.(2014) Struct em c. Disponivel em:<http://linguagemc.com.br/struct-em-c/>. Acesso em: 24/12/2018

Sonoda, Eloiza. Manipulação de arquivos. Disponivel em: <https://www.ime.usp.br/~elo/IntroducaoComputacao/Manipulacao%20de%20arquivo.htm>. Acesso em: 24/12/2018.

Wikibooks.(2017) Programar em c/Algoritmos de ordenação. Disponivel em: <https://pt.wikibooks.org/wiki/Programar_em_C/Algoritmos_de_ordenação>. Acesso em: 24/11/2018.

Fei.(2018) Arquivos binários. Disponivel em: <https://fei.edu.br/~isanches/nca210/arquivos_binarios.html>. Acesso em: 24/11/2018.

Feofiloff, Paulo.(2018) Listas encadeadas. Disponivel em: <https://www.ime.usp.br/~pf/algoritmos/aulas/lista.html>. Acesso em: 25/12/2018.