

ACH2024

Aula 13

Organização interna de arquivos

Profa. Ariane Machado Lima

Objetivo e Programa

- **Objetivo:**

Capacitar o aluno a:

i) modelar e resolver problemas baseados em grafos;

ii) compreender as estruturas de dados para memória secundária, comparando-as com as alternativas para memória principal.

- **Programa:**

i) Estruturas de dados para representação de grafos e seus algoritmos (buscas em largura e em profundidade, caminhos mais curtos de única origem, árvore geradora mínima).

ii) Conceitos básicos de organização e acesso a arquivos. Árvores-B e hashing em memória principal e secundária. Algoritmos de ordenação externa.

- **Programação em C** – quantos sabem?



Segunda parte da disciplina

- Organização interna de arquivos
- Tipos de alocação de arquivos:
 - Sequencial
 - Ligada
 - Indexada
 - Árvores-B
 - Hashing (veremos também hashing em memória principal)
- Algoritmos de processamento cossequencial e ordenação em disco

Segunda parte da disciplina

- **Organização interna de arquivos**
- Tipos de alocação de arquivos:
 - Sequencial
 - Ligada
 - Indexada
 - Árvores-B
 - Hashing (veremos também hashing em memória principal)
- Algoritmos de processamento cossequencial e ordenação em disco

Arquivos

- O que é um arquivo?

Arquivos

- O que é um arquivo?
 - Físico:
 - Uma sequência de bytes armazenados em algum lugar (disco, SSD, pen-drive, fita, ...)
 - Obs: sim! Fitas ainda são muito usadas para backups de grandes volumes de dados!
 - Pode ser visto como um espaço de endereçamento (em disco, SSD, etc...)
 - Lógico: fluxo de bytes (stream) – ex: pode-se ler do teclado, escrever na tela, etc...

Arquivos

- Quais alguns tipos de arquivos?

Arquivos

- Quais alguns tipos de arquivos?
 - Texto (.txt, .doc, .pdf, ...)
 - Imagem (.tiff, .jpeg, .gif, .pdf, ...)
 - Planilha (.xls, .csv, .txt, ...)
 - Executáveis
 - ...

Arquivos

- Quais alguns tipos de arquivos?
 - Texto (.txt, .doc, .pdf, ...)
 - Imagem (.tiff, .jpeg, .gif, .pdf, ...)
 - Planilha (.xls, .csv, .txt, ...)
 - Executáveis
 - ...

O que define um tipo de arquivo?

Arquivos

- Quais alguns tipos de arquivos?
 - Texto (.txt, .doc, .pdf, ...)
 - Imagem (.tiff, .jpeg, .gif, .pdf, ...)
 - Planilha (.xls, .csv, .txt, ...)
 - Executáveis
 - ...

O que define um tipo de arquivo?

Seu conteúdo, a definição do seu formato

Ex: Arquivos .txt

- Texto simples (sequências de caracteres) estruturado em linhas
- Pode ser aberto (lido/editado) em qualquer programa editor de texto

Arquivo .txt

Olá

Como vai? Fiquei
sabendo que você
estava doente...
Melhoras!

Vários outros

<https://docs.fileformat.com/>

Search here... Q

- 3D File Formats
- Audio File Formats
- CAD File Formats
- Compression File Formats
- Database
- Disc and Media
- EBook File Formats
- Email File Formats
- Executable
- Finance
- Font
- Game File Formats
- GIS File Formats
- Image File Formats
- Misc File Formats
- Note Taking File Formats
- Page Description Language
- PDF
- Presentation File Formats
- Programming
- Project Management File Formats
- Publisher File Formats
- Reporting
- Settings File Formats
- Spreadsheet File Formats
- System File Formats
- Video File Formats
- Visio File Formats
- Web File Formats
- Word Processing File Formats

Home / Image File Formats / PNG

What's on this Page

- [What is a PNG file?](#)
- [Brief History of PNG File Format](#)
- [Functional Comparison of GIF and PNG](#)
- [PNG File Structure](#)
- [PNG File Header](#)
- [Chunks](#)
- [Compression](#)
- [References](#)

What is a PNG file? [↗](#)

A **PNG** (Portable Network Graphics) file is a raster image file format that uses lossless compression. This file image compression that makes it popular among its users. With the passage of time, PNG has evolved as o

Brief History of PNG File Format [↗](#)

The main reason behind the creation of PNG file format was the patented compression algorithm, Lempel-; events with respect to PNG file formats are listed below:

