Arquivos

SSC0301

Prof Márcio Delamaro

O que é

- Em arquivologia, arquivo é um conjunto de documentos criados ou recebidos por uma organização, firma ou indivíduo, que os mantém ordenadamente como fonte de informação para a execução de suas atividades.
- Um arquivo de computador é um recurso para armazenamento de informação, que está disponível a um programa de computador e é normalmente baseado em algum tipo de armazenamento durável. Um arquivo é durável no sentido que permanece disponível aos programas para utilização após o programa em execução ter sido finalizado.

Arquivo texto

- Pode ser visto como um string
- Pode ser visto como uma sequência de linhas
- Cada linha contém um string

A luz, o sol, o ar livre
envolvem o sonho do engenheiro.
O engenheiro sonha coisas claras:
superfícies, tênis, um copo de água.

(João Cabral de Melo Neto)

ICMC USP SÃO CARLOS SSC

Arquivo texto

- Pode ser visto como um string
- Pode ser visto como uma sequência de linhas
- Cada linha contém um string

A luz, o sol, o ar livre

envolvem o sonho do engenheiro.

O engenheiro sonha coisas claras:

superfícies, tênis, um copo de água.

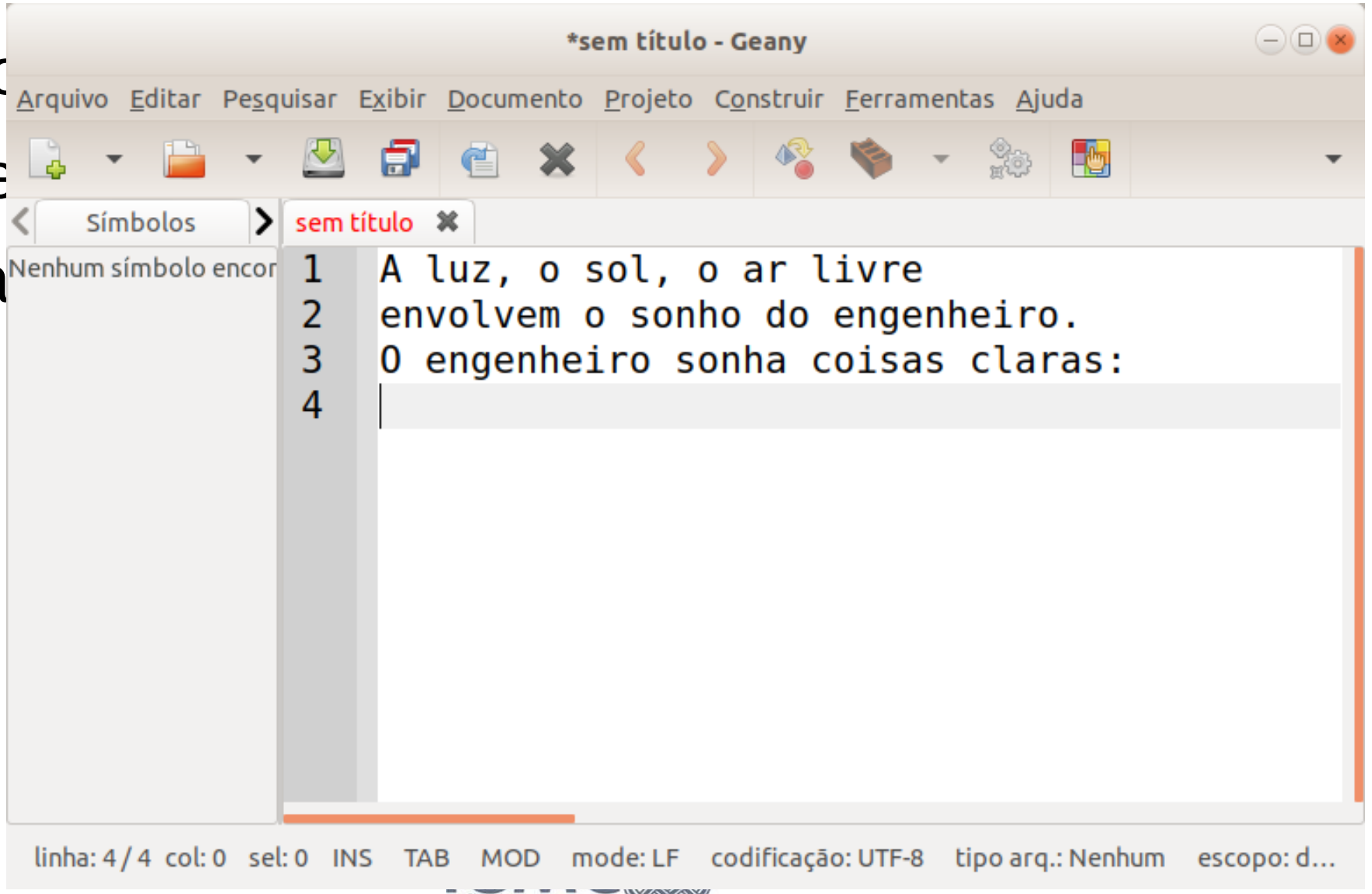
A luz, o sol, o ar livre\nenvolvem o sonho do engenheiro.\nO eng...

