

Computing Fundamentals

Input/Output

Salvatore Filippone

`salvatore.filippone@uniroma2.it`

2014–2015

Files (normally) are collections of bytes that live in *permanent* storage. Octave/MATLAB provide means to access these data collections:

- 1 Simple save, load for the workspace;
- 2 High-level I/O functions;
- 3 Low-level I/O functions.

You can save and restore the contents of the workspace (n.b.: *the data*)

```
>> save 'mydata.oct'
```

```
>> load 'testdata.oct'
```

It is also possible to save selected variables from the workspace

```
>> save 'mydata.oct' a b c*
```

There exist some high-level I/O functions, not all of them are also available in Octave:

- * `textscan` – `fprintf`
- * `csvread` – `csvwrite`
- * `dlmread` – `dlmwrite`
- `xlsread` – `xlswrite`
- `hdfread`
- `xmlread` – `xmlwrite`
- * `imread` – `imwrite`
- * `wavread` – `wavwrite`
- `aviread`

Do it yourself: low-level file interface.

Open/close: create a *file handle* for subsequent operations

```
fh= fopen(<filename >, <purpose >)
```

```
then close when done fclose(fh)
```

Read a line: of text `srt=fgets(fh)`

Tokenize: that is, split in pieces

```
[tok , rest]=strtok(line)
```

Parse a line: of text, potentially multiple lines

```
ca = textscan(fh, <format>)
```

Example: write a file to output

```
fn=input('filename: ','s');  
fh=fopen(fn,'r');  
ln=fgets(fh)  
while ischar(ln)  
    fprintf('%s',ln);  
    ln = fgets(fh);  
end  
fclose(fh);
```

Example: read from file a student database