6897 - Organização e Recuperação de Dados: Segundo Trabalho

Chen Po Hsiang[1], **Rafael Sanchez**[1]

[1]Departamento de Informática – Universidade Estadual de Maringá (UEM)
Maringá – PR – Brasil

ra83473@uem.br, ra82357@uem.br

Abstract. *This report lists a set of instructions on how to compile and operate a program, which is used to create and maintain a binary file of records. External fragmentation is managed through a free space list (FSL), and the indexation is made through a B-Tree file.*

Resumo. *Esse relatório descreve instruções de operação de um programa que cria e mantém um arquivo binário de registros, o qual possui uma lista de espaços disponíveis (LED) para controlar a fragmentação externa. O programa também gerencia um arquivo de índices em árvore B, para referenciar os registros do arquivo principal.*

Instruções de Compilação

Os arquivos que compoem o projeto são os seguintes:

- *main.c*: Arquivo principal com as chamadas de menu e interação com o usuário.
- *arq_reg.h*: Cabeçalho das funções que operam sobre os arquivos de catálogo e de registro. Cada função está comentada de forma a explicitar como ela opera.
- *arq_reg.c*: Código fonte das funções descritas no cabeçalho acima.
- *arvore_b.h*: Cabeçalho com as funções e dados necessários para se manter um arquivo de índice para o arquivo de registros presente em *arq_reg.h*.
- *arvore_b.c*: Código fonte das funções descritas no cabeçalho acima.

O projeto vem com um Makefile para facilitar a compilação em ambiente Linux.

Linux

Para compilar no Linux, abra o terminal no diretorio raiz do projeto. Então basta entrar com o comando ”make”nesse diretório. O Makefile será chamado e um executável será criado no diretório ”./bin”.

Windows

Usando o MinGW, as instruções sao semelhantes às usadas para compilar no Linux. Através de uma IDE, é necessario criar um novo projeto com os arquivos main.c, arq_reg.c, arq_reg.h, arvore_b.c e arvore_b.h. Nenhuma biblioteca adicional eh necessaria.

É importante salientar que a pasta ”./res”deve estar acessível a partir do diretório de execução, caso deseje-se utilizar os arquivos padrão de catálogo e de registro.

Instruções de Execução

Linux

Basta executar o comando "./bin/gerenciador_arq_led.out <PARAMETROS>"

Em <PARAMETROS> é possível inserir os seguintes modos:

- -r <CAMINHO>: Especifica o caminho desejado para o arquivo de registro a ser editado. Caso nao seja especificado, o padrão usado na execução eh "./res/rezistro.rez", a partir do diretório de execução.

Windows

A execução é feita de modo semelhante àquela feita em Linux, mas caso seja usada uma IDE, os parâmetros de execução devem ser passados nas propriedades do projeto.

O executável também pode ser chamado por linha de comando usando o *cmd* do Windows. Assim, os parâmetros podem ser passados da mesma forma que em Linux.

Decisões de Projeto

Arquivos

O arquivo de registro padrão para alteração é "./res/rezistro.rez". Entretanto, esse pode ser alterado na execução através do parâmetro -r", como descrito na sessão sobre execução. Seu respectivo arquivo de indice será criado no diretório "./res"com o mesmo nome do arquivo de registros, mas com a extensão *.idx*. Portanto, o arquivo de índice é criado em *./res/rezistro.idx* por padrão.

O arquivo de registro possui em seus dois primeiros bytes um *uint16_t* que indica qual é o *offset* do primeiro espaço disponível no arquivo.

Os registros e espaços possuem tamanho variável, sendo seus primeiros 2 bytes um *uint16_t* indicando seu tamanho. No caso dos registros, esse tamanho é seguido pelo ID do registro e seus campos, separados pelo byte de caractére '|'.

Para os espaços disponíveis, os 2 bytes de tamanho são seguidos por um byte de caractére '*', seguido por dois bytes com um *uint16_t*, o qual sinaliza o *offset* do próximo espaço disponível da lista.

Quanto ao arquivo de índices, ele é organizado em formato de árvore B, com registros de tamanho fixo. A ordem da árvore é fixada por uma diretiva de compilação no cabeçalho *arvore_b.h*.

Os dois primeiros bytes do arquivo de índice contém o ponteiro de uma PED, utilizada para gerenciar espaços em branco (os registros de indice têm tamanho fixo). Já os dois bytes seguintes guardam o RRN da página raiz da árvore B, dentro do arquivo de índice.

Tipos

Foi utilizado para os marcadores de *offset* o tipo *uint16_t* da biblioteca *stdint.h*. Decidiu-se usar esse tipo de tamanho fixo ao invés do *short int* para garantir maior portabilidade ao código.

Como os marcadores de *offset* são inteiros sem sinal, optou-se por marcar o fim da lista de espaços disponíveis com 0 (zero) ao invés de -1. Ou seja, o último elemento da LED aponta para o cabeçalho da lista. O mesmo vale para a PED do arquivo de índices.

Usar inteiros sem sinal permite referenciar um maior espaço de memória em disco, uma vez que a representação em complemento de dois reduz à metade o tamanho máximo representável de um inteiro.

Importação do Catálogo

Arquivos de catálogo são arquivos texto usados para alimentar o gerenciador de registro. O formato padrão deles são registros separados por linhas, campos do registro separados por ponto-e-vírgula. Exemplos podem ser encontrados no diretório *./res* do projeto.

Quando importado um catálogo, todos os registros contidos nele são inseridos no arquivo de registro em que se está trabalhando. Se o arquivo de registro já existir, os novos registros são adicionados ao arquivo existente. Caso contrário, um novo arquivo de registro é criado com os registros presentes no catálogo.

Remoção no Arquivo de Índice

Ao se remover um registro do arquivo principal, o arquivo de índices não remove de sua página o item correspondente ao registro removido. Ao invés disso, ele altera o *offset* desse elemento para 0, indicando que o registro não existe mais no arquivo principal, apesar de ter referência em uma das páginas da árvore.

Referências

Folk, M. J.; Zoellick, B.; (1992). *File Structures*. Addison-Wesley, 2nd edition.