

Universidade de São Paulo – USP
Instituto de Ciências Matemáticas e de Computação – ICMC
Departamento de Ciências de Computação – SCC

<http://coteia.icmc.usp.br/coteia/mostra.php?ident=572>

SCE-203 - Algoritmos e Estruturas de Dados II

Responsável: Prof. Gustavo Batista
gbatista@icmc.usp.br

Estagiário PAE: Victor Laguna
vlaguna@icmc.usp.br

**Aula de Laboratório II – Arquivos Organizados em
Campos e Registros**

É comum organizar arquivos de dados utilizando campos e registros. Embora se utilize com frequência uma estrutura com campos e registros de tamanho fixo, isso não é obrigatório. Conforme discutido em aula, existem diversas formas de organizar arquivos de campos e registros.

A aula de hoje consiste em fazer alguns programas que sejam capazes de ler arquivos organizados em campos e registros. Os arquivos necessários para a aula se encontram na coteia. Existem três arquivos:

- **fixo.dad** é um arquivo com campos e registros de tamanho fixo;
- **campo_var_reg_fixo.dad** é um arquivo com campos de tamanho variável e registros de tamanho fixo;
- **campo_var_reg_var.dad** é um arquivo com campos e registros de tamanho variável.

Todos os arquivos possuem a mesma informação: dez registros com os seguintes campos:

1. Um nome com no máximo 15 caracteres incluindo o caractere terminador de string;
2. Um sobrenome com no máximo 15 caracteres incluindo o caractere terminador de string;
3. Um nome de rua com no máximo 25 caracteres incluindo o caractere terminador de string;
4. Um número (da casa) que é um inteiro com 4 bytes (**int**).

O objetivo é fazer um programa para cada organização de arquivo descrita acima. Esse programa deve inicialmente imprimir os registros contidos no arquivo e logo em seguida pedir ao usuário um número entre 1 e 10 e imprimir o registro correspondente.

Como material de apoio, utilize os slides da aula 1 de arquivos disponível na coteia. Nessa aula pode-se encontrar as principais funções e procedimentos para manipulação de arquivos em C e Pascal.

Tarefa 1: Arquivo fixo.dad

O arquivo **fixo.dad** possui 10 registros gravados em um formato de campos e registros de tamanho fixo. Essa organização é uma das mais simples. Para a leitura de todos os dados tem-se duas opções principais:

1. Pode-se fazer uma chamada a **fread** para cada registro;
2. Pode-se definir um vetor de registros e ler todos os 10 registros com uma única chamada a **fread**. Para isso, ajuste o terceiro parâmetro do **fread**.

Em ambas as opções, utilize **sizeof** para calcular o tamanho em bytes ocupado por um registro. Por exemplo, **sizeof(reg)** retorna o número de bytes ocupados pelo registro **reg**.

Para a leitura de um registro pedido pelo usuário deve-se utilizar a função **fseek**. Para tanto, deve-se calcular o número de bytes existentes entre o início do arquivo e o registro de interesse. Novamente, **sizeof** é útil para descobrir quantos bytes ocupa cada registro.

A figura a seguir mostra a saída do utilitário hexdump do Unix para parte do arquivo de campos e registros fixos. Repare na quantidade de espaço não utilizado e na codificação dos números inteiros em binário.

```
0000: 4A 6F 61 6F 00 00 00 00 00 00 00 00 00 00 53  Joao.....S
0010: 6F 75 7A 61 00 00 00 00 00 00 00 00 00 00 58 56 ouza.....XV
0020: 20 4E 6F 76 65 6D 62 72 6F 00 00 00 00 00 00 00 Novembro.....
0030: 00 00 00 00 00 00 00 00 97 00 00 00 45 64 75 61 .....Edua
0040: 72 64 6F 00 00 00 00 00 00 00 00 4D 61 72 74 69 rdo.....Marti
0050: 6E 73 00 00 00 00 00 00 00 00 53 61 6F 20 53 65 ns.....Sao Se
0060: 62 61 73 74 69 61 6F 00 00 00 00 00 00 00 00 00 bastiao.....
0070: 00 00 00 00 C9 00 00 00 4D 61 72 69 61 00 00 00 .....Maria...
0080: 00 00 00 00 00 00 00 43 61 6C 64 61 73 00 00 00 .....Caldas...
0090: 00 00 00 00 00 00 41 71 75 69 64 61 62 61 6D 00 .....Aquidabam.
00A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00B0: 2D 00 00 00 4A 61 6E 61 69 6E 61 00 00 00 00 00 -...Janaina.....
00C0: 00 00 00 53 6F 75 74 6F 00 00 00 00 00 00 00 00 ...Souto.....
00D0: 00 00 4E 61 72 63 69 73 6F 20 52 6F 6D 61 6E 6F ..Narciso Romano
00E0: 00 00 00 00 00 00 00 00 00 00 00 00 BF 15 00 00 .....
00F0: 45 6C 69 73 61 00 00 00 00 00 00 00 00 00 4D Elisa.....M
```

