

Campos e Registros

Organização e Recuperação de Dados

Profa. Valéria

UEM – CTC – DIN

Organização em campos e registros

- Queremos criar um programa que permita cadastrar pessoas
 - Para cada pessoa, armazenaremos as seguintes informações:
 - **Sobrenome, Nome, Endereço, Cidade, Estado e CEP**
 - Por exemplo:

Alan Silva,
Rua Tiete 123,
Maringa, PR, 87100

Andre Flores,
Rua Braga 34,
Sarandi, PR, 87111

- Cada uma dessas informações constitui um **agregado** com um significado próprio, como: “Alan” ou “Rua Braga 34”
- Chamamos esses agregados de **campos** e eles são as menores unidades de formação com significado dentro de um arquivo

Organização em campos e registros

- Considere o pseudocódigo abaixo para criar o nosso cadastro de pessoas como um fluxo de bytes

Pseudocódigo

```
PROGRAMA: escreve_fluxo
receba o nome do arquivo a ser criado e abra o arquivo com o nome lógico SAIDA
receba o SOBRENOME
enquanto comprimento(SOBRENOME) > 0 faça
    receba NOME, ENDEREÇO, CIDADE, ESTADO e CEP
    escreva SOBRENOME no arquivo SAIDA
    escreva NOME no arquivo SAIDA
    escreva ENDEREÇO no arquivo SAIDA
    escreva CIDADE no arquivo SAIDA
    escreva ESTADO no arquivo SAIDA
    escreva CEP no arquivo SAIDA
    receba o SOBRENOME
fim /* enquanto */
feche SAIDA
fim PROGRAMA
```

Organização em campos e registros

- Supondo que cadastramos os dados de duas pessoas usando o nosso programa:

Alan Silva,
Rua Tiete 123,
Maringa, PR 87100

Andre Flores,
Rua Braga 34,
Sarandi, PR 87111

Arquivo gerado:

SilvaAlanRua Tiete 123MaringaPR87100FloresAndreRua Braga 34SarandiPR87111

- Perdemos a integridade das unidades de informação das entradas de dados
- ***Não conseguiremos mais separar um campo de outro***

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

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 |
| ... | ... | ... | ... | .. | ... |
| 8 | 8 | 15 | 12 | 2 | 5 |

- Força os dados em campos de tamanho fixo
- 📢 Aumento do tamanho do arquivo: normalmente se aloca mais espaço do que o necessário
- 📢 Possível perda de informação → pelo limite imposto ao tamanho dos campos, os valores dos campos podem ser truncados
- 👍 Indicado quando o tamanho dos valores a serem armazenados for fixo ou com pouca variação
- 👍 **Facilita a implementação em linguagens que utilizam tipos com tamanhos pré-definidos**

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 uma quantidade pré-definida de bytes reservada para armazenar o tamanho do campo
 - O campo 'tamanho do campo' tem tamanho fixo, p.e., 1 byte
 - Para campos com até 255 bytes de tamanho um byte é suficiente
 - Tamanho igual a 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-se a identidade de cada campo separando-os por meio de um delimitador
- O delimitador deve ser um caractere especial (ou uma sequência de caracteres especiais) que não aparece nos valores dos campos
 - No exemplo acima, a barra vertical (ou “*pipe*”) está sendo utilizada como delimitador → ‘|’

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

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

- Cada campo é delimitado e tem uma palavra-chave que o descreve – metadados
- Facilita a identificação do conteúdo do arquivo
- Facilita o tratamento de campos ausentes
- 📢 As *palavras-chaves* podem ocupar uma porção significativa do arquivo
 - Neste exemplo, mais de 50% do espaço é gasto com palavras-chaves e delimitadores

Arquivo de registros

Prática 1: Programas 1 e 2

- Ver slides da atividade prática

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 necessariamente uma entidade física), assim como os campos
- É um nível adicional 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

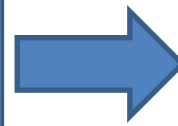
Tamanho fixo

Tamanho variável

Método 1 - Registros de tamanho fixo

- O registro tem um **tamanho fixo em bytes**
 - Todos os registros do arquivo ocuparam 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
- Facilmente implementado em linguagens com tipos de tamanho fixo pré-definido

Exemplo de registro de tamanho fixo com campos de tamanho fixo usando uma struct da linguagem C



```
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 Joa 12 | Maringa | PR | 87100 |
| Flores | Adriana | Rua Braga 34 | Sarandi | PR | 87111 |

| Registros de tamanho fixo com campos de tamanho variável | | | | | | | |
|--|---------|--------------|---------|----|-------|----------------------|--|
| Silva | Alan | Rua Joa 12 | Maringa | PR | 87100 | \0\0\0\0\0\0\0\0\0\0 | |
| Flores | Adriana | Rua Braga 34 | Sarandi | PR | 87111 | \0\0\0\0\0\0\0\0\0\0 | |

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

Método 2 - Registros com número fixo de campos

- O registro é composto por uma **quantidade fixa de campos**
 - Em vez de se especificar que cada registro no arquivo contém um número fixo de bytes, fixamos o número de campos
- Cada campo pode ter tamanho fixo ou variável
- O tamanho do registro, em bytes, é variável
 - Corresponde ao somatório dos tamanhos dos campos (mais seus separadores, se for o caso)
 - Não há fragmentação interna

Exemplo de registros com **número fixo de campos** → número de campos = 6

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

reg1

reg2

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 armazenam o seu tamanho → o tamanho é um campo de tamanho fixo
- 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 arquivo

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

42Silva | 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 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 registros é o caractere de fim de linha ('\n')
 - Em Python, é possível iterar em um arquivo por linha!
- Os delimitadores de campos permanecem inalterados

Exemplo com delimitadores de campos (|) e delimitador de registro (\n)

Silva|Alan|Rua Tiete 123|Maringa|PR|87100|\nFlores|Andre|Rua Braga 34 ... \n

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 o melhor em todas as situações
 - A escolha de um método ou outro depende da **natureza dos dados e de como esses dados serão utilizados/processados**

Arquivo de registros

Prática 1: Programas 3 e 4

- Ver slides da atividade prática