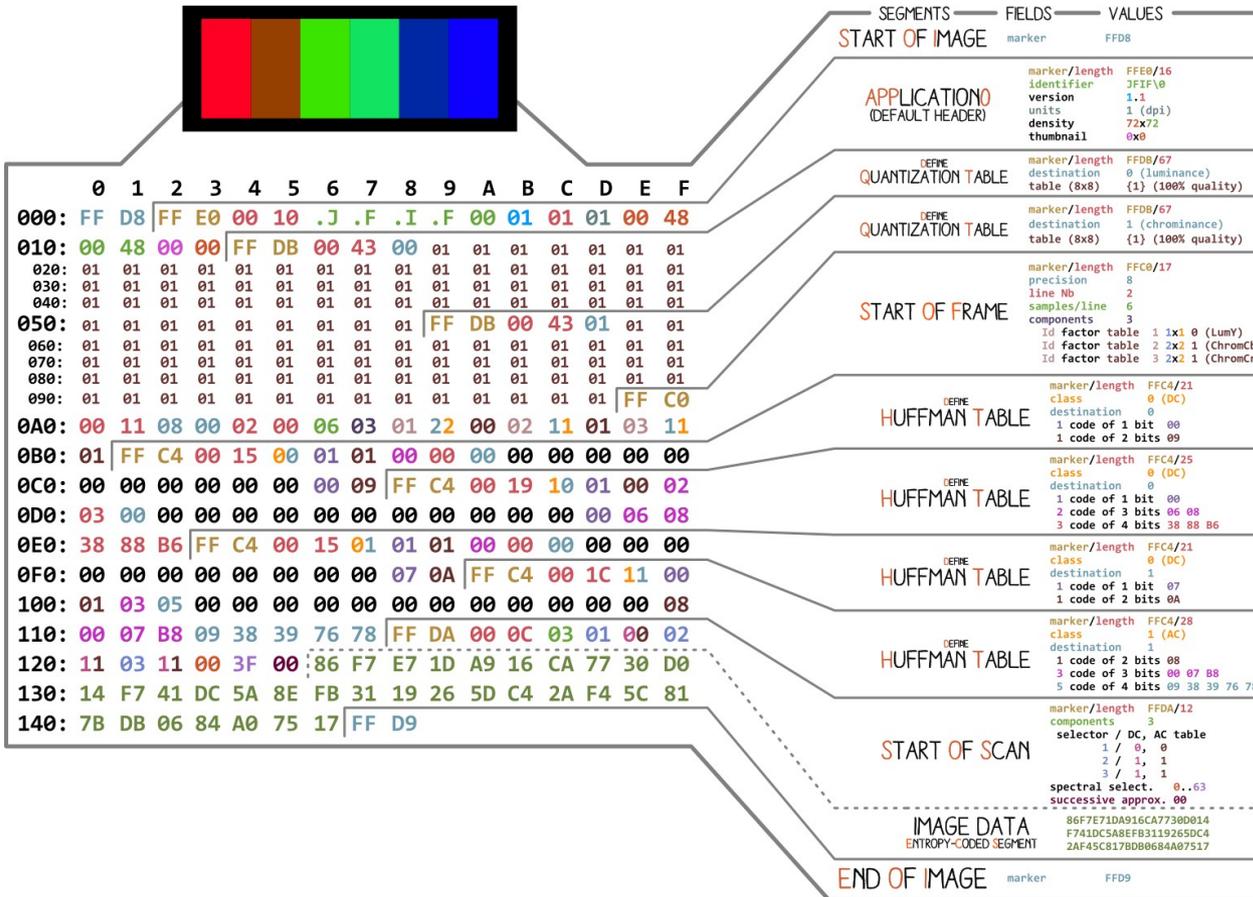
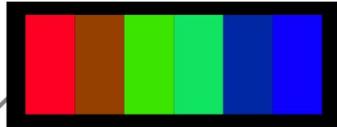
- October 1996: PNG specifications Version 1.0 were released and later appeared as [RFC 2083](#). The san
- December 1998: Version 1.1, with some small changes and the addition of three new chunks, was rel
- August 1999: Version 1.2, adding one extra chunk, was released.
- November 2003: PNG became an International Standard (ISO/IEC 15948:2003). This version of PNG di
- March 2004: ISO/IEC 15948:2004

Functional Comparison of GIF and PNG [↗](#)

PNG file format was designed to be simple and portable, legally unencumbered, interchangeable, flexible a

Feature
Index-color images of up to 256 colors
Support for streaming

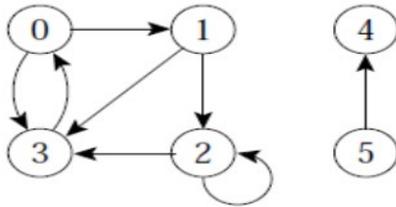
Arquivos .jpg



Muitos arquivos possuem dados e meta-dados (dados que descrevem os dados)

Ex: Serialização de estruturas de dados

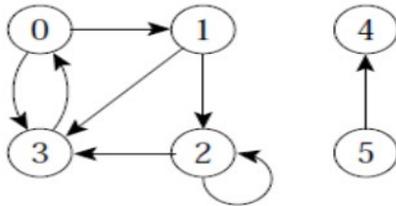
Ex: como serializo (transformo em uma sequência de caracteres) uma matriz de adjacências de um grafo em um arquivo?



	0	1	2	3	4	5
0		1		1		
1			1	1		
2			1	1		
3	1					
4						
5					1	

Ex: Serialização de estruturas de dados

Ex: como serializo (transformo em uma sequência de caracteres) uma matriz de adjacências de um grafo em um arquivo?



	0	1	2	3	4	5
0		1		1		
1			1	1		
2			1	1		
3	1					
4						
5					1	

Arquivo .txt

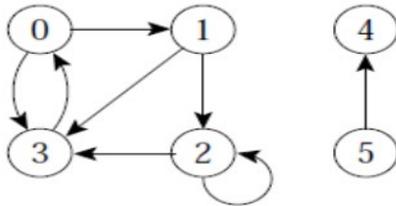
```
6
0 1 0 1 0 0
0 0 1 1 0 0
0 0 1 1 0 0
1 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 1 0
```

Acabo de definir um formato
Poderia definir outro para
listas de adjacências

Em C:
fprintf para serializar
fscanf para ler de volta

Ex: Serialização de estruturas de dados

Ex: como serializo (transformo em uma sequência de caracteres) uma matriz de adjacências de um grafo em um arquivo?



Pode-se também escrever em um arquivo binário (um int e v*v bits !!!)