Tarefa 2: Arquivo campo_var_reg_fixo.dad

O arquivo **campo_var_reg_fixo.dad** possui 10 registros gravados em um formato de campos de tamanho variável e registros de tamanho fixo. Foi utilizado o caractere '|' como separador de campos. Cada registro possui um tamanho fixo igual à soma dos tamanhos máximos de cada um dos campos (da mesma maneira que na Tarefa 1). Um exemplo de registro nesta organização é:

```
João|Moreira|XV de Novembro|110|<== espaço não utilizado ==>Eduardo|...
```

Para a leitura de todos os registros, pode-se ler blocos de dados cujo tamanho é igual a um registro. Entretanto, é necessário criar uma sub-rotina capaz de identificar cada um dos campos separado pelo caractere '|'.

Para a leitura de um registro especificado pelo usuário, pode-se calcular o número de bytes existentes entre o início do arquivo e o registro de interesse, da mesma maneira que na Tarefa 1. Mas a sub-rotina para identificar os campos também é necessária.

A figura a seguir mostra a saída do utilitário hexdump do Unix para parte do arquivo de campos de tamanho variável e registros de tamanho fixos. Repare na quantidade de espaço não utilizado no final de cada registro (espaço ocupado por lixo). Repare também na codificação dos números inteiros em ASCII.

```
0000: 4A 6F 61 6F 7C 53 6F 75 7A 61 7C 58 56 20 4E 6F  Joao|Souza|XV No
0010: 76 65 6D 62 72 6F 7C 31 35 31 7C 00 58 FA 22 00  vembro|151|.X.".
0020: 00 30 FD 7F F0 FC 22 00 F2 8B 16 77 2E AE 1A 77  .0...."....w...w
0030: FE FF FF FF 37 ED 18 77 41 4C 19 77 45 64 75 61  ....7..wAL.wEdua
0040: 72 64 6F 7C 4D 61 72 74 69 6E 73 7C 53 61 6F 20  rdo|Martins|Sao
0050: 53 65 62 61 73 74 69 61 6F 7C 32 30 31 7C 00 7F  Sebastiao|201|..
0060: F0 FC 22 00 F2 8B 16 77 2E AE 1A 77 FE FF FF FF  .."....w...w....
0070: 37 ED 18 77 41 4C 19 77 4D 61 72 69 61 7C 43 61  7..wAL.wMaria|Ca
0080: 6C 64 61 73 7C 41 71 75 69 64 61 62 61 6D 7C 34  ldas|Aquidabam|4
0090: 35 7C 00 61 6F 7C 32 30 31 7C 00 7F F0 FC 22 00  5|.ao|201|....".
00A0: F2 8B 16 77 2E AE 1A 77 FE FF FF FF 37 ED 18 77  ...w...w....7..w
00B0: 41 4C 19 77 4A 61 6E 61 69 6E 61 7C 53 6F 75 74  AL.wJanaina|Sout
00C0: 6F 7C 4E 61 72 63 69 73 6F 20 52 6F 6D 61 6E 6F  o|Narciso Romano
00D0: 7C 35 35 36 37 7C 00 7F F0 FC 22 00 F2 8B 16 77  |5567|...."....w
00E0: 2E AE 1A 77 FE FF FF FF 37 ED 18 77 41 4C 19 77  ...w....7..wAL.w
00F0: 45 6C 69 73 61 7C 4D 69 6C 61 72 65 7C 43 6F 72  Elisa|Milare|Cor
```

Tarefa 3: Arquivo campo_var_reg_var.dad

O arquivo **campo_var_reg_var.dad** possui 10 registros gravados em um formato de campos e registros de tamanho variável. Foi utilizado o caractere '|' como separador de campos. Cada registro possui um tamanho variável igual à soma dos tamanhos de cada um dos campos (diferente do utilizado nas Tarefas 1 e 2). Para auxiliar na leitura dos registros, cada registro inicia com um inteiro de 1 byte (**char**) que indica o tamanho do registro. Um exemplo de registro nesta organização é:

```
28João|Moreira|XV de Novembro|110|33Eduardo|...
```

No qual 28 e 33 são os tamanhos do primeiro e segundo registros, respectivamente. Não esqueça que esses dois valores estão armazenados em um único byte.

Para a leitura de todos os registros, pode-se ler um único byte e logo em seguida um bloco de dados cujo tamanho é igual ao número indicado no byte. Logo em seguida deve-se utilizar uma sub-rotina idêntica à da Tarefa 2 para identificar os campos.

Para a leitura de um registro especificado pelo usuário, é necessário ler os bytes indicativos de tamanho e logo em seguida usar a função **fseek** para saltar para o próximo registro. Isso deve ser repetido até localizar o registro de interesse.

A figura a seguir mostra a saída do utilitário hexdump do Unix para parte do arquivo de campos e registros de tamanho variável. Repare que não há espaço não utilizado no final de cada registro ou entre os campos. Repare também que no início de cada registro existe um byte indicativo do tamanho do registro mostrado em hexadecimal. Por exemplo, 1Bh = 27d e 22h = 34d são os tamanhos dos dois primeiros registros.

```

0000: 1B 4A 6F 61 6F 7C 53 6F 75 7A 61 7C 58 56 20 4E .Joao|Souza|XV N
0010: 6F 76 65 6D 62 72 6F 7C 31 35 31 7C 22 45 64 75 ovembro|151|"Edu
0020: 61 72 64 6F 7C 4D 61 72 74 69 6E 73 7C 53 61 6F ardo|Martins|Sao
0030: 20 53 65 62 61 73 74 69 61 6F 7C 32 30 31 7C 1A Sebastiao|201|.
0040: 4D 61 72 69 61 7C 43 61 6C 64 61 73 7C 41 71 75 Maria|Caldas|Aqu
0050: 69 64 61 62 61 6D 7C 34 35 7C 22 4A 61 6E 61 69 idabam|45|"Janai
0060: 6E 61 7C 53 6F 75 74 6F 7C 4E 61 72 63 69 73 6F na|Souto|Narciso
0070: 20 52 6F 6D 61 6E 6F 7C 35 35 36 37 7C 21 45 6C Romano|5567|!El
0080: 69 73 61 7C 4D 69 6C 61 72 65 7C 43 6F 72 6F 6E isa|Milare|Coron
0090: 61 6C 20 53 70 69 6E 6F 6C 61 7C 34 32 30 7C 1F al Spinola|420|.
00A0: 43 6C 61 75 64 69 61 7C 46 65 72 72 61 7A 7C 32 Claudia|Ferraz|2
00B0: 35 20 64 65 20 6D 61 72 63 6F 7C 33 31 31 7C 25 5 de marco|311|&
00C0: 44 69 6F 67 6F 7C 4F 6C 69 76 65 69 72 61 7C 53 Diogo|Oliveira|S
00D0: 61 6E 74 69 61 67 6F 20 53 61 6E 63 68 65 73 7C antiago Sanches|
00E0: 31 30 30 30 7C 1F 52 61 66 61 65 6C 7C 53 6F 61 1000|.Rafael|Soa
00F0: 72 65 73 7C 53 67 74 2E 20 4D 65 6E 65 7A 65 73 res|Sgt. Menezes

```

Responda as seguintes perguntas:

1. Existe uma relação de perda/ganho com cada uma das organizações utilizadas? Qual é o ganho de cada uma? Qual é a perda?
2. No arquivo de registros de tamanho fixo os inteiros são codificados em binário com 4 bytes. Por exemplo, o número da casa do João Souza ficou como 97 00 00 00. Isso está correto? Não deveria ser 00 00 00 97? Qual a relação disso com as representações little-endian e big-endian?
3. Com campos de tamanho variável, foi utilizada a codificação em ASCII dos números. Por quê? Isso ajuda de alguma forma a prevenir que o caractere separador de campos '|' apareça de forma indesejada?