

# 6897 - Organização e Recuperação de Dados: Primeiro Trabalho

Chen Po Hsiang<sup>1</sup>, Rafael Sanchez<sup>1</sup>

<sup>1</sup>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).*

**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.*

## 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.

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

## Linux

Para compilar no Linux, abra o terminal no diretório 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 são semelhantes às usadas para compilar no Linux. Através de uma IDE, é necessário criar um novo projeto com os arquivos *main.c*, *arq\_reg.c* e *arq\_reg.h*. Nenhuma biblioteca adicional é necessária.

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 não seja especificado, o padrão usado na execução é "./res/registro.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.

Esse 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 caractere `'|'`.

Para os espaços disponíveis, os 2 bytes de tamanho são seguidos por um byte de caractere `'*'`, seguido por dois bytes com um *uint16\_t*, o qual sinaliza o *offset* do próximo espaço disponível da lista.

### 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.

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.

## Referências

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