Como criar/usar

- Bloco de notas
- Geany
- Salvar, em geral usamos a extensão “.txt”

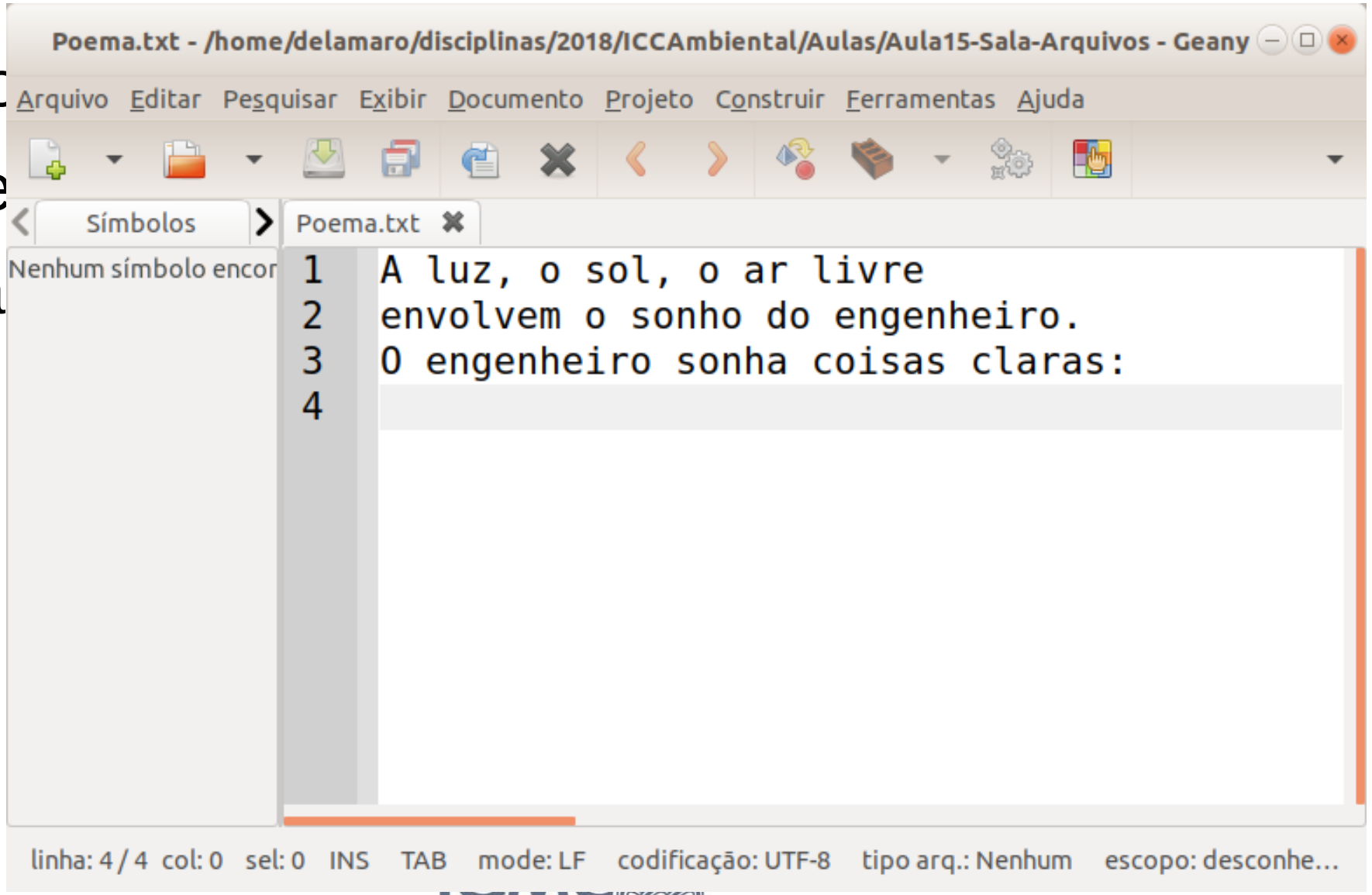
Como criar/usar

- Blo
- Ge
- Sa



Como criar/usar

- Blo
- Ge
- Sa



Vamos ao que interessa

- Dentro dos nossos programas podemos acessar dados que estão nos arquivos
- Para isso, existem funções
- Inicialmente, precisamos dizer qual arquivo queremos usar

Abrir um arquivo

```
f = open('Poema.txt')
```

Abre o arquivo Poema.txt
Localizado na mesma pasta que o meu programa

A partir daí, a variável **f** representa esse arquivo. Tudo que for feito na variável **f** será feito no arquivo.

Abrir um arquivo

```
f = open('Poema.txt')
```

Abre o arquivo Poema.txt
Localizado na mesma pasta que o meu programa

A partir daí, a variável **f** representa esse arquivo. Tudo que for feito na variável **f** será feito no arquivo.

```
f = open('c:\Users\Eng\Documentos\Poema.txt')
```

Abre o arquivo Poema.txt
Localizado na pasta c:\Users\Eng\Documentos\

ICMC USP SÃO CARLOS SSC

Fechar um arquivo

```
f = open('Poema.txt')

...

...

...

f.close()
```

Abre o arquivo Poema.txt
Localizado na mesma pasta que o meu programa

A partir daí, a variável **f** representa esse arquivo. Tudo que for feito na variável **f** será feito no arquivo.

Fecha o arquivo **f.** Isso significa que a partir daí, a variável não aponta mais para esse arquivo.