Em C: fwrite para serializar fread para ler de volta

	0	1	2	3	4	5
0		1		1		
1			1	1		
2			1	1		
3	1					
4						
5					1	

<6 em binário ocupando um certo nr bytes>01010000110000110010000000000000010

Arquivos

- Quais são algumas características desejáveis de arquivos (em geral)?
Algumas das principais:
 - Persistência: conteúdo precisa ser mantido mesmo quando o computador é desligado, isto é, preciso ser armazenado em um dispositivo de memória não volátil (memória secundária), normalmente o disco
 - Concorrência: múltiplos processos devem poder acessar o mesmo arquivo simultaneamente
 - Capacidade de armazenamento: quero armazenar uma grande quantidade de informação, maior do que a capacidade da memória principal (às vezes até mesmo para um ÚNICO arquivo!)

Arquivos

- Quais são algumas características desejáveis de arquivos (em geral)?
Algumas das principais:
 - Persistência: conteúdo precisa ser mantido mesmo quando o computador é desligado, isto é, preciso ser armazenado em um dispositivo de memória não volátil (memória secundária), normalmente o disco
 - Concorrência: múltiplos processos devem poder acessar o mesmo arquivo simultaneamente
 - Capacidade de armazenamento: quero armazenar uma grande quantidade de informação, maior do que a capacidade da memória principal (às vezes até mesmo para um ÚNICO arquivo!)
 - Quais tipos de arquivos costumam ser grandes assim?

Arquivos

- Quais são algumas características desejáveis de arquivos (em geral)?
Algumas das principais:
 - Persistência: conteúdo precisa ser mantido mesmo quando o computador é desligado, isto é, preciso ser armazenado em um dispositivo de memória não volátil (memória secundária), normalmente o disco
 - Concorrência: múltiplos processos devem poder acessar o mesmo arquivo simultaneamente
 - Capacidade de armazenamento: quero armazenar uma grande quantidade de informação, maior do que a capacidade da memória principal (às vezes até mesmo para um ÚNICO arquivo!)
 - Quais tipos de arquivos costumam ser grandes assim? **Arquivos de DADOS, como planilhas (conjunto de registros – linhas, cada registro com seus campos).**

Arquivos

- Nas próximas aulas veremos como esses arquivos (conjunto de registros) são organizados no disco, pensando na viabilização de:
 - Leitura sequencial e aleatória
 - Alteração de registros
 - Inserção de registros
 - Exclusão de registros

Tudo isso lembrando que o arquivo tem que passar pela memória principal (ou seja, sua organização em disco tem que ser compatível com a estrutura da memória principal).
- Ligação direta com as disciplinas de Sistemas Operacionais e Banco de Dados

Estrutura interna e alocação de arquivos de dados

- Estrutura de serialização de uma estrutura de dados (ex: grafos)
- Muitos arquivos de dados são uma lista de dados, cada um podendo conter vários campos
 - Ex: linhas de uma tabela, seja de uma planilha, de um .csv, de uma tabela de banco de dados, ...
 - Cada dado normalmente tem uma chave, pela qual é possível realizar operações de busca e ordenação
- Chamaremos cada dado (contendo a chave e demais campos) de **registro**
- Como separar os campos? Como separar os registros?

Estrutura interna e alocação de arquivos de dados

- Estrutura de serialização de uma estrutura de dados (ex: grafos)
- Muitos arquivos de dados são uma **lista de dados, cada um podendo conter vários campos**
 - Ex: linhas de uma tabela, seja de uma planilha, de um .csv, de uma tabela de banco de dados, ...
 - Cada dado normalmente tem uma chave, pela qual é possível realizar operações de busca e ordenação
- Chamaremos cada dado (contendo a chave e demais campos) de **registro**
- Como separar os campos? Como separar os registros?

FOCO PRINCIPAL DO RESTANTE DA DISCIPLINA

Ex: Tabela
de vendas

No. Venda	Data	Valor da Venda	No. Itens	Vendedor
1	1/mar	R\$ 280,00	2	Alfredo
2	1/mar	R\$ 510,00	4	Marcos
3	1/mar	R\$ 370,00	3	João
4	2/mar	R\$ 205,00	3	João
5	2/mar	R\$ 190,00	2	João
6	2/mar	R\$ 720,00	5	Marcos
7	2/mar	R\$ 130,00	1	Marcos
8	3/mar	R\$ 300,00	3	João
9	3/mar	R\$ 250,00	2	Alfredo
10	4/mar	R\$ 390,00	3	Alfredo
11	4/mar	R\$ 275,00	2	João
12	4/mar	R\$ 110,00	1	João
13	6/mar	R\$ 640,00	3	Marcos
14	6/mar	R\$ 140,00	1	Alfredo
15	6/mar	R\$ 170,00	1	Alfredo

Como gravo/organizo esse arquivo em disco?

Ex: Tabela
de vendas

No. Venda	Data	Valor da Venda	No. Itens	Vendedor
1	1/mar	R\$ 280,00	2	Alfredo
2	1/mar	R\$ 510,00	4	Marcos
3	1/mar	R\$ 370,00	3	João
4	2/mar	R\$ 205,00	3	João
5	2/mar	R\$ 190,00	2	João
6	2/mar	R\$ 720,00	5	Marcos
7	2/mar	R\$ 130,00	1	Marcos
8	3/mar	R\$ 300,00	3	João
9	3/mar	R\$ 250,00	2	Alfredo
10	4/mar	R\$ 390,00	3	Alfredo
11	4/mar	R\$ 275,00	2	João
12	4/mar	R\$ 110,00	1	João
13	6/mar	R\$ 640,00	3	Marcos
14	6/mar	R\$ 140,00	1	Alfredo
15	6/mar	R\$ 170,00	1	Alfredo

Como gravo/organizo esse arquivo em disco?

