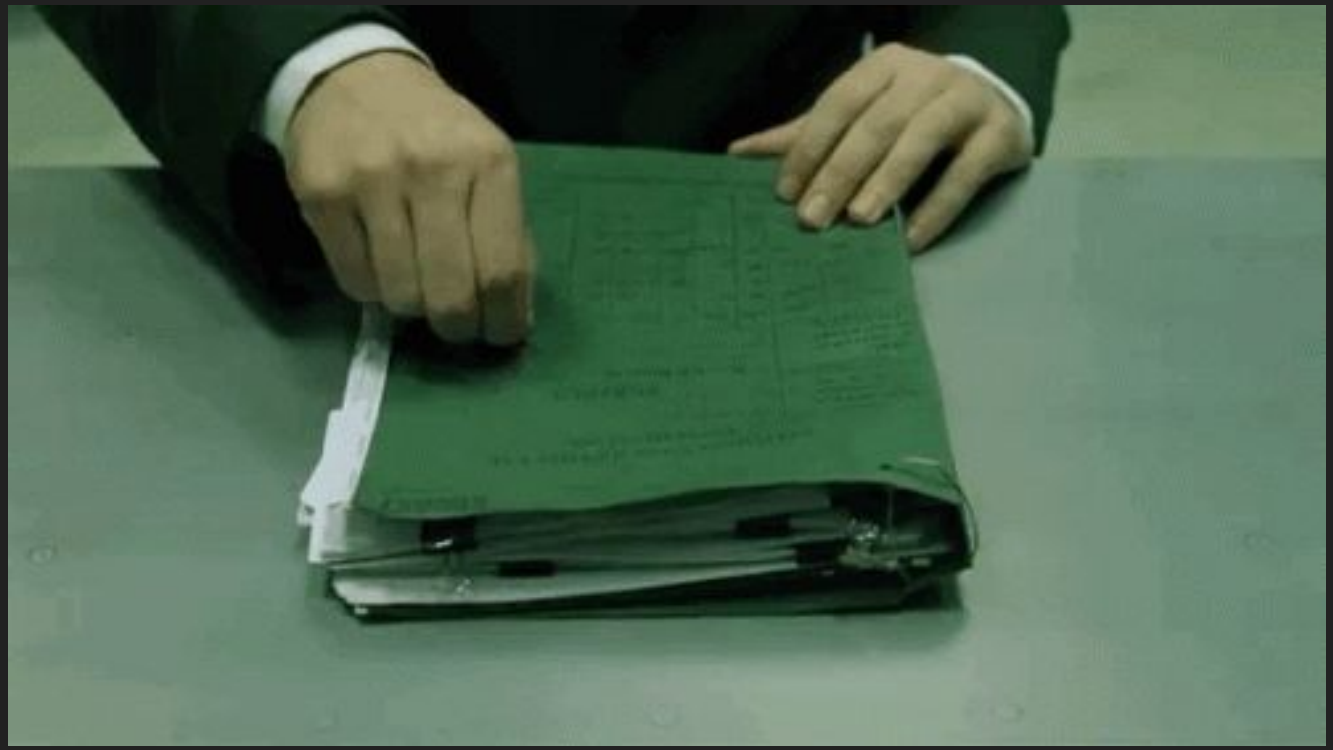
SSC0800 - Introdução à Ciência de Computação I

Arquivos

Prof.: Leonardo Tórtoro Pereira
leonardop@usp.br
Baseado no material do prof Cesar Henrique Comin

O que vamos aprender hoje?

Arquivos



Arquivos

- Mecanismo de organização de informação mantida em memória secundária
  - Acessada via computador
- Disco (HDD)
- Disquete, Fitas Magnéticas
- CD, DVD
- Pen-drive, cartões, SSD
- ...

Arquivos

- Por que utilizamos arquivos?
  - Armazena grande quantidade de memória a um custo relativamente baixo
  - Armazena memória de maneira não-volátil
  - Permite a persistência dos dados

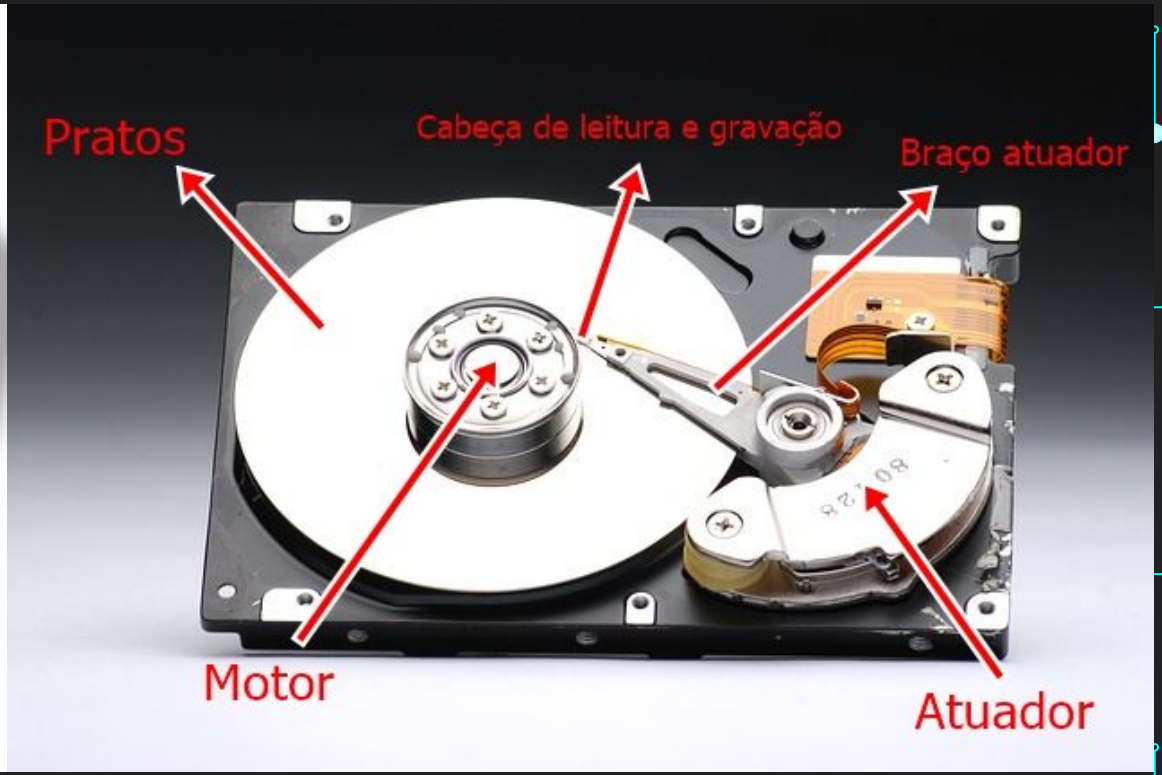
Arquivos

- Desvantagens
  - Alto tempo de acesso
  - Discos e outros dispositivos de armazenamento secundário são lentos*