Seu uso é muito importante!

ICMC USP SÃO CARLOS SSC

Ler dados de um arquivo

- Para isso existe função read()
- Parâmetro é uma variável do tipo arquivo.
- Retorna o conteúdo do arquivo na forma de um string

Ler dados

```
f = open('Poema.txt')
texto = f.read()
print(texto)
f.close()
```

Ler dados

```
f = open('Poema.txt')
texto = f.read()
print(texto)
f.close()
```

```
A luz, o sol, o ar livre
envolvem o sonho do engenheiro.
O engenheiro sonha coisas claras:
```

Quebrando o string

- Strings possuem uma função muito útil
- Serve para quebrar o string em “palavras”
- As palavras são armazenadas em uma lista

```
f = open('Poema.txt')
texto = f.read()
lista = texto.split()
print(lista)
f.close()
```

Quebrando o string

- Strings possuem uma função muito útil
- Serve para quebrar o string em “palavras”
- As palavras são armazenadas em uma lista

```
f = open('Poema.txt')
texto = f.read()
lista = texto.split()
print(lista)
f.close()
```

['A', 'luz,', 'o', 'sol,', 'o', 'ar', 'livre', 'envolvem', 'o', 'sonho', 'do', 'engenheiro.', 'O', 'engenheiro', 'sonha', 'coisas', 'claras:']

Exemplo

- Vamos supor que temos, em um arquivo, “dados.txt”, os dados sobre o qual queremos computar as nossas estatísticas
- Vamos definir uma função le_dados que recebe como parâmetro o nome do arquivo e devolve a lista de floats

```
12.8   5.4   3.3
22.7  4.4
27.11  33.09  22.0  13.5     29.4
```