Exemplos de possibilidades:

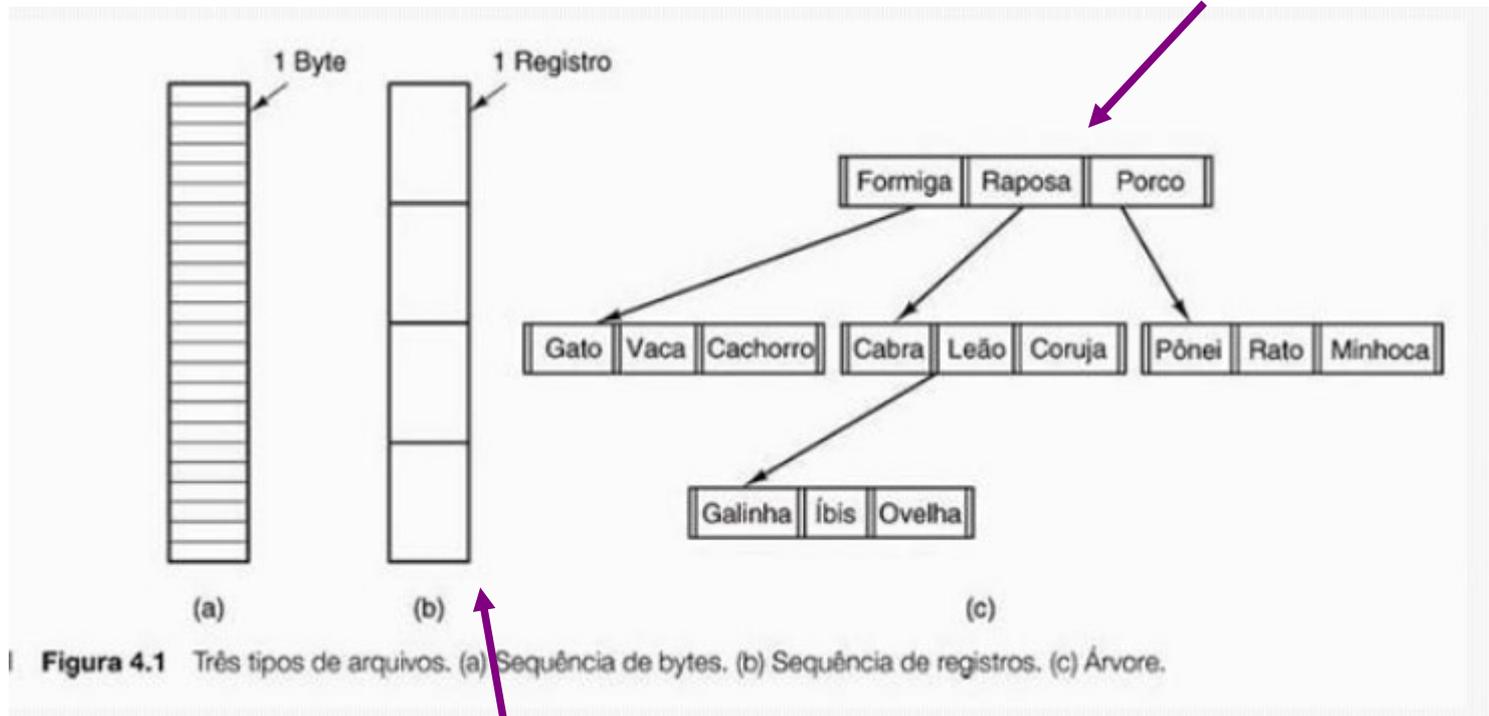


Figura 4.1 Três tipos de arquivos. (a) Sequência de bytes. (b) Sequência de registros. (c) Árvore.

Alocação sequencial

Organização INTERNA de arquivos (em memória secundária)

Em arquivos de dados, como organizar:

- os vários **campos** dentro de cada registro?
- os vários **registros** dentro do arquivo?

Organização INTERNA de arquivos (em memória secundária)

Em arquivos de dados, como organizar:

- os vários **campos** dentro de cada registro?
- os vários **registros** dentro do arquivo?



Métodos para organizar os campos dentro dos registros

- Comprimento fixo
- Indicador de comprimento
- Delimitadores
- Tags (etiquetas)

Campos com comprimento fixo

- Cada campo ocupa no arquivo um tamanho fixo (não necessariamente o mesmo para diferentes campos).

Ex: nome (12 chars), logradouro (10 chars), nr (10 chars), cidade (10 chars)

Maria	Rua 1	123	São Carlos
João	Rua A	255	Rio Claro
Pedro	Rua 10	56	Rib. Preto

- Se eu sei o tamanho fixo de cada campo, como uso essa informação para ler o arquivo?

Campos com comprimento fixo

- Cada campo ocupa no arquivo um tamanho fixo (não necessariamente o mesmo para diferentes campos).

Ex: nome (12 chars), logradouro (10 chars), nr (10 chars), cidade (10 chars)

Maria	Rua 1	123	São Carlos
João	Rua A	255	Rio Claro
Pedro	Rua 10	56	Rib. Preto

- Se eu sei o tamanho fixo de cada campo, como uso essa informação para ler o arquivo?
 - Leio exatamente aquele tanto (de chars, bytes, etc) para ler cada campo

Campos com comprimento fixo

- Cada campo ocupa no arquivo um tamanho fixo (não necessariamente o mesmo para diferentes campos).

Ex: nome (12 chars), logradouro (10 chars), nr (10 chars), cidade (10 chars)

Maria	Rua 1	123	São Carlos
João	Rua A	255	Rio Claro
Pedro	Rua 10	56	Rib. Preto

- E se eu quiser acessar apenas o terceiro campo da primeira linha?