*Pode variar bastante

Dispositivos de Armazenamento

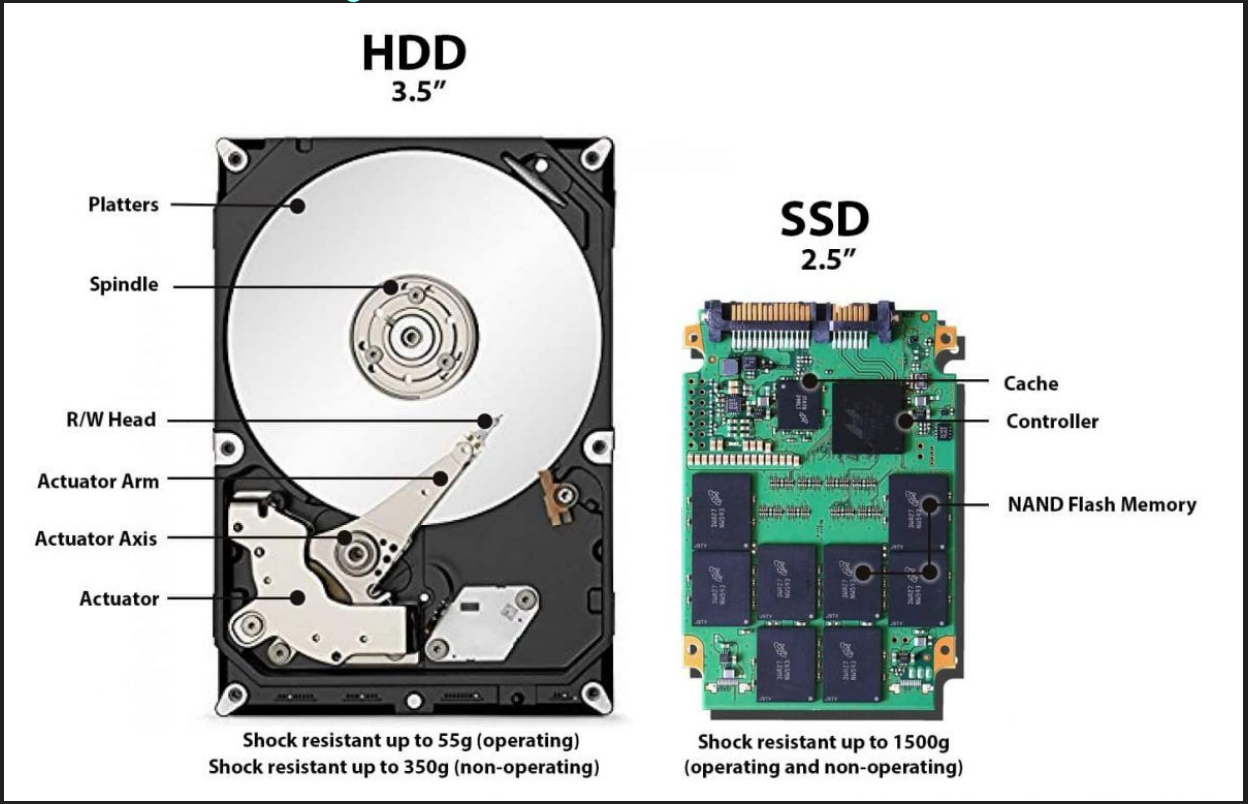
0 HDD



Disco (HDD)

Fonte (Dir.): https://techenter.com.br/o-que-e-hdd-hard-disk-drive/

O SSD



HDD vs SSD (Fonte: https://www.leak.pt/ssd-vs-hdd-existe-desenvolvimento-de-hdds/)

A RAM

Fonte: https://i.imgur.com/ALKguuw.jpg

Fonte: https://www.infoescola.com/anfibios/ra-animal/

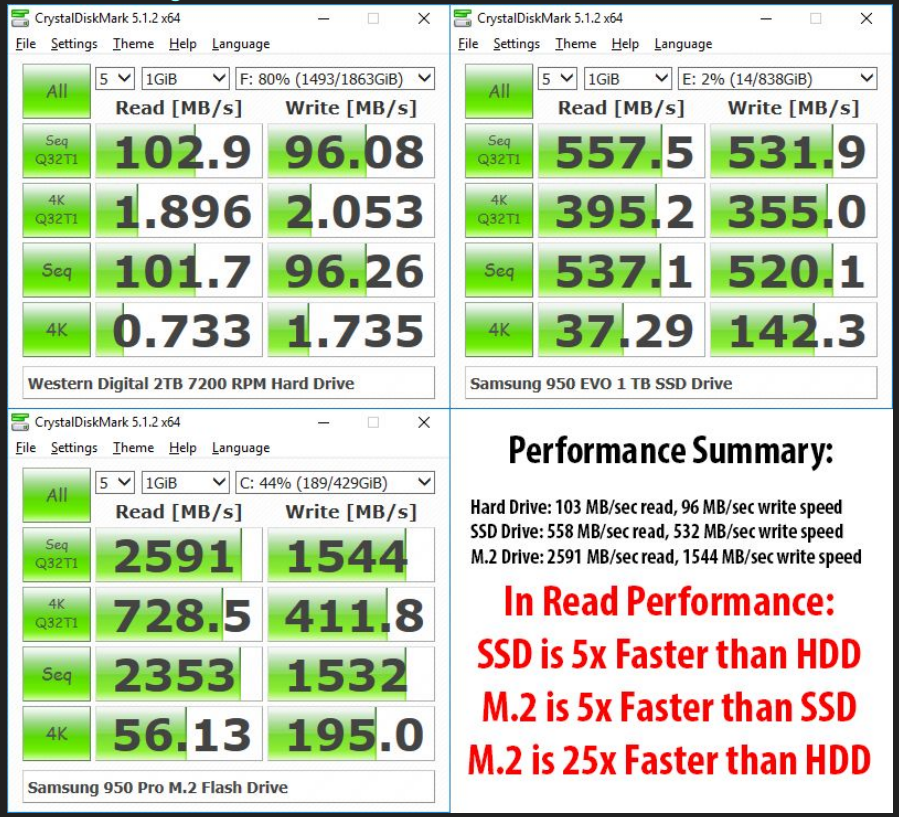
Fonte: Chrono Trigger

RAM

Fontes: https://br.crucial.com/memory/ddr4/ct8g4dfs824a

Dispositivos de Armazenamento

- HDD
  - Disco mecânico
  - Rotacionar e apontar uma agulha no local adequado
- SSD
  - Semicondutores, Controlador acessa células NAND
  - Memória NAND ou NVM
- Ram
  - Circuitos integrados, Semicondutores