`le_dados`: como era

```
def le_dados():
    dados = []
    r = 0
    i = 1
    while r >= 0:
        r = float(input('Digite o valor {}: '.format(i)))
        if r < 0:
            print('Entrada de dados terminou')
        else:
            dados.append(r)
        i += 1
```

le_dados: como fica

```
def le_dados(arquivo):
    f = open(arquivo)
    s = f.read()
    lista = s.split()
    f.close()
    return lista


l = le_dados('dados.txt')
print l
```

le_dados: como fica

```
def le_dados(arquivo):
    f = open(arquivo)
    s = f.read()
    lista = s.split()
    f.close()
    return lista


l = le_dados('dados.txt')

print l
```

`le_dados`: como fica

```
def le_dados(arquivo):
    f = open(arquivo)
    s = f.read()
    lista = s.split()
    f.close()
    return lista


l = le_dados('dados.txt')

print l
```

```
12.8   5.4   3.3\n22.7  4.4\n27.11
33.09  22.0  13.5    29.4
```

`le_dados`: como fica

```
def le_dados(arquivo):
    f = open(arquivo)
    s = f.read()
    lista = s.split()
    f.close()
    return lista


l = le_dados('dados.txt')

print l
```

```
12.8   5.4   3.3\n22.7  4.4\n27.11
33.09  22.0  13.5    29.4
```

['12.8', '5.4', '3.3', '22.7', '4.4', '27.11', '33.09', '22.0', '13.5', '29.4']

le_dados: como fica

```
def le_dados(arquivo):
    f = open(arquivo)
    s = f.read()
    lista = s.split()
    f.close()
    return lista


l = le_dados('dados.txt')

print l
```

```
12.8   5.4   3.3\n22.7  4.4\n27.11
33.09  22.0  13.5    29.4
```

```
['12.8', '5.4', '3.3', '22.7', '4.4',
'27.11', '33.09', '22.0', '13.5', '29.4']
```

le_dados: como fica

```
def le_dados(arquivo):
    f = open(arquivo)
    s = f.read()
    lista = s.split()
    f.close()
    return lista


l = le_dados('dados.txt')
print l
```

```
12.8   5.4   3.3\n22.7  4.4\n27.11
33.09  22.0  13.5    29.4
```

```
['12.8', '5.4', '3.3', '22.7', '4.4',
'27.11', '33.09', '22.0', '13.5', '29.4']
```

le_dados: como fica

```
def le_dados(arquivo):
    f = open(arquivo)
    s = f.read()
    lista = s.split()
    f.close()
    lista2 = []
    for c in lista:
        lista2.append(float(c))
    return lista2
```

`le_dados`: como fica

```
def le_dados(arquivo):
    f = open(arquivo)
    s = f.read()
    lista = s.split()
    f.close()
    lista2 = []
    for c in lista:
        lista2.append(float(c))
    return lista2
```

Cria uma nova lista, em que cada elemento
é um float, não um string

Exercício

- Escreva um programa que lê uma sequência de números inteiros de um arquivo “numeros.txt” e diz quantos são ímpares e quantos são pares.

Exercício

```
f = open('numeros.txt')
l = f.read().split()
impar = 0
par = 0
for c in l:
t = int(c)
if t % 2 == 0:
    par += 1
else:
    impar += 1
print('Impares: {} Pares: {}'.format(impar,par))
f.close()
```

Arquivos muito grandes

- Se tivermos arquivos muito grandes
- Se tivermos arquivos organizados por linhas
- É possível ler linha por linha de uma arquivo

- Voltamos ao arquivo “Poema.txt”
- Mostrar uma linha por vez

Linha a linha

```
f = open('Poema.txt')
linha = f.readline()
cont = 0
while len(linha) > 0:
    cont += 1
    print('Linha {}: {}'.format(cont, linha))
    linha = f.readline()

f.close()
```

Linha a linha

```
f = open('Poema.txt')
linha = f.readline()
cont = 0
while len(li
    cont +=
    print('L                                    a))
    linha =

f.close()
```

Linha 1: A luz, o sol, o ar livre

Linha 2: envolvem o sonho do engenheiro.

