

INTRODUZIONE AD EMACS

LEONARDO TAMIANO

Created: 2023-12-31 Sun 11:42

TABLE OF CONTENTS

- Che cos'è Emacs?
- Come utilizzo Emacs
 - Interfacciamento OS
 - Programmazione
 - Org-Mode
- Perché imparare Emacs?
 - Produttività
 - Introduzione a LISP
 - Crescita personale

CHE COS'È EMACS?

La prima versione pubblica di **GNU Emacs** (versione 13) fu rilasciata il 20 marzo del 1985 come il primo software del **GNU Project**, iniziativa fondata da **Richard Stallman** nel 1978.



A prima vista Emacs può essere descritto come un "text editor molto potente". Emacs infatti nasce per gestire contenuti testuali di ogni natura.

Questo è un file di testo.

Emacs permette di modificare e gestire i file di testo.

Ma Emacs non si limita ad essere un text editor.■

test.txt 5:48
Wrote /home/leo/test.txt

11:46 LF UTF-8 Text (+5)

Emacs contiene al suo interno un interprete per il linguaggio **elisp** (emacs-lisp).

Codice elisp che calcola l' n esimo numero di Fibonacci.

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

```
(defun fib (n)
  (cond ((eq n 1) 1)
          ((eq n 2) 1)
          (t
            (+ (fib (- n 1))
                (fib (- n 2)))))))
```

$$\begin{aligned} F_1 &= F_2 = 1 \\ F_n &= F_{n-2} + F_{n-1}, \quad n \geq 3 \end{aligned}$$

Questo è un file di testo.

Emacs permette di modificare e gestire i file di testo.

Ma Emacs non si limita ad essere un text editor.

Tramite Emacs siamo infatti in grado di eseguire codice elisp.

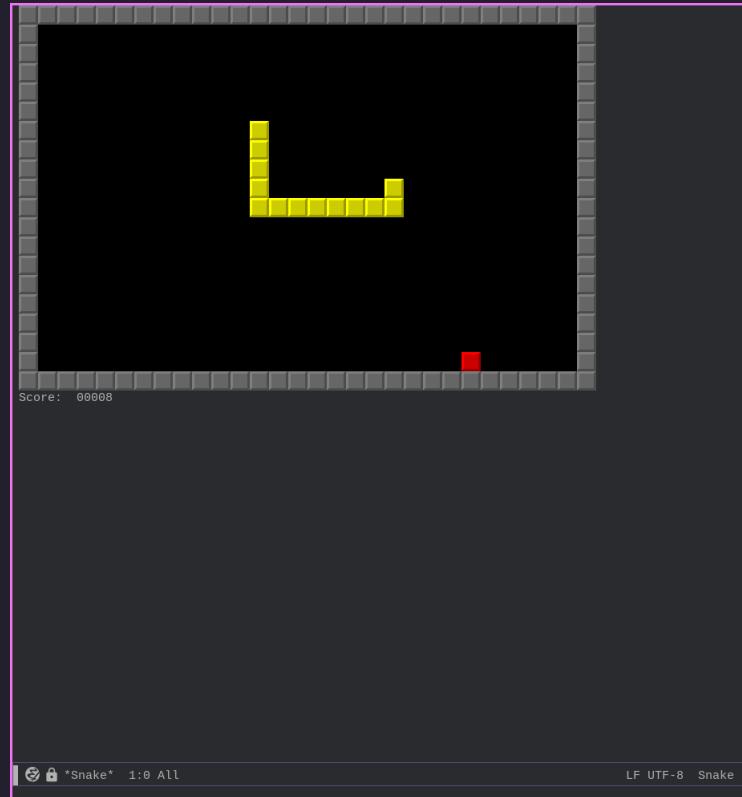
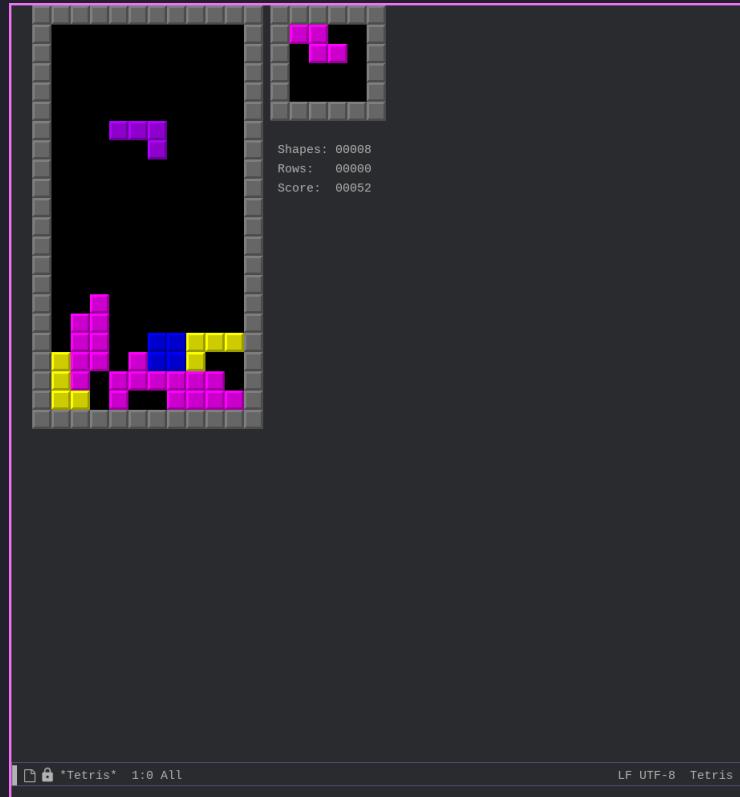
```
#+begin_src elisp
(defun fib (n)
  (cond ((eq n 1) 1)
        ((eq n 2) 1)
        (t
         (+ (fib (- n 1))
            (fib (- n 2))))))

(fib 10)
#+end_src
```

#+RESULTS:

: 55

È proprio l'abilità di eseguire codice elisp (linguaggio Turing-Completo) ciò che rende Emacs estremamente flessibile e potente.