Fonte: https://emolike.net/nvme-vs-ssd-vs-hdd-performance
Programa usado: https://crystalmark.info/en/software/crystaldiskmark/

Friendly name	Industry name	Peak Transfer Rate	Data transfers/second (in millions)
DDR4-2400	PC4-19200	19200 MB/s	2400
DDR4-2666	PC4-21300	21300 MB/s	2666
DDR4-4000	PC4-32000	32000 MB/s	4000
DDR4-4400	PC4-35200	35200 MB/s	4400

Velocidade DDR4

Fonte: https://www.crucial.com/support/memory-speeds-compatability

Friendly name	Industry name	Peak Transfer Rate	Data transfers/second (in millions)
DDR2-400	PC2-3200	3200 MB/s	400
DDR2-533	PC2-4200	4266 MB/s	533
DDR2-667	PC2-5300	5333 MB/s	667
DDR2-800	PC2-6400	6400 MB/s	800
DDR2-1000	PC2-8000	8000 MB/s	1000

Velocidade DDR2

Fonte: https://www.crucial.com/support/memory-speeds-compatability

Dispositivos de Armazenamento

- Portanto, mesmo as memórias RAM mais antigas (DDR2) superam os melhores SSDs atuais.
- Para as memórias mais recentes (DDR4), a transferência chega a ser 10x mais rápido!
- E também temos a questão da latência!

Dispositivos de Armazenamento

- HDD
  - Alguns milisegundos => ~2ms-30ms [1, 2];
- SSD
  - Alguns microssegundos => ~10µs-100µs [3];
- RAM
  - Alguns nanossegundos => ~10ns-30ns [4];

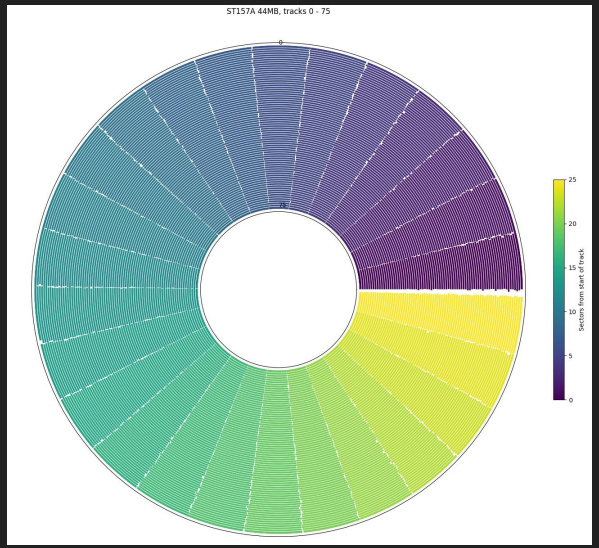
HDD	SSD	RAM
10^-3s	10^-6s	10^-9s

	RAM	Discos (HDD)	SSD
Custo	Alto	Baixo	Intermediário
Tempo de Acesso	**Baixo**	**Alto**	**Médio**
Capacidade	Baixa	Alto	Alta
Princípio	Elétrico	Magnético	Elétrico
Persistência	Volátil	Não-volátil	Não-volátil
Acesso	Aleatório	Aleatório	Aleatório
Organização	Células	Trilhas/Setores	Células

Sistema de Arquivos

Sistema de Arquivos

- ➔ Formatação Física
  - ◆ Organização do disco em setores/trilhas
    - Vem da fábrica

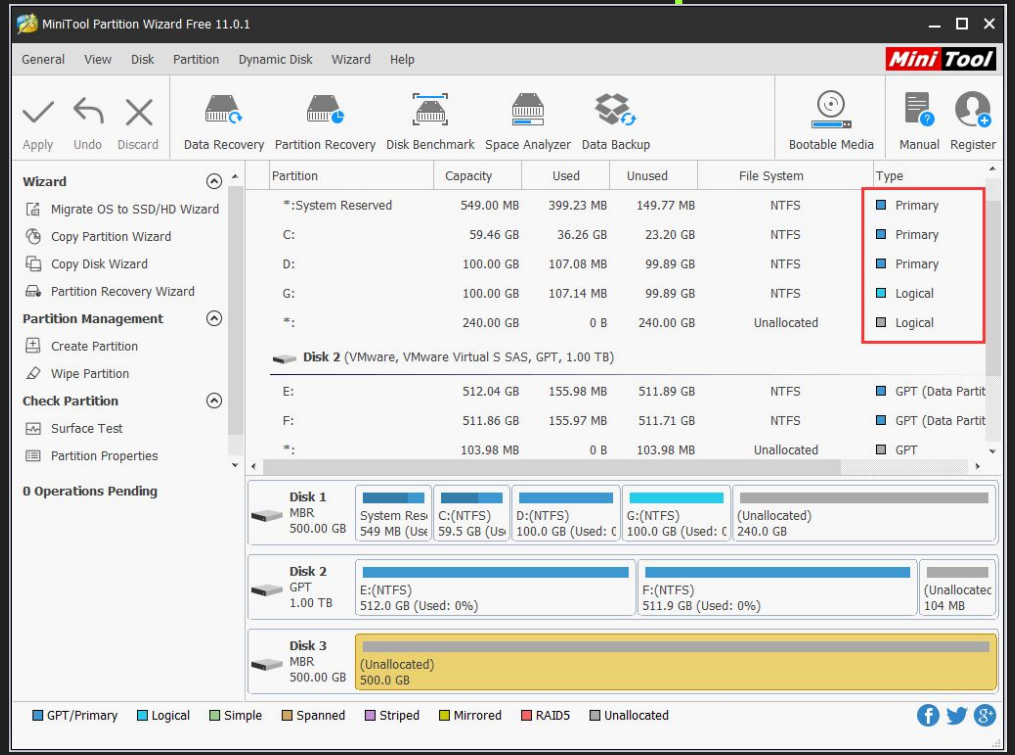


Fonte: https://blog.stuffedcow.net/2019/09/hard-disk-geometry-microbenchmarking/

Sistema de Arquivos

- Formatação Lógica
  - ‘Instala’ o sistema de arquivos no disco:
    - Subdivide o disco em regiões endereçáveis;
    - Grava estruturas de gerenciamentos dos arquivos.
    - Overhead de espaço ocupado com informações para gerenciamento

Sistema de Arquivos



Fonte: https://www.partitionwizard.com/partitionmanager/primary-partition-vs-logical-drive.html

Sistema de Arquivos

- Sistema de Arquivos
  - Faz parte do sistema operacional (S.O.)
  - Fornece a infraestrutura básica para a manipulação de arquivos em memória secundária via software
  - Oferece um conjunto de operações para a manipulação de arquivos

Operações Sistema de Arquivos

criar (create, open)	destruir ou remover (delete)
renomear (rename)	abrir (open)
fechar (close)	ler dados (read)
escrever dados (write)	escrever dados no final (append)
posicionar (seek)	...

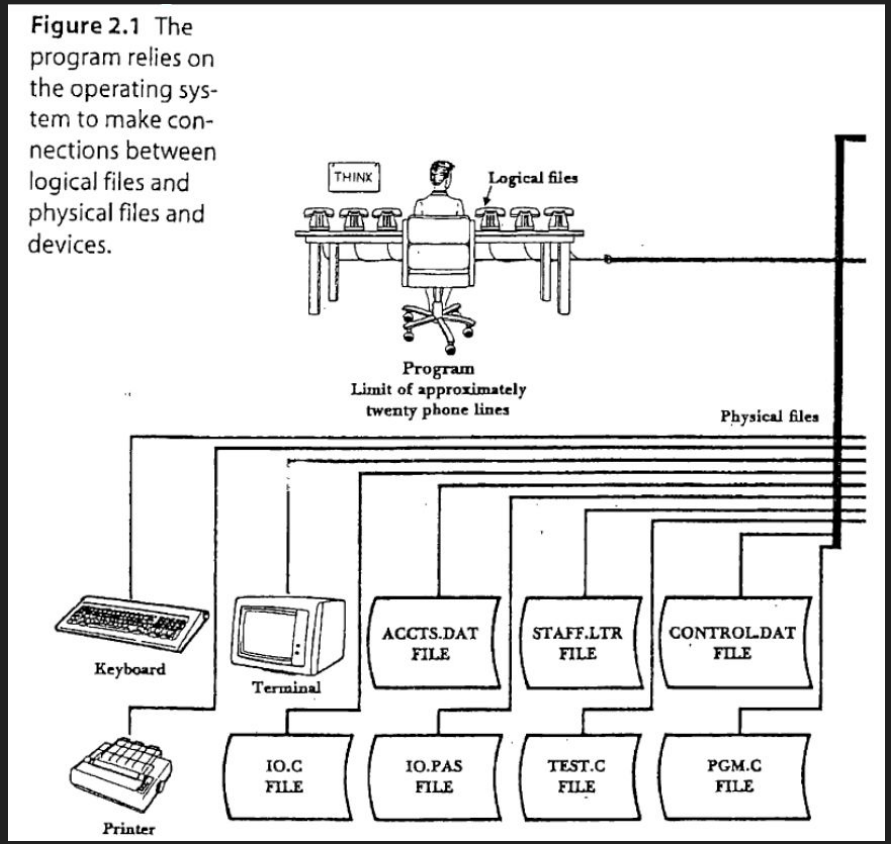
Arquivos Físico e Lógico

Arquivos Físico e Lógico

- Arquivo Físico:
  - Sequência de bytes armazenada no disco em blocos distribuídos em trilhas/setores.
- Arquivo Lógico:
  - Arquivo como visto pelo aplicativo que o acessa – sequência contínua de registros ou bytes.
- Associação arquivo físico/arquivo lógico:
  - Gerenciada pelo Sistema de Arquivo/S.O.

Arquivos Físico e Lógico

- Para uma aplicação, um arquivo é como uma conexão de telefone em uma rede de telefones
- O programa pode receber e enviar bytes pela linha
  - Mas não sabe para onde eles vão ou da onde eles vem
- O programa sabe apenas sobre o fim da sua linha
- Essa linha é o “arquivo lógico”



Analogia de arquivo lógico e físico. Fonte: [1]

Arquivo Físico

➔ Sequência de bytes

0	1		4095	4096	4097		8191	8192	8193		12287	12288	12289		16383
B0	80	...	D1	28	19	...	01	00	13	...	2A	3E	1F	...	33

byte 1 = 2º byte

byte 0 = 1º byte

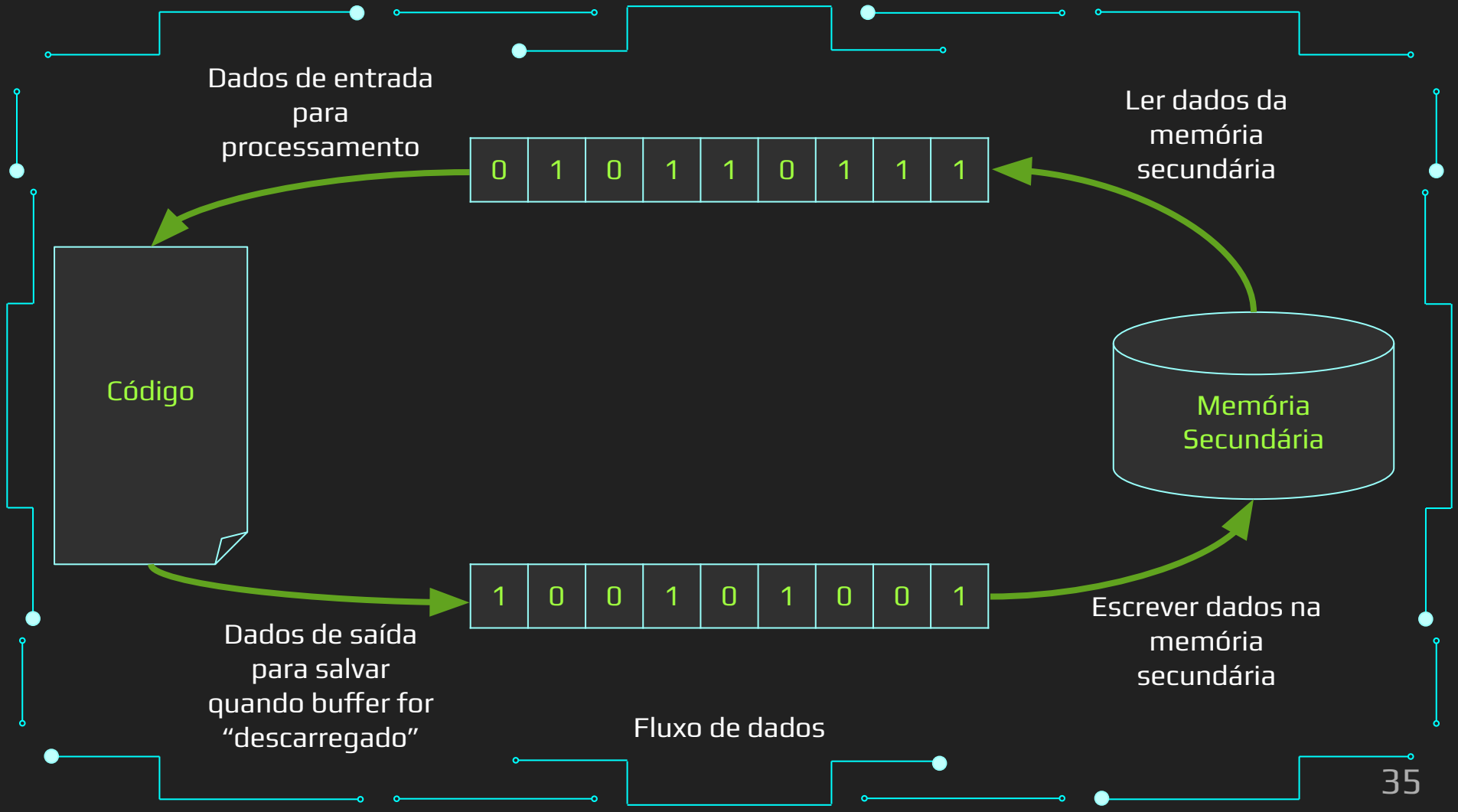
byte n = (n+1) byte

tamanho:
16.384 bytes ou 16KB

Arquivo Lógico

Arquivo Lógico

- Para a maioria das linguagens de programação:
  - Informações lidas e gravadas em arquivos como streams de bytes.
  - Gerenciar arquivos é gerenciar esses streams.
- Não importa qual a fonte e organização do arquivo físico!
- Tudo isso é abstraído pelo SO (ainda bem)


Dados de entrada para processamento
0 1 0 1 1 0 1 1 1
Ler dados da memória secundária
Código
Memória Secundária
1 0 0 1 0 1 0 0 1
Dados de saída para salvar quando buffer for “descarregado”
Escrever dados na memória secundária
Fluxo de dados

Fechamento de Arquivos

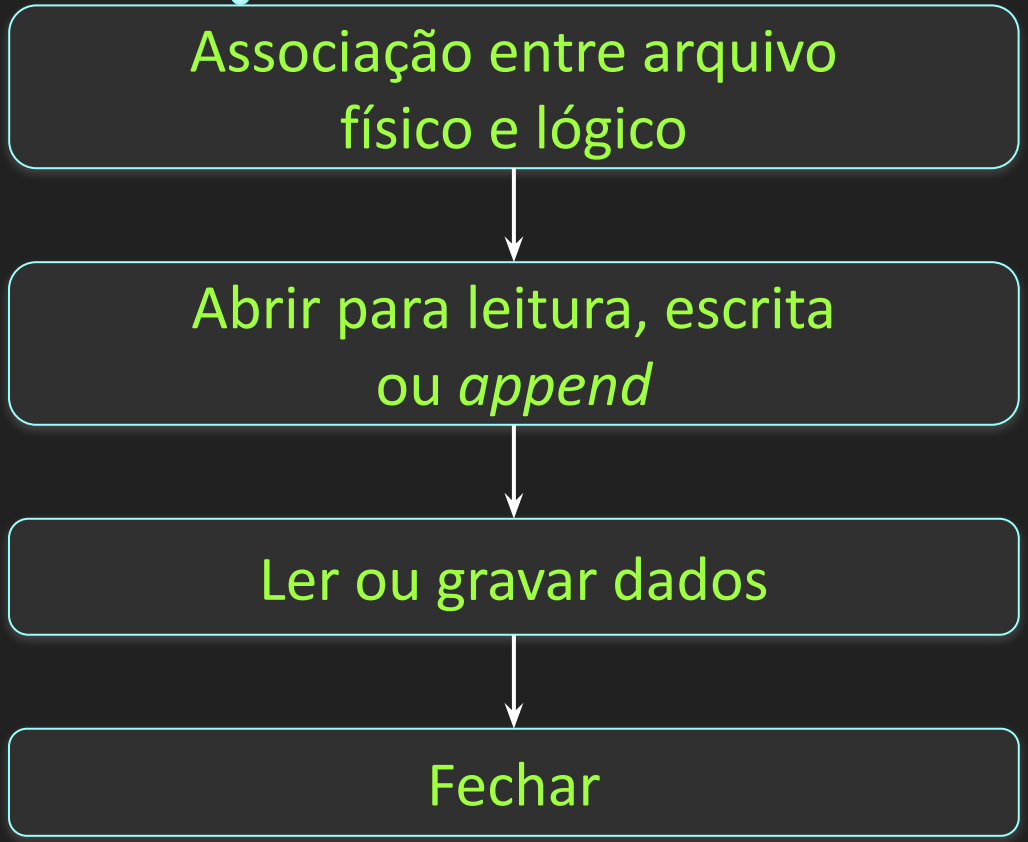
- Encerra a associação entre arquivos lógico e físico
  - Todas as informações são atualizadas e salvas
  - Conteúdo dos buffers de E/S enviados para o arquivo.

