

MC-102 – Aula 25

Arquivos Binários

Instituto de Computação – Unicamp

23 de Maio de 2018

Motivação

- Vimos que existem dois tipos de arquivos: textos e binários.
- Objetos (como **int**, **float**, **list**, etc.) na sua representação em binário, ocupam pouco espaço na memória, quando comparado com sua representação em formato texto.
 - ▶ Se representássemos 123456789.00 na forma textual, gastaríamos 12 bytes para representar este número.
 - ▶ Mas sua representação binária ocupa sempre 64 bits (ou 8 bytes).
- Armazenar dados em arquivos de forma análoga a utilizada em memória permite:
 - ▶ Reduzir o tamanho do arquivo.
 - ▶ Guardar estruturas complicadas tendo acesso simples.

Abrindo um Arquivo Binário

- Assim como em arquivos texto, devemos abrir o arquivo com a função **open** e atribuir o objeto arquivo resultante para uma variável.
- Desta forma a variável estará associada ao arquivo, com métodos para leitura e escrita sobre este.

```
arq = open(nome_do_arquivo , modo)
```

Abrindo um Arquivo Binário

```
arq = open(nome_do_arquivo , modo)
```

Modos de abertura de arquivo binário

modo	operações
rb	leitura
wb	escrita
r+b	leitura e escrita

- Se um arquivo for aberto para leitura (**rb**) e não existir a função gera uma exceção.
- Se um arquivo for aberto para escrita (**wb**) e não existir um novo arquivo é criado. Se ele existir, é sobre-escrito.
- Se um arquivo for aberto para leitura/gravação (**r+b**) e existir ele NÃO é sobreescrito. Se o arquivo não existir a função gera uma exceção.

Lendo e Escrevendo no Arquivo

Utilizaremos o módulo **pickle** para ler e escrever objetos para um arquivo.

- Primeiramente deve-se importar este módulo:

```
import pickle
```

- Após isso podemos escrever um objeto em arquivo com o método **pickle.dump** e podemos ler um objeto em arquivo com o método **pickle.load**.

Lendo e Escrevendo no Arquivo

Para escrever um objeto em um arquivo binário usamos o método **pickle.dump**.

```
pickle.dump(objeto, var_arquivo)
```

- **objeto**: este é o objeto a ser salvo em arquivo.
- **var_arquivo**: esta é a variável associada a um arquivo previamente aberto em modo binário.

O programa abaixo salva uma lista em arquivo:

```
import pickle
try:
    arq = open("teste.bin", "wb")
    l = [1, 2, 3]
    pickle.dump(l, arq)
    arq.close()
except:
    print("Problemas com o arquivo teste.bin")
```

Lendo e Escrevendo no Arquivo

Para ler um objeto de um arquivo binário usamos o método **pickle.load**.

```
var_objeto = pickle.load(var_arquivo)
```

- **var_arquivo**: esta é a variável associada a um arquivo previamente aberto em modo binário.
- O método automaticamente reconhece o tipo de objeto salvo em arquivo, carrega este para a memória e atribui para a variável **var_objeto**.

O programa abaixo lê a lista previamente salva em arquivo:

```
import pickle
try:
    arq = open("teste.bin", "rb")
    l = pickle.load(arq)
    print(l)
    arq.close()
except:
    print("Problemas com o arquivo teste.bin")
```