Instituto Federal Sudeste de Minas Gerais – Campus Barbacena
Curso de Tecnologia em Sistemas para Internet
Disciplina: Estruturas de Dados II
Prof. Wender Magno Cota

Segundo Trabalho Prático – Ordenação Externa

1. Introdução

O objetivo deste trabalho é projetar e implementar um sistema para ordenar arquivos que não cabem na memória principal, o que força a utilizar um algoritmo de ordenação externa.
Existem muitos métodos para ordenação externa. Entretanto, a grande maioria utiliza a seguinte estratégia geral: blocos de entrada tão grandes quanto possível são ordenados internamente e copiados em arquivos intermediários. A seguir os arquivos intermediários são intercalados e copiados em outros arquivos, até que todos os registros são intercalados em um único arquivo ordenado.

1. Divida o arquivo original em blocos que caibam na memória principal.
   Enquanto não atingir o final do arquivo
   - Carrega para a memória principal um bloco de registros;
   - Ordena o bloco na memória principal;
   - Escreva o bloco ordenado em arquivos temporários;

2. Abra todos os arquivos temporários gerados.
3. Crie um buffer de entrada para cada arquivo temporário. O tamanho n deve ser passado pela linha de comando.
4. Preencha cada buffer de entrada com os n primeiros registros de cada arquivo.
5. Crie uma fila de prioridades com tamanho igual ao número de arquivos temporários gerados;
6. Crie um buffer de saída de tamanho M.
7. Enquanto a fila de prioridades não estiver vazia
   - se o buffer de saída estiver cheio então
     escreva seu conteúdo no arquivo final
   - retire e escreva no buffer de saída o primeiro item da fila de prioridades.
   - Se o buffer de entrada de onde o último retirado estiver estiver vazio então
     preencha-o com os próximos registros, se existir, do arquivo temporário associado a ele.
   - Adicione a fila o próximo item do buffer de entrada de onde o último retirado na fila de prioridade originou.
8. Apague todos os arquivos temporários.

Algoritmos e Estruturas

a) O TAD fila de prioridades implmentado:
   1) Usando heap;
   2) Usando uma lista sismplesmete endadeada ordenada
b) Use o quicksort para ordenar os dados em memória primária.
c) Implemente os buffers(entrada e saída) como TAD.

Restrições

1. O código deve ser escrito em C.
2. A documentação do TP deve conter no máximo 8 páginas.
3. O seu código deve compilar usando C padrão ANSI. Basta evitar utilizar funções específicas do Windows ou Linux.
4. O seu código deve compilar sem nenhuma warning! Você pode verificar se seu código compila sem warnings com o seguinte comando no Linux

gcc -Werror <arquivos.c>

Entrada
Deverá ser fornecido, através da linha de comando, o nome do arquivo a ser ordenado, o tamanho de cada bloco, o tamanho do buffer de entrada e de saída.

Para testar seu sistema, gere um arquivo com 1.000.000 registros (com 512 bytes cada), sendo que cada registro possua um campo chave, que será usado na ordenação. Faça a medida de tempo necessário para ordenar este arquivo.

Saída
O arquivo ordenado juntamente com o tempo gasto para ordená-lo.

Documentação

Escreva um documento explicando seu código(e não apenas colocando trechos dele) e avaliando o desempenho de sua implementação. Separe em 5 seções.
1 – Introdução
Escreva uma pequena introdução, dando uma ideia geral do sistema e de cada um dos algoritmos e estruturas usadas.

2 – Implementação
Descreva os algoritmos implementados.

3 – Experimentos
Você deverá rodar o sistema usando a fila de prioridades implementada por heap e lista ordenada, bem como para buffer de entrada e saída de tamanhos diferentes.. Apresente o tempo de execução em gráficos ou tabelas e faça uma análise dos mesmos.

4- Conclusões
Resuma o que você fez e deixou de fazer neste TP. Explique quais foram as dificuldades que você encontrou para fazer este TP.

5 – Referências
Cite as fontes utilizadas para realização deste TP, mesmo que sejam sites da internet.
Obs.:
a) O trabalho pode ser feito em dupla;
b) Data de Entrega: 11/06/2015;
c) Caprichem no código fonte: comentários na dose certa, identação e modularização;
d)
Comecem logo pois a data de entrega jamais estará tão distante como agora.
Obs: