

# Campos e Registros

6897/9895 – Organização e Recuperação de Dados

Profa. Valéria D. Feltrim

UEM – CTC – DIN

Slides preparados com base no Cap. 4 do livro FOLK, M.J. & ZOELLICK, B. *File Structures*. 2<sup>nd</sup> Edition, Addison-Wesley Publishing Company, 1992, e nos slides disponibilizados pelo Prof. Pedro de Azevedo Berger (DCC/UnB)

# Organização em campos e registros

- Suponha que o arquivo que estamos construindo contém nomes e endereços de pessoas e que um programa aceita os nomes e os endereços via teclado e os escreve em um arquivo como um fluxo contínuo de bytes

## Pseudocódigo

```
PROGRAM: writstrm
get output file name and open it with the logical name OUTPUT
get LAST name as input
while length(LAST) > 0 do
    get FIRST name, ADDRESS, CITY, STATE and ZIP as input
    write LAST to the file OUTPUT
    write FIRST to the file OUTPUT
    write ADDRESS to the file OUTPUT
    write CITY to the file OUTPUT
    write STATE to the file OUTPUT
    write ZIP to the file OUTPUT
    get LAST name as input
end /* while */
close OUTPUT
end PROGRAM
```

# Organização em campos e registros

- Supondo que os seguintes nomes e endereços são utilizados como entradas do programa:

Alan Silva,  
Rua Tiete 123,  
Maringa, PR 87100

Andre Flores,  
Rua Braga 34,  
Sarandi, PR 87111

## Saída do programa

SilvaAlanRua Tiete 123MaringaPR87100FloresAndreRua Braga 34SarandiPR87111

- Perdemos a integridade das unidades de organização das entradas de dados!
- Essas unidades não são caracteres individuais, mas **agregados** com significado, como: [Alan Silva] ou [Rua Braga 34]
- Chamamos esses “agregados com significado” de **campos** e eles são as menores unidades de formação com significado dentro de um arquivo

# Estruturas de campos

- Existem várias formas de adicionar estrutura aos arquivos para manter a integridade dos campos

- Método 1: Definir campos de tamanhos fixos
- Método 2: Iniciar cada campo com um indicador de tamanho
- Método 3: Separar os campos por um delimitador
- Método 4: Usar uma expressão do tipo *palavra-chave = valor* para identificar cada campo e seu conteúdo
  - Estrutura auto-descritiva

Tamanho fixo

Tamanho variável

# Método 1 - Campos com tamanho fixo

Silva	Alan	Rua Tiete 123	Maringa	PR	87100
Flores	Andre	Rua Braga 34	Sarandi	PR	87111
...	...	...	...	...	...

- Força os dados em campos de tamanho fixo
- 🚫 Aumento do tamanho do arquivo: pode alocar mais espaço do que o requerido para armazenar valores menores do que o tamanho de cada campo
- 🚫 Perda da qualidade da informação → pelo limite imposto, valores de campos podem ser truncados (p.e., nomes ou endereços)
- 👍 Indicado quando o tamanho dos valores a serem armazenados for fixo ou com pouca variação
- 👍 **Facilita a implementação e a recuperação dos registros**

## Método 2 - Campos com indicador de tamanho

```
05Silva04Alan13Rua Tiete 12307Maringa02PR0587100
06Flores05Andre12Rua Braga 3407Sarandi02PR0587111
...
```

- O tamanho de cada campo é armazenado imediatamente antes do valor do campo
  - Sabe-se que sempre haverá um ou mais bytes reservados para armazenar o tamanho do campo
    - Para campos com até 256 bytes de tamanho um byte é suficiente
  - Tamanho zero indica valor de campo ausente


## Método 3 - Campos com delimitador

```
Silva|Alan|Rua Tiete 123|Maringa|PR|87100  
Flores|Andre|Rua Braga 34|Sarandi|PR|87111  
...
```