Linha 3: O engenheiro sonha coisas claras:

Linha a linha: comando for

```
f = open('Poema.txt')
cont = 0
for texto in f:
    cont += 1
    print('Linha {}: {}'.format(cont, texto))

f.close()
```

O resultado é o mas o `for` se encarrega de atribuir à variável `texto` cada linha do arquivo

Escrevendo

- É possível, também, escrever dados em um arquivo
- Comando `write(s)` – `s` tem que ser do tipo string
- Além disso, é preciso “avisar” que o arquivo vai ser escrito, e não lido
- Isso é feito no comando `open`
- Vamos escrever um poema em um arquivo

Escrevendo

```
f = open('Poema2.txt', 'w')
f.write('A luz, o sol, o ar livre\n')
f.write('envolvem o sonho do engenheiro.\n')
f.write('O engenheiro sonha coisas claras:')
f.close()
```

Escrevendo

```
f = open('Poema2.txt', 'w')
f.write('A luz, o sol, o ar livre\n')
f.write('envolvem o sonho do engenheiro.\n')
f.write('O engenheiro sonha coisas claras:')
f.close()
```

Modo de abrir o arquivo. Quer dizer que o arquivo vai ser criado e os dados serão escritos

Escrevendo

```
f = open('Poema2.txt', 'w')
f.write('A luz, o sol, o ar livre\n')
f.write('envolvem o sonho do engenheiro.\n')
f.write('O engenheiro sonha coisas claras:')
f.close()
```

Modo de abrir o arquivo. Quer dizer que o arquivo vai ser criado e os dados serão escritos

Comando write é parecido com o print. Mas só aceita string como parâmetro. E não coloca automaticamente um fim de linha.

Escrevendo

```
f = open('Poema2.txt', 'w')
f.write('A luz, o sol, o ar livre\n')
f.write('envolvem o sonho do engenheiro.\n')
f.write('O engenheiro sonha coisas claras:')
f.close()
```

Modo de abrir o arquivo. Quer dizer que o arquivo vai ser criado e os dados serão escritos

Comando write é parecido com o print. Mas só aceita string como parâmetro. E não coloca automaticamente um fim de linha. Poderíamos ter usado um único write para escrever tudo de uma vez.

ICMC USP SÃO CARLOS SSC

Exercício

- Escreva um programa que leia várias frase ou strings do usuário (pelo teclado) e escreva cada uma delas em um arquivo chamado “frases.txt”. Cada frase digitada deve estar em uma linha separada no arquivo. A entrada de dados termina quando o usuário digitar apenas <enter>.

Exercício

```
f = open('frases.txt', 'w')
while True:
    s = input()
    if len(s) == 0:
        break
    f.write('{}\n'.format(s))
f.close()
```

Exercício

```
f = open('frases.txt', 'w')
while True:
    s = input()
    if len(s) == 0:
        break
    f.write('{}\n'.format(s))
f.close()
```

Lê o string do teclado

Escreve o string no arquivo

Modos de abertura mais usados

- Abrir para leitura (se o arquivo não existe, ocorre um erro)
  - open(‘nome arquivo’)
  - open(‘nome arquivo’, ‘r’)
- Abrir para escrever (cria o arquivo vazio. Se arquivo já existe, apaga o antigo)
  - open(‘nome arquivo’, ‘w’)

Modos de abertura mais usados

- Abrir para escrever e ler (se o arquivo não existe, cria o arquivo vazio. Se existe, vai escrevendo por cima do conteúdo antigo)
  - open(‘nome arquivo’, ‘r+’)

- Abrir para escrever (se o arquivo não existe, cria o arquivo vazio. Se existe, vai escrevendo no fim do arquivo antigo)
  - open(‘nome arquivo’, ‘a’)

Exercício

- Escreva um programa que leia uma sequência de números inteiros de um arquivo “numeros.txt” e escrevaos números pares em um arquivo “par.txt” e os impares em um arquivo “impar.txt”.

Exercício

```
f = open('numeros.txt')
l = f.read().split()
f_impar = open('impar.txt', 'w')
f_par = open('par.txt', 'w')
for c in l:
    t = int(c)
    if t % 2 == 0:
        f_par.write('{} '.format(c))
    else:
        f_impar.write('{} '.format(c))
```

```
f.close()
f_impar.close()
f_par.close()
```

ICMC USP SÃO CARLOS SSC

Exercício

- O que acontece, no exercício anterior, se substituirmos o modo ‘w’ pelo modo ‘a’?
- Execute varias vezes depois de fazer a mudança e verifique o resultado.