Campos com comprimento fixo

- Cada campo ocupa no arquivo um tamanho fixo (não necessariamente o mesmo para diferentes campos).

Ex: nome (12 chars), logradouro (10 chars), nr (10 chars), cidade (10 chars)

Maria	Rua 1	123	São Carlos
João	Rua A	255	Rio Claro
Pedro	Rua 10	56	Rib. Preto

- E se eu quiser acessar apenas o terceiro campo da primeira linha?
 - Quero acessar os chars 23 a 32 (do arquivo)

Campos com comprimento fixo

- Cada campo ocupa no arquivo um tamanho fixo (não necessariamente o mesmo para diferentes campos).

Ex: nome (12 chars), logradouro (10 chars), nr (10 chars), cidade (10 chars)

Maria	Rua 1	123	São Carlos
João	Rua A	255	Rio Claro
Pedro	Rua 10	56	Rib. Preto

E se eu quiser acessar apenas o terceiro campo da i -ésima linha?

Campos com comprimento fixo

- Cada campo ocupa no arquivo um tamanho fixo (não necessariamente o mesmo para diferentes campos).

Ex: nome (12 chars), logradouro (10 chars), nr (10 chars), cidade (10 chars)

Maria	Rua 1	123	São Carlos
João	Rua A	255	Rio Claro
Pedro	Rua 10	56	Rib. Preto

E se eu quiser acessar apenas o terceiro campo da i -ésima linha?

Corresponde aos chars

$$(i-1) * \underline{(12+10+10+10)} + 23 \text{ até } (i-1) * \underline{(12+10+10+10)} + 32$$

Tamanho total de um registro inteiro

Campos com comprimento fixo

É como se o conjunto de registros fosse um vetor de estruturas, sendo que a estrutura define os campos.

```
struct {  
    char last[10];  
    char first[10];  
    char city[15];  
    char state[2];  
    char zip[9];  
} set_of_fields;
```

Campos com comprimento fixo

Quais as **vantagens** dessa abordagem?

Campos com comprimento fixo

Quais as **vantagens** dessa abordagem?

A leitura sequencial é fácil (desserialização total)

Campos com comprimento fixo

Quais as **desvantagens** dessa abordagem?

Campos com comprimento fixo

Quais as **desvantagens** dessa abordagem?

- O espaço alocado (e não usado) aumenta desnecessariamente o tamanho do arquivo (**desperdício**)
 - Solução inapropriada quando se tem uma grande quantidade de dados com tamanho variável
 - Razoável apenas se o comprimento dos campos é realmente fixo ou apresenta pouca variação

Campos com indicador de comprimento

- O **tamanho de cada campo** é armazenado diretamente antes do dado
 - Para arquivos binários: Se o tamanho do campo é inferior a 256 bytes, um byte é o suficiente para o indicador de comprimento
 - Para arquivos texto: são necessários $\text{floor}(\log_{10} \text{max})$ caracteres para o indicador de comprimento, sendo *max* o comprimento máximo do campo

```
05Maria05Rua 10312310São Carlos
04João05Rua A0325509Rio Claro
05Pedro06Rua 10025610Rib. Preto
```

- **Desvantagens:**

Campos com indicador de comprimento

- O **tamanho de cada campo** é armazenado diretamente antes do dado
 - Para arquivos binários: Se o tamanho do campo é inferior a 256 bytes, um byte é o suficiente para o indicador de comprimento
 - Para arquivos texto: são necessários $\text{floor}(\log_{10} \text{max})$ caracteres para o indicador de comprimento, sendo *max* o comprimento máximo do campo

```
05Maria05Rua 10312310São Carlos
04João05Rua A0325509Rio Claro
05Pedro06Rua 10025610Rib. Preto
```

- **Desvantagens:**
 - 1) Gasto de espaço para cada campo (vale a pena ou não dependendo da variabilidade do tamanho dos campos)

2) Complica o cálculo da posição de um dado campo

Campos separados por delimitadores

Caractere(s) especial(ais) (que não fazem parte do dado) são escolhido(s) para ser(em) inserido(s) ao final de cada campo

- Ex.: | , , ; , \$, % , ...
- Espaços em branco não servem na maioria dos casos

```
Maria|Rua 1|123|São Carlos|
João|Rua A|255|Rio Claro|
Pedro|Rua 10|56|Rib. Preto|
```

Ex: ?

Campos separados por delimitadores

Caractere(s) especial(ais) (que não fazem parte do dado) são escolhido(s) para ser(em) inserido(s) ao final de cada campo

- Ex.: | , , ; , \$, % , ...
- Espaços em branco não servem na maioria dos casos

```
Maria|Rua 1|123|São Carlos|
João|Rua A|255|Rio Claro|
Pedro|Rua 10|56|Rib. Preto|
```

Ex: arquivos .csv !!!

Campos separados por delimitadores

Caractere(s) especial(ais) (que não fazem parte do dado) são escolhido(s) para ser(em) inserido(s) ao final de cada campo

- Ex.: | , , ; , \$, % , ...
- Espaços em branco não servem na maioria dos casos

```
Maria|Rua 1|123|São Carlos|
João|Rua A|255|Rio Claro|
Pedro|Rua 10|56|Rib. Preto|
```

Ex: arquivos .csv !!!