Fechamento de Arquivos

- S.O. fecha o arquivo se o aplicativo não o fizer ao final da execução do programa.
- Interessante para:
  - Garantir que os buffers sejam descarregados;
  - Liberar as estruturas associadas ao arquivo para outros arquivos.
- É importante fechar ainda assim para evitar perdas de dados por interrupção e liberar os nomes lógicos cedo

Fechamento de Arquivos

```c
int main ()
{
  FILE * pFile;
  pFile = fopen ("myfile.txt","w");
  if (pFile != NULL)
  {
    fclose (pFile);
  }
  return 0;
}
```



Operações Sobre Arquivos

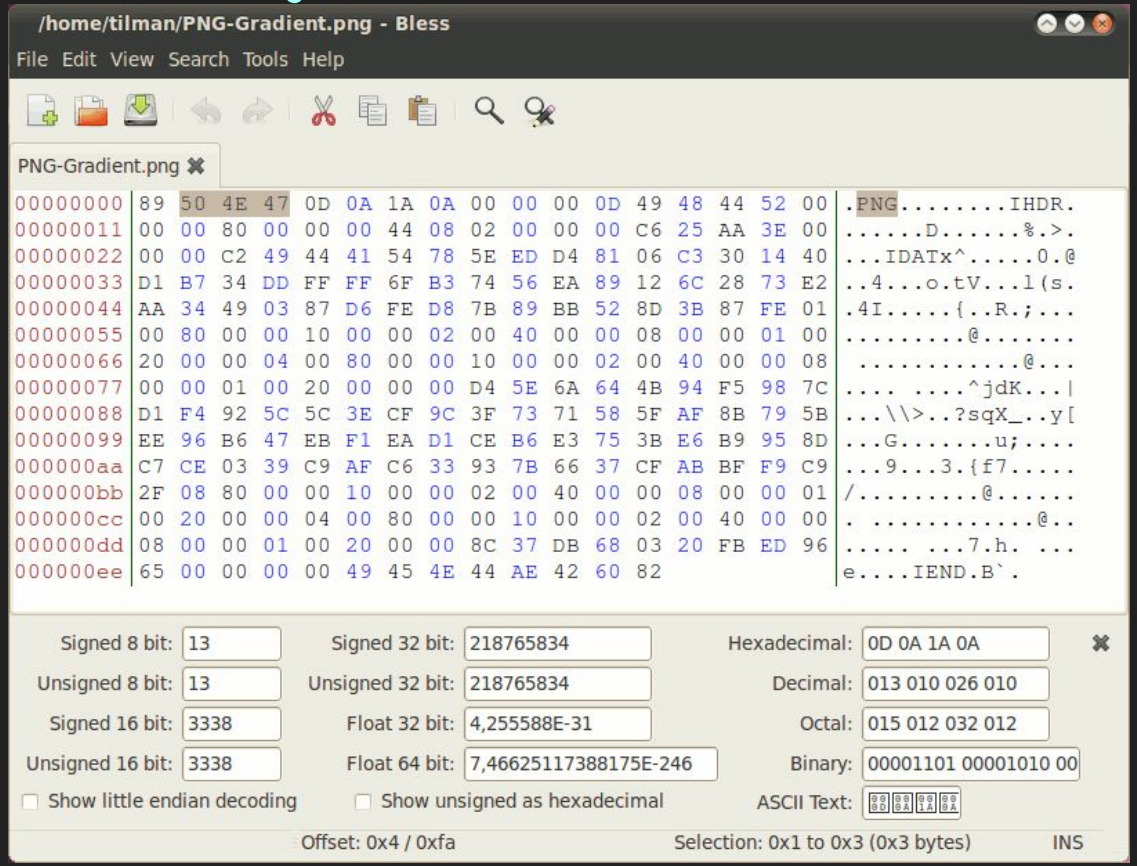
Aplicações

Aplicações

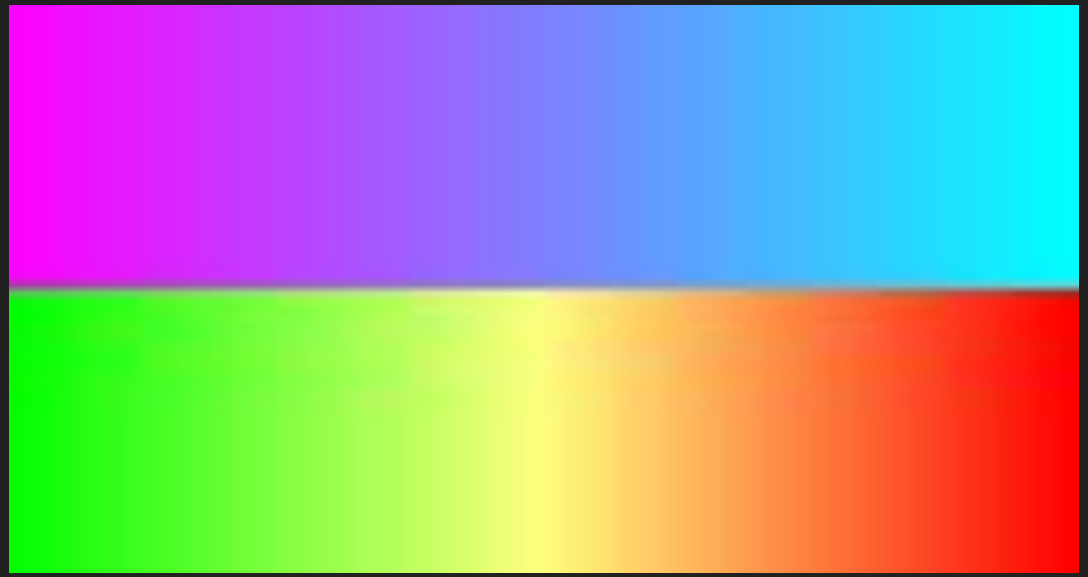
- O uso de arquivos na memória secundária são vastos
- Em geral, podem ser resumidos a:
  - Permanência dos dados
    - Não são deletados ao final do programa
  - Salvar dados maiores
    - Limite muito maior que o da RAM
  - Portabilidade dos dados
    - Pode-se transferir dados entre dispositivos

Aplicações

- Alguns exemplos de uso de arquivos
  - Salvar dados das aplicações
    - Logs, resultados, backups, perfis, histórico, etc.
  - Salvar arquivos multimídia
    - Imagens, vídeos, músicas, etc.
  - Jogos
    - Progresso, modelos 3D, texturas, sons, etc.



Uma imagem PNG. Fonte: https://en.wikipedia.org/wiki/PNG



Uma imagem PNG. Fonte: https://en.wikipedia.org/wiki/PNG

Arquivo de progresso do Stardew Valley. Fonte: https://stardewvalleywiki.com/Saves

```xml
<?xml version="1.0" encoding="utf-8"?>
<SaveGame xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <player>
        <name>Jesse</name>
        <isEmoting>false</isEmoting>
        <isCharging>false</isCharging>
        <willDestroyObjectsUnderfoot>true</willDestroyObjectsUnderfoot>
        <isGlowing>false</isGlowing>
        <coloredBorder>false</coloredBorder>
        <flip>false</flip>
        <drawOnTop>false</drawOnTop>
        <faceTowardFarmer>false</faceTowardFarmer>
        <faceAwayFromFarmer>false</faceAwayFromFarmer>
        <ignoreMovementAnimation>false</ignoreMovementAnimation>
        <scale>1</scale>
        <timeBeforeAIMovementAgain>0</timeBeforeAIMovementAgain>
        <glowingTransparency>0.7000001</glowingTransparency>
        <glowRate>0.05</glowRate>
        <Position>
            <X>1760</X>
            <Y>864</Y>
        </Position>
        <Speed>5</Speed>
        <IsEmoting>false</IsEmoting>
        <CurrentEmote>24</CurrentEmote>
        <questLog>
            <Quest>
                <_currentObjective>&amp;quot;Give the sand dragon his final meal.&amp;quot;</_currentObjective>
                <_questDescription>You've found yet another strange note within the Mayor's fridge. This time, the instructions are more cryptic.</_questDescription>
                <_questTitle>Der mysteriöse Qi</_questTitle>
                <rewardDescription>-1</rewardDescription>
                <accepted>false</accepted>
                <completed>false</completed>
                <dailyQuest>false</dailyQuest>
                <showNew>false</showNew>
                <canBeCancelled>false</canBeCancelled>
                <destroy>false</destroy>
                <id>4</id>
                <moneyReward>0</moneyReward>
                <questType>1</questType>
                <daysLeft>0</daysLeft>
                <nextQuests>
                    <int>-1</int>
                </nextQuests>
                <questTitle>The Mysterious Qi</questTitle>
            </Quest>
```

Arquivo de progresso do Stardew Valley. Fonte: https://stardewvalleywiki.com/Saves

Vamos Praticar

Base de Dados

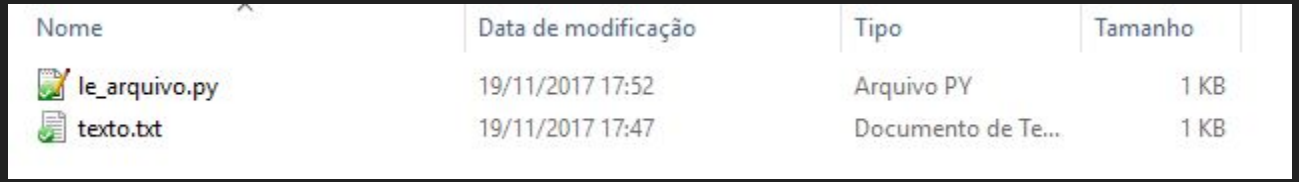
- https://www.kaggle.com/datasets/hamzacyberpatcher/data-of-1010-pokemons/

Colab

Arquivos em Python

Arquivos em Python

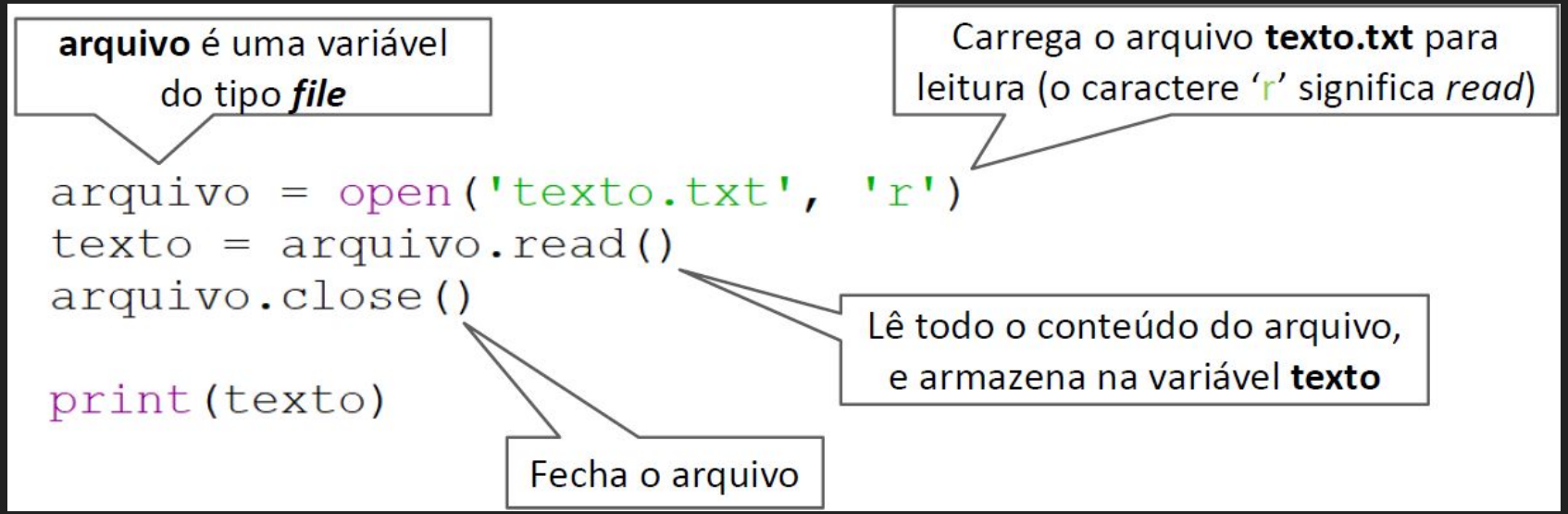
- Suponha que temos em um diretório do computador o arquivo “texto.txt”, contendo informações que queremos processar utilizando um programa Python.
- Como podemos ler o conteúdo do arquivo em um programa?



