Universidade de Brasília – UnB
Instituto de Ciências Exatas – IE
Departamento de Ciência da Computação – CIC

Trabalho 02

Disciplina: *Organização de Arquivos – Turma A*

Semestre: 1º/2016

Prof.: André Drummond

Título: *Processamento Co-sequencial*

Entrega: 05/07/2016 – 23h55 (Arquivos-Fontes via Aprender.unb.br, num único arquivo zipado. Incluir um comentário com os nomes e matriculas dos alunos. O arquivo zipado deve ser a matricula do líder do grupo, ex.: 0912345.zip)

1. Objetivos

O objetivo deste trabalho é que o aluno aprenda a realizar algoritmos de indexação, ordenação, busca e processamento co-sequencial. Realizando um trabalho de pesquisa utilizando algoritmos similares aos vistos em sala de aula.

Implementar, utilizando obrigatoriamente a linguagem C ou C++.

2. Especificação

2.1 Desenvolver a implementação de um algoritmo que realize a criação de um arquivo de índices a partir dos arquivos de *benchmarks (lista1.txt, lista2.txt e lista3.txt)* disponíveis. O programa deve receber os arquivos por linha de comando e assumir que o primeiro arquivo é de registros de tamanho fixo e os outros arquivos são de tamanho variável. O nome do programa deve ser 'gera_index'.

Ex:.\gera_index lista1.txt lista2.txt lista3.txt

O programa deve gerar arquivos de índices primários para cada arquivo de dados. Sendo que deve existir um arquivo de índices primários para cada arquivo de *benchmark*. A chave primária é do tipo: 'ID001'. Este arquivo deve ter como nome: index_<nome_arquivo_de_dados>.txt. Por exemplo, para o arquivo lista1.txt, o arquivo de índices primários deve ser index_lista1.txt.

Todos os arquivos de índices devem estar ordenados por chave de forma crescente. O arquivo de índice primários do arquivo lista1.txt deve incluir um número de três caracteres como cabeçalho indicando a quantidade de registros apagados no arquivo de dados (inicialmente deve ser ZERO).

O programa deve também gerar um arquivo de índice secundário, utilizando listas invertidas encadeadas, para permitir fazer busca por nome dos alunos no arquivo *lista1.txt*. Esse arquivo de índice deve ter o nome sec_lista1.txt.

Note que os arquivos de Dados devem permanecer sempre em memória secundária e ser lidos somente registro a registro.

2.2 Desenvolver um programa ('Unb_system.c') que mostre o seguinte menu:

```
Sistema de Notas UnB v0.1 beta
1. Inserir novo aluno na UnB
2. Jubilar um aluno da UnB
3. Dar nota ao aluno numa disciplina
4. Relatório
5. Buscar
6. Sair
```

Caso a opção 1 seja escolhida, o programa deve inserir um novo registro no arquivo lista1.txt (no final do arquivo). O programa deve gerar a chave primária automaticamente. Verificando qual a chave primária maior no arquivo de índices. Os outros dados são preenchidos pelo usuário. O programa deve inserir essa nova chave (e dados necessários) no arquivo de índice primário.

Caso a opção 2 seja escolhida, o programa deve simplesmente colocar -1 na posição da NRR no arquivo de índice primários. O programa deve incrementar a quantidade de arquivos apagados no arquivo de índices primário. Caso existir mais de 5 registros apagados, o programa deve então apagar todos as entradas nos arquivos *benchmarks (lista1.txt, lista2.txt e lista3.txt)* referentes aos registros apagados e refazer os arquivos de índice (chamando o programa 'gera_index' já implementado).

Caso a opção 3 seja escolhida, o programa deve pedir a chave primária do aluno e a disciplina: 1 (referente a lista2.txt) ou 2 (referente a lista3.txt), verificar que o aluno esta no arquivo lista1 (mediante uma busca binária no arquivo de índice), e colocar a nota no arquivo lista2.txt ou lista3.txt, dependendo da disciplina escolhida. Se o aluno já tem uma nota, esta deve ser atualizada. Assumir que NUNCA será atualizada de forma de mudar a quantidade de caracteres da nota.

Caso a opção 4 seja escolhida o programa deve fazer um processamento co-sequencial dos 3 arquivos para gerar o relatório da seguinte maneira:

<MATRICULA NOME OPERAÇÃO CURSO TURMA>
Computação Quântica Avançada III: <nota>
Modelagem Aeroespacial Alienígena II: <nota>
<MATRICULA NOME OPERAÇÃO CURSO TURMA>
Computação Quântica Avançada III: <nota>
Modelagem Aeroespacial Alienígena II: <nota>

Na opção 5 o programa deve perguntar o nome do aluno e buscar no arquivo secundário da lista1.txt, pelo aluno. Se não encontrar deve indicar. Se achar o aluno deve então procurar nos arquivos de índice primário dos arquivos lista2.txt e lista3.txt (por meio da chave primaria), obter as respectivas PRR para os arquivos de dados e mostrar na tela as notas dos alunos em cada disciplina.

Todos os alunos do arquivo lista1.txt devem estar no relatório mesmo sem nota (nesse caso deve ser colocado como nota na disciplina o valor de SR). O programa deve detectar erros como alunos com nota que não estão no arquivo de lista1.txt ou alunos com mais de uma nota numa mesma disciplina. O relatório deve dar uma mensagem de erro na hora do erro acontecer, mas deve terminar de mostrar o relatório.

Note que os arquivos de Dados devem permanecer sempre em memória secundária e ser lidos somente registro a registro. Os arquivos de índices podem ficar sempre em memória primaria, mas devem ser atualizados (em memória secundaria) após o usuário realizar alguma operação. Toda busca deve ser feita nos arquivos de índice, nunca nos arquivos de dados.

3. Grupos

Neste projeto será permitido a formação de grupos de no máximo 2 (dois) alunos. A partir do grupo formado, deverá ser indicado um líder que será o responsável pelo envio dos arquivos fontes para o sistema *Aprender.unb.br*. Somente serão aceitos os arquivos fontes enviado pelo líder do grupo.

4. Critérios de Correção

- Trabalho identificado como cópia entre alunos ou de algoritmos da Internet (qualquer uma das partes) recebe a nota ZERO no trabalho.
- Os trabalhos devem seguir boas práticas de programação: Identação, não declarar variáveis no meio do código, comentários, etc.
- A Parte 2.1 da especificação vale 2 pontos.
- A Parte 2.2 vale 8 pontos.
- Não serão aceitos trabalhos atrasados EM NENHUMA HIPOTESE.