Vantagens e desvantagens:

Campos separados por delimitadores

Caractere(s) especial(ais) (que não fazem parte do dado) são escolhido(s) para ser(em) inserido(s) ao final de cada campo

- Ex.: | , , ; , \$, % , ...
- Espaços em branco não servem na maioria dos casos

```
Maria|Rua 1|123|São Carlos|
João|Rua A|255|Rio Claro|
Pedro|Rua 10|56|Rib. Preto|
```

Ex: arquivos .csv !!!

Vantagens e desvantagens:

- **Gasta menos espaço que o indicador de conteúdo**
- **Tem que conferir cada caracter para achar o fim do campo**
 - Mas “ferramentas” atuais (ex: linguagens de programação com suporte a expressão regular) auxiliam bastante neste processo

Uso de uma *tag* do tipo "keyword=value"

- **Vantagem:** o campo fornece **informação semântica** sobre si próprio
 - Fica mais fácil identificar o conteúdo do arquivo

Ex: XML !!!

- **Desvantagem:** as *keywords* podem ocupar uma porção significativa do arquivo

```
Nome=Maria|Endereço=Rua 1|Número=123|Cidade=São Carlos|
Nome=João|Endereço=Rua A|Número=255|Cidade=Rio Claro|
Nome=Pedro|Endereço=Rua 10|Número=56|Cidade=Rib. Preto|
```

Organização INTERNA de arquivos (em memória secundária)

Em arquivos de dados, como organizar:

- os vários **campos** dentro de cada registro?
- os vários **registros** dentro do arquivo?



Métodos para organização em registros

- Tamanho fixo
- Número fixo de campos
- Indicador de tamanho
- Uso de índice
- Utilizar delimitadores

Registros de tamanho fixo

- Analogamente ao conceito de **campos de tamanho fixo**, assume que todos os registros têm o **mesmo tamanho**, com **campos de tamanho fixo ou não**
 - Um dos métodos mais comuns de organização de arquivos

Registro de tamanho fixo e campos de tamanho fixo:

Maria	Rua 1	123	São Carlos
João	Rua A	255	Rio Claro
Pedro	Rua 10	56	Rib. Preto

Registro de tamanho fixo e campos de tamanho variável:

Maria	Rua 1	123	São Carlos	← Espaço vazio →
João	Rua A	255	Rio Claro	← Espaço vazio →
Pedro	Rua 10	56	Rib. Preto	← Espaço vazio →

Como acesso os dados do i-ésimo registro?

Registros de tamanho fixo

- Analogamente ao conceito de **campos de tamanho fixo**, assume que todos os registros têm o **mesmo tamanho**, com **campos de tamanho fixo ou não**
 - Um dos métodos mais comuns de organização de arquivos

Registro de tamanho fixo e campos de tamanho fixo:

Maria	Rua 1	123	São Carlos
João	Rua A	255	Rio Claro
Pedro	Rua 10	56	Rib. Preto

Registro de tamanho fixo e campos de tamanho variável:

Maria	Rua 1	123	São Carlos	← Espaço vazio →
João	Rua A	255	Rio Claro	← Espaço vazio →
Pedro	Rua 10	56	Rib. Preto	← Espaço vazio →

Como acesso os dados do i -ésimo registro?

Começo em $(i-1)*R + 1$ (R = tamanho de um registro)

Vantagens e desvantagens?

Registros de tamanho fixo

- Analogamente ao conceito de **campos de tamanho fixo**, assume que todos os registros têm o **mesmo tamanho**, com **campos de tamanho fixo ou não**
 - Um dos métodos mais comuns de organização de arquivos

Registro de tamanho fixo e campos de tamanho fixo:

Maria	Rua 1	123	São Carlos
João	Rua A	255	Rio Claro
Pedro	Rua 10	56	Rib. Preto

Registro de tamanho fixo e campos de tamanho variável:

Maria	Rua 1	123	São Carlos	← Espaço vazio →
João	Rua A	255	Rio Claro	← Espaço vazio →
Pedro	Rua 10	56	Rib. Preto	← Espaço vazio →

Como acesso os dados do i -ésimo registro?

Começo em $(i-1)*R + 1$ (R = tamanho de um registro)

Vantagens e desvantagens? **Facilidade de cálculo** x **desperdício de espaço**

Registros com número fixo de campos

- Ao invés de especificar que cada registro contém um tamanho fixo, podemos especificar um **número fixo de campos**
 - O tamanho do registro é **variável**
 - Neste caso, os campos seriam separados por delimitadores

Registro com número fixo de campos:

```
Maria|Rua 1|123|São Carlos|João|Rua A|255|Rio Claro|Pedro|Rua  
10|56|Rib. Preto|
```

Registros com número fixo de campos

- Ao invés de especificar que cada registro contém um tamanho fixo, podemos especificar um **número fixo de campos**
 - O tamanho do registro é **variável**
 - Neste caso, os campos seriam separados por delimitadores

Registro com número fixo de campos:

```
Maria|Rua 1|123|São Carlos|João|Rua A|255|Rio Claro|Pedro|Rua  
10|56|Rib. Preto|
```

Como acesso o i-ésimo registro?

Registros com número fixo de campos

- Ao invés de especificar que cada registro contém um tamanho fixo, podemos especificar um **número fixo de campos**
 - O tamanho do registro é **variável**
 - Neste caso, os campos seriam separados por delimitadores

Registro com número fixo de campos:

```
Maria|Rua 1|123|São Carlos|João|Rua A|255|Rio Claro|Pedro|Rua  
10|56|Rib. Preto|
```

Como acesso o i -ésimo registro?

Preciso passar pelos $(i-1)*C$ campos (C = número de campos de cada registro)

Vantagens e desvantagens?