Nome	Data de modificação	Tipo	Tamanho
le_arquivo.py	19/11/2017 17:52	Arquivo PY	1 KB
texto.txt	19/11/2017 17:47	Documento de Te...	1 KB

Arquivos em Python

- Primeiro passo: criar um programa Python na mesma pasta onde o arquivo se encontra
- Segundo passo: utilizar a função open



Fonte: Slides do Prof. Cesar Henrique Comin

Arquivos em Python

- É muito importante lembrar de fechar o arquivo ao terminar de utilizá lo! Se o arquivo não for fechado, ele pode ficar “preso” pelo sistema operacional.

Arquivos em Python

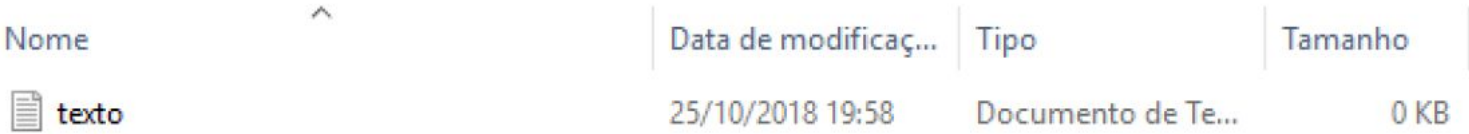
- No Windows, dependendo de como ele foi configurado, pode ser que a extensão do arquivo esteja escondida.
- Se isso ocorrer, o arquivo texto.txt aparecerá na pasta apenas como texto
  - Nesse caso, ainda precisamos utilizar o nome de arquivo juntamente com a extensão dele para realizar a leitura

Arquivos em Python

- Caso a extensão esteja escondida, é um erro comum o arquivo ser nomeado como texto.txt.txt

Arquivo de texto com a extensão sendo mostrada

Arquivo de texto com a extensão escondida



Nesse caso, se nomeássemos o arquivo como ***texto.txt***, ele na verdade seria salvo como ***texto.txt.txt***

Fonte: Slides do Prof. Cesar Henrique Comin


```
arquivo = open('texto.txt', 'r')
texto = arquivo.read()
arquivo.close()

print(texto)
```

```
Pedro 6.4
Juliana 7.6
Samara 9.5
Lucas 5.6
Silvio 8.3
```

Fonte: Slides do Prof. Cesar Henrique Comin

```
>>> texto
'Pedro 6.4\nJuliana 7.6\nSamara 9.5\nLucas 5.6\nSilvio 8.3\n'
>>> print(texto)
Pedro 6.4
Juliana 7.6
Samara 9.5
Lucas 5.6
Silvio 8.3
```

Fonte: Slides do Prof. Cesar Henrique Comin

Arquivos em Python

- A variável texto na realidade possui a string indicada no primeiro comando.
- Ao utilizarmos a função print , a string contida em texto é modificada para ser mostrada na tela de forma mais intuitiva.
- O caractere ‘\n’ é um caractere especial, chamado de caractere de controle. Ele marca o fim de uma linha.

Arquivos em Python

- E se quisermos separar as linhas do arquivo?
- Abordagem 1, utilizando o método splitlines , que opera sobre strings

```
arquivo = open('texto.txt', 'r')
texto = arquivo.read()
arquivo.close()
linhas = texto.splitlines()

print(linhas)
```

```
['Pedro 7.8', 'Juliana 4.5', 'Samara 9.4', 'Lucas 7.4', 'Silvio 6.4']
```

Arquivos em Python

- E se quisermos separar as linhas do arquivo?
- Abordagem 2, utilizando o método readlines , que opera sobre arquivos:

```
arquivo = open('texto.txt', 'r')
texto = arquivo.readlines()
arquivo.close()

print(texto)
```

```
['Pedro 6.4\n', 'Juliana 7.6\n', 'Samara 9.5\n', 'Lucas 5.6\n', 'Silvio 8.3\n']
```

Arquivos em Python

- E se quisermos separar as linhas do arquivo?
- Abordagem 3, utilizando estrutura de repetição:

```
arquivo = open('texto.txt', 'r')
texto = []
for linha in arquivo:
    texto.append(linha)
arquivo.close()

print(texto)
```

```
['Pedro 6.4\n', 'Juliana 7.6\n', 'Samara 9.5\n', 'Lucas 5.6\n', 'Silvio 8.3\n']
```

Escrita de arquivo

- Utilizamos novamente a função open, mas agora passamos o parâmetro ‘w ’ ao invés de ‘ r ’.
- Também utilizamos a função write
- Se o arquivo não existir, ele é automaticamente criado. Se ele existir, o conteúdo do arquivo é apagado

```
texto = "Esse texto sera escrito no arquivo"

arquivo = open('mensagem.txt', 'w')
arquivo.write(texto)
arquivo.close()
```

O caractere ‘w’ significa *write*

Fonte: Slides do Prof. Cesar Henrique Comin

Escrita de arquivo

- E se quisermos escrever diversas linhas?
- Utilizamos o caractere ‘\n’

```
texto = "Pedro\nSamara\nAline\nClara\nSilvio\n"

arquivo = open('mensagem.txt', 'w')
arquivo.write(texto)
arquivo.close()
```

Escrita de arquivo

- E se quisermos escrever diversas linhas?
- Se tivermos uma lista de strings , precisamos utilizar uma estrutura de repetição para escrever cada linha:

```
nomes = ["Pedro","Samara","Aline","Clara","Silvio"]

arquivo = open('mensagem.txt', 'w')
for i in range(len(nomes)):
    nome = nomes[i]
    arquivo.write(nome+'\n')
arquivo.close()
```

Escrita de arquivo

- Vimos que ao abrirmos o arquivo para escrita, o texto contido no arquivo é apagado. Para evitarmos isso, podemos abrir o arquivo em modo append , utilizando o parâmetro ‘a’
- Nesse caso, o conteúdo que for escrito no arquivo é adicionado ao final do mesmo.

```
arquivo = open("mensagens.txt", "a")
```

Referências

1. M. J. Folk, B. Zoellick and G. Riccardi. File Structures: An object-oriented approach with C++, Addison Wesley, 1998.
2. https://www.learnpython.org/
3. https://www.w3schools.com/python/
4. https://panda.ime.usp.br/cc110/static/cc110/index.html
5. https://www.youtube.com/playlist?list=PLcoJJSvnDgcKpOi_UeneTNTIVOigRQwcn