- Preserva a identidade de cada campo separando-os com delimitadores
- Escolhe-se algum caracter especial (ou uma sequência de caracteres) que não pode aparecer como valores dos campos para ser o delimitador
  - No exemplo acima, a barra vertical (“pipe”) → |

## Método 4 - Campo com *palavra-chave= valor*

```
<last>Silva </last> <first>Alan </first><address>Rua Tiete  
123</address><city>Maringa</city><state>PR</state><zip>87100</zip>  
<last>Flores </last><first>Andre</first><address>Rua Braga  
34</address><city>Sarandi</city><state>PR</state><zip>87111</zip>...
```

- O campo fornece informação sobre si próprio (**metadado**)
- Fica fácil identificar o conteúdo do arquivo
- É um bom formato para lidar com campos ausentes: a *palavra-chave* simplesmente está ausente
- Normalmente é utilizado **em conjunto com delimitadores**
-  As *palavras-chaves* podem ocupar uma porção significativa do arquivo
  - Neste exemplo, mais da metade do espaço é gasto com palavras-chaves e delimitadores



# Estruturas de registros

- Apenas a noção lógica de **campo** pode não ser suficiente para trabalhar com arquivos
  - No exemplo visto até agora, como diferenciar os dados de uma pessoa de outra?
- Precisamos de mais uma abstração → a noção de **Registro**
- Um **registro** é um conjunto de campos (de quaisquer tipos) que formam uma unidade de significado em um nível mais alto de abstração dentro do arquivo
- É uma ferramenta conceitual (e não uma entidade física), assim como o conceito de campo
- É um outro nível de organização que se impõe aos dados para preservar seu significado

# Estruturas de registros

- Os métodos de organização por registro mais frequentes são:

- Método 1: Registros de tamanho fixo
- Método 2: Registros com número fixo de campos
- Método 3: Registros com indicação de tamanho
- Método 4: Registros apontados por índices
- Método 5: Registros com delimitadores

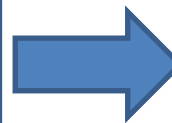
Tamanho fixo

Tamanho variável

# Método 1 - Registros de tamanho fixo

- O registro tem **tamanho fixo, em bytes**
- São as estruturas mais simples para organização de arquivos
- Todos os registros do arquivo ocupam o mesmo número de bytes
  - Análogo ao conceito de campos de tamanho fixo
- O número fixo de bytes em um registro não implica em número fixo de bytes para os campos do registro
  - Podemos ter campos de tamanho fixo ou variável

Exemplo de registro de tamanho fixo  
com campos de tamanho fixo



```
struct {  
    char last[10];  
    char first[10];  
    char address[15];  
    char city[15];  
    char state[3];  
    char zip[9];  
} set_of_fields;
```

# Método 1 - Registros de tamanho fixo

Registros de tamanho fixo com campos de tamanho fixo					
Silva	Alan	Rua Tiete 123	Maringa	PR	87100
Flores	Andre	Rua Braga 34	Sarandi	PR	87111

Registros de tamanho fixo com campos de tamanho variável					
Silva	Alan	Rua Tiete 123	Maringa	PR	87100
Flores	Andre	Rua Braga 34	Sarandi	PR	87111

- No segundo caso, a soma do tamanho dos campos e do tamanho dos elementos de controle ainda tem como limite o tamanho fixado pelo registro (o registro é visto como um “*container*”)
- **Fragmentação interna** em nível de registro
  - Suponha um registro com tamanho de 62B contendo os campos: nome (20B), endereço (20B), cidade (20B), UF (2B)
  - Se os dados ocuparem 32 caracteres (32B), mesmo assim o registro ocupará 62B

## Método 2 - Registros com número fixo de campos

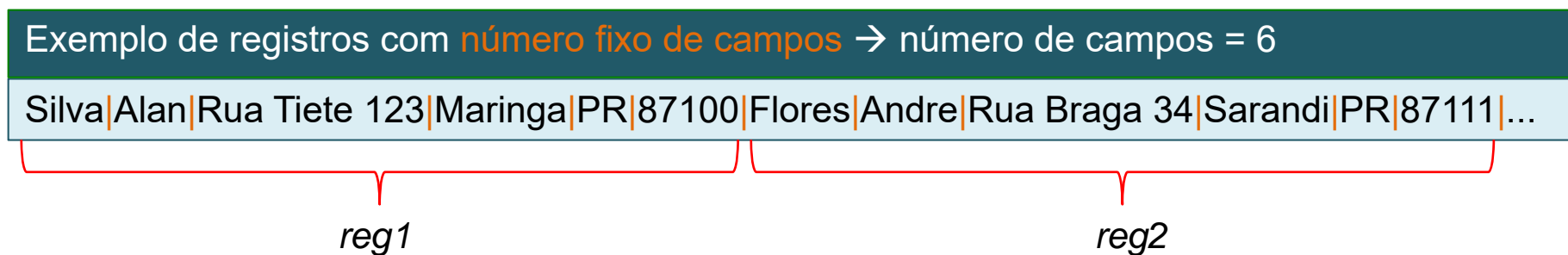
- Quantidade fixa de campos

- Em vez de se especificar que cada registro no arquivo contém um número fixo de bytes, podemos especificar um número fixo de campos

- Cada campo pode ter tamanho fixo ou variável

- O tamanho do registro, em bytes, é variável

- É o somatório dos tamanhos dos campos (mais seus separadores, se for o caso)



- Não temos mais o problema da fragmentação interna

## Método 3 - Registros com indicador de tamanho

- O registro começa com um indicador do seu **tamanho em bytes**
  - Por ex., podemos determinar que os primeiros 2 bytes de todo registro armazena o seu tamanho
- Método frequentemente empregado para tratar registros de tamanho variável
- É preciso conhecer o tamanho em bytes do espaço ocupado pelos campos e delimitadores dos campos antes da gravação do registro
  - Uso de um *buffer* para armazenar os dados e calcular o tamanho do registro em bytes antes das informações serem enviadas para o disco

Exemplo de registros com **indicador de tamanho no início**

**42**Silva|Alan|Rua Tiete 123|Maringa|PR|87100|**43**Flores|Andre|Rua Braga 34|Sarandi|...

2 bytes

42 bytes

## Método 4 - Registros apontados por índices

- Uso de um **arquivo adicional de índices** que aponta para os registros no arquivo de dados
  - Cada entrada do índice mantém o endereço de deslocamento (*offset*) para cada registro no arquivo original
  - A diferença dos endereços entre índices consecutivos representa o tamanho do registro
  - Os registros no arquivo de dados e seus endereços no arquivo de índices são consecutivos e correspondentes

Exemplo com delimitadores de campos no arquivo de dados

**Arquivo de dados:**

Silva|Alan|Rua Tiete 123|Maringa|PR|87100|Flores|Andre|Rua Braga 34|Sarandi ...

**Arquivo de índices:**

00 42 ...



## Método 5 - Registros com delimitadores

- Mesma ideia usada na delimitação de campos
- Assim como ocorre com o delimitador de campos, o delimitador de registro não pode fazer parte do alfabeto usado nos valores dos campos
- Uma opção comum de delimitador de registro para arquivos texto é o caracter de fim de linha ('\n')
- Os delimitadores de campos permanecem inalterados

Exemplo com delimitadores de campos (|) e delimitador de registro (#)

Silva|Alan|Rua Tiete 123|Maringa|PR|87100#Flores|Andre|Rua Braga 34 ...



# Concluindo...

- Há **várias combinações possíveis** entre os métodos de organização de campos e de registros
- Nenhum dos métodos descritos para organizar um arquivo em registros é apropriado para todas as situações
- A escolha de um método ou outro depende da **natureza dos dados** e **de como esses dados serão utilizados/processados**