Registros com número fixo de campos

- Ao invés de especificar que cada registro contém um tamanho fixo, podemos especificar um **número fixo de campos**
 - O tamanho do registro é **variável**
 - Neste caso, os campos seriam separados por delimitadores

Registro com número fixo de campos:

```
Maria|Rua 1|123|São Carlos|João|Rua A|255|Rio Claro|Pedro|Rua  
10|56|Rib. Preto|
```

Como acesso o i -ésimo registro?

Preciso passar pelos $(i-1)*C$ campos (C = número de campos de cada registro)

Vantagens e desvantagens? Economia de espaço x dificuldade de acesso

Indicador de tamanho para registros

- O indicador que precede o registro fornece o seu **tamanho total**
 - Os campos são separados internamente por **delimitadores**
 - Boa solução para registros de tamanho variável

```
27Maria|Rua 1|123|São Carlos|25João|Rua A|255|Rio Claro|27Pedro|Rua 10|56|Ri  
b. Preto|
```

Como acesso o i-ésimo registro?

Indicador de tamanho para registros

- O indicador que precede o registro fornece o seu **tamanho total**
 - Os campos são separados internamente por **delimitadores**
 - Boa solução para registros de tamanho variável

```
27Maria|Rua 1|123|São Carlos|25João|Rua A|255|Rio Claro|27Pedro|Rua 10|56|Ri  
b. Preto|
```

Como acesso o i -ésimo registro?

Para cada $j < i$, acesso registro j para ler seu tamanho t_j , e pulo t_j chars/bytes

Vantagens e desvantagens?

Indicador de tamanho para registros

- O indicador que precede o registro fornece o seu **tamanho total**
 - Os campos são separados internamente por **delimitadores**
 - Boa solução para registros de tamanho variável

27Maria|Rua 1|123|São Carlos|25João|Rua A|255|Rio Claro|27Pedro|Rua 10|56|Ri
b. Preto|

Como acesso o i -ésimo registro?

Para cada $j < i$, acesso registro j para ler seu tamanho t_j , e pulo t_j chars/bytes

Vantagens e desvantagens?

Acesso mais simples que nr fixo de campos x gasto extra de espaço

Utilizar um índice

- Um **índice externo** poderia indicar o deslocamento de cada registro relativo ao início do arquivo
 - Pode ser utilizado também para calcular o **tamanho dos registros**
 - Os campos seriam separados por **delimitadores**

Arquivos de dados + arquivo de índices:

Dados: Maria|Rua 1|123|São Carlos|João|Rua A|255|Rio Claro|Pedro|Rua
10|56|Rib. Preto|

Índice: 00 27 52 †

Como acesso o i-ésimo registro?

Utilizar um índice

- Um **índice externo** poderia indicar o deslocamento de cada registro relativo ao início do arquivo
 - Pode ser utilizado também para calcular o **tamanho dos registros**
 - Os campos seriam separados por **delimitadores**

Arquivos de dados + arquivo de índices:

Dados:	Maria		Rua 1		123		São Carlos		João		Rua A		255		Rio Claro		Pedro		Rua	10		56		Rib. Preto		
Índice:					00				27																52	

Como acesso o i -ésimo registro?

Leio a i -ésima entrada do arquivo de índices (desloc_i) e pulo ($\text{desloc}_i - 1$) chars/bytes

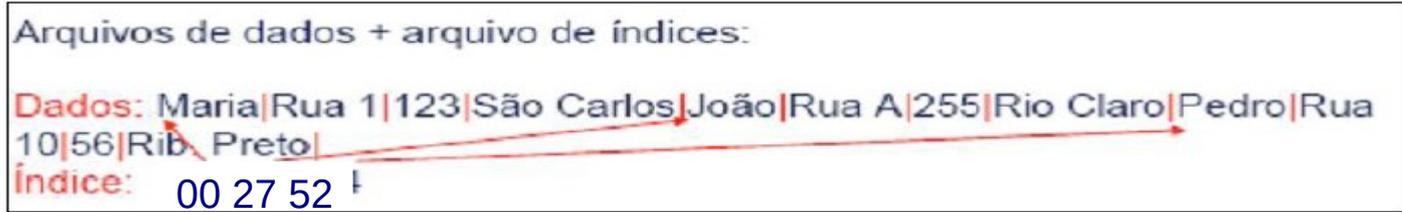
Vantagens e desvantagens?

Utilizar um índice

- Um **índice externo** poderia indicar o deslocamento de cada registro relativo ao início do arquivo
 - Pode ser utilizado também para calcular o **tamanho dos registros**
 - Os campos seriam separados por **delimitadores**

Arquivos de dados + arquivo de índices:

```
Dados: Maria|Rua 1|123|São Carlos|João|Rua A|255|Rio Claro|Pedro|Rua
10|56|Rib. Preto|
Índice: 00 27 52 †
```



Como acesso o i -ésimo registro?

Leio a i -ésima entrada do arquivo de índices ($start_i$) e pulo ($start_i - 1$) chars/bytes

Vantagens e desvantagens?

Acesso aleatório BEM mais simples

Gasto extra de espaço similar ao de indicador de tamanho

Utilizar delimitadores

- Separar os registros com **delimitadores** análogos aos de fim de campo
 - O delimitador de campos é mantido, sendo que o método combina os dois delimitadores
 - Note que delimitar fim de campo é diferente de delimitar fim de registro

Registro delimitado por marcador (#):

```
Maria|Rua 1|123|São Carlos|#João|Rua A|255|Rio Claro|#Pedro|Rua 10|56|Rib. Preto|
```

Como acesso o i-ésimo registro?