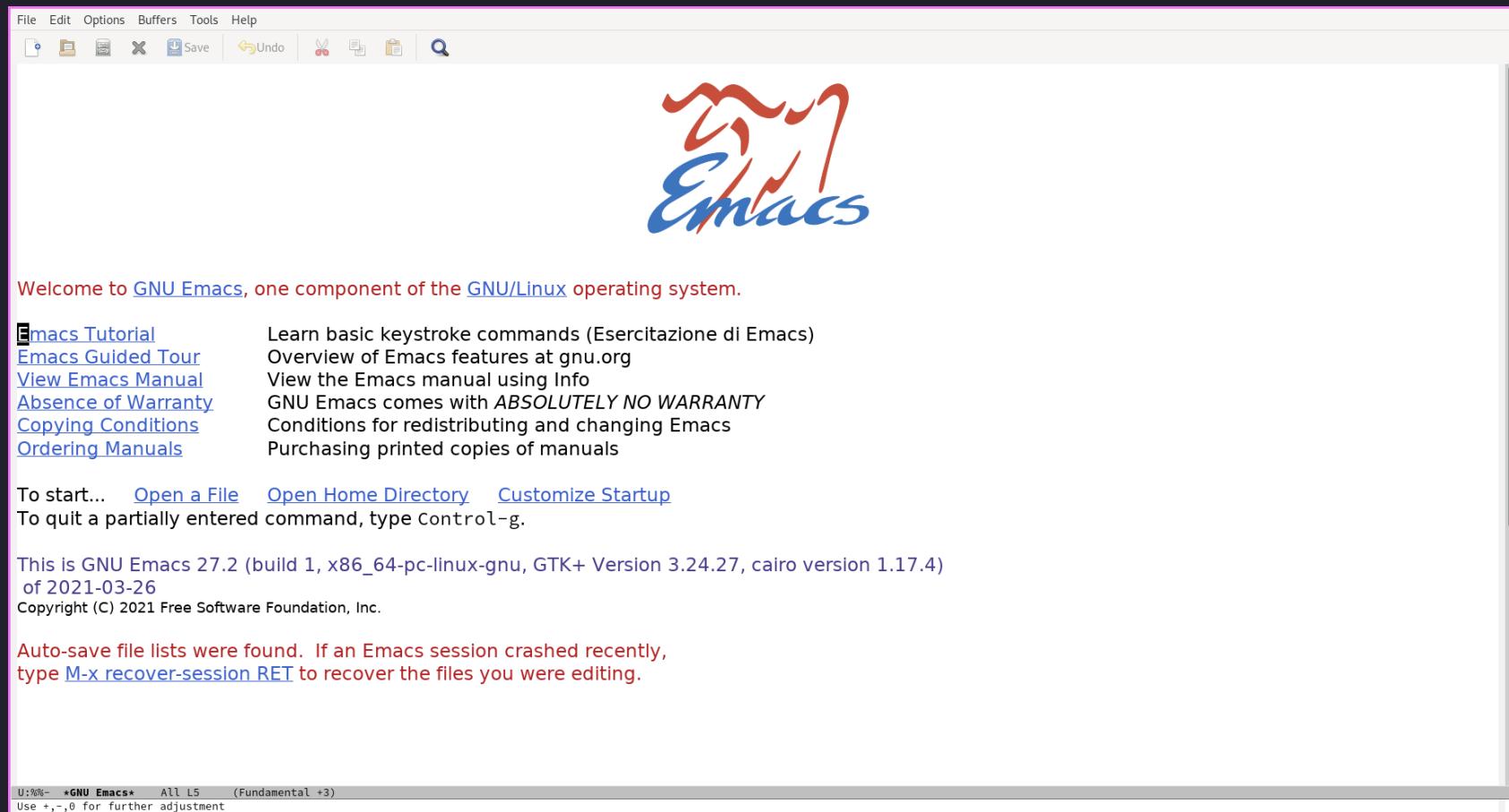


Imparare (bene) Emacs significa principalmente capire come **configurarolo** in modo da ottenere un comportamento che si avvicini il più possibile a quello desiderato.

Per fare questo possiamo sia scrivere noi del codice elisp, che utilizzare del codice elisp scritto da altre persone e disponibile in archivi pubblici, come ad esempio l'archivio **MELPA**.

COME UTILIZZO EMACS

Per quanto Emacs sia un tool estremamente potente, la configurazione iniziale lascia molto a desiderare, sia in termini di estetica che di funzionalità.



Investendo abbastanza tempo ed impegno però è possibile trasformarlo in ciò che si vuole.

File Edit Options Buffers Tools Table Org Text Help

Save Undo

#+TITLE: **Emacs Configuration**

#+AUTHOR: Leonardo Tamiano

#+PROPERTY: header-args :tangle dotemacs.el

* Startup...
* Packages...
* General...
* Hooks...
* Indentation...
* Backups...
* Files Exts...
* Colored Texts...
* Tabs...
* Keybinds...
* My functions... █

→

#+TITLE: **Emacs Configuration**

#+AUTHOR: Leonardo Tamiano

#+PROPERTY: header-args :tangle dotemacs.el

+ Startup...
+ Packages...
+ General...
+ Hooks...
+ Indentation...
+ Backups...
+ Files Exts...
+ Colored Texts...
+ Tabs...
+ Keybinds...
+ My functions... █

U:--- dotemacs.