Utilizar delimitadores

- Separar os registros com **delimitadores** análogos aos de fim de campo
 - O delimitador de campos é mantido, sendo que o método combina os dois delimitadores
 - Note que delimitar fim de campo é diferente de delimitar fim de registro

Registro delimitado por marcador (#):

```
Maria|Rua 1|123|São Carlos|#João|Rua A|255|Rio Claro|#Pedro|Rua 10|56|Rib. Preto|
```

Como acesso o i -ésimo registro?

Preciso passar pelos $(i-1)$ delimitadores

Vantagens e desvantagens?

Utilizar delimitadores

- Separar os registros com **delimitadores** análogos aos de fim de campo
 - O delimitador de campos é mantido, sendo que o método combina os dois delimitadores
 - Note que delimitar fim de campo é diferente de delimitar fim de registro

Registro delimitado por marcador (#):

```
Maria|Rua 1|123|São Carlos|#João|Rua A|255|Rio Claro|#Pedro|Rua 10|56|Rib. Preto|
```

Como acesso o i -ésimo registro?

Preciso passar pelos $(i-1)$ delimitadores

Vantagens e desvantagens? Economia de espaço x dificuldade de acesso

Cabeçalho de arquivos

- Um formato de arquivo precisa definir como os campos/registros são organizados
- Cabeçalho do arquivo (descriptor na parte inicial do arquivo) pode conter informações (meta-informações) como:
 - Descrição dos formatos (tamanhos) dos campos de um registro
 - como os campos/registros são organizados
 - Data de criação e atualização
 - Outras que veremos mais adiante...

Permite definição de padrões (ex: pdf, tiff, jpg) e facilita conversão entre padrões

Exercícios

ACH2024 Algoritmos e Estruturas de Dados II

Semestre 01/2017 - turma 02

Prof. Ivandré Paraboni ivandre@usp.br

Estagiário PAE - Fernando Henrique da Silva Costa fhscosta0993@gmail.com

Exercícios básicos em arquivos

Serialização de EDs - faça você mesmo.

Passo 1: Escolha uma estrutura de dados dentre *lista sequencial* (i.e., vetor), *lista ligada dinâmica* (simples ou dupla), *árvore de busca binária*, *grafo em listas de adjacências* ou *grafo em matriz de adjacências*.

Passo 2: Escolha a operação de *leitura* (do arquivo para a memória) ou *escrita* (da ED em arquivo). Tipicamente a leitura é mais trabalhosa, pois envolve recriar a ED usando *malloc()* etc.

Passo 3: Escolha o tipo de arquivo: texto com delimitadores (use *fprintf/fscanf*) ou binário (use *fwrite/fread*).

Passo 4: Implemente o algoritmo correspondente à configuração escolhida (ED, operação, tipo de arquivo) e, em caso de dúvidas, teste no próprio compilador.

Exercícios em arquivos simples (sem índices)

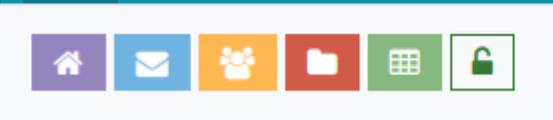
Considere arquivos de registros de tamanho fixo do tipo *REGISTRO* como segue:

```
typedef struct {  
    int NroUSP; // chave primária  
    int curso;  
    int estado;  
    int idade;  
    bool valido; // para exclusão lógica  
} REGISTRO;
```

Para evitar reorganização do arquivo inteiro a cada exclusão de registro

Todos os arquivos são mantidos em ordem aleatória de chaves. Para os exercícios a seguir, não há nenhuma estrutura de índices disponível.

1. Reescreva um arquivo arq1 em um novo arquivo arq2 eliminando os registros inválidos.
2. Faça uma cópia invertida de arq1 em um novo arquivo arq2, ou seja: copie o último registro (n) de arq1 para o início de arq2, depois copie o registro n-1 para a segunda posição etc.
3. Escreva uma função para inserir um novo registro r no arquivo, tomando cuidado para evitar chaves duplicadas.
4. Escreva uma função que, dada um nroUSP X, retorne o registro correspondente.
5. Escreva uma função para excluir todos os registros do curso X.
6. Escreva uma função para alterar o curso de um aluno de nroUSP X para o curso Y.
7. Implemente o procedimento de ordenação *KeySort*, que dado um arquivo arq1 cria uma tabela temporária de chaves em memória (idêntica a uma tabela de índices primários) e então reescreve o arquivo em um novo arquivo de saída arq2, na ordem correta de chaves (exercício completo e altamente recomendável).



ACH2024 - Algoritmos e Estruturas de Dados II



ARQUIVO

[serializa_binario_grafo.c](#)

Programa para serializar um grafo (matriz de adjacências) em um arquivo binário



ARQUIVO

[le_arquivo_delimitadores.c](#)

Útil para os exercícios do slide anterior!!!

Programa que lê um arquivo assumindo que tanto registros quanto campos são separados por delimitadores.

Referências

- Slides da Profa. Graça (ICMC) - [http://wiki.icmc.usp.br/index.php/SCC-203_\(gracan\)](http://wiki.icmc.usp.br/index.php/SCC-203_(gracan)) (Arquivos 8, 9 e 12)
- GOODRICH et al, **Data Structures and Algorithms in C++**. Ed. John Wiley & Sons, Inc. 2nd ed. 2011. Seção 14.2
- ELMARIS, R.; NAVATHE, S. B. **Fundamentals of Database Systems**. 4 ed. Ed. Pearson-Addison Wesley. Cap 13 (até a seção 13.7).
- TANEMBAUM, A. S. & BOS, H. **Modern Operating Systems**. Pearson, 4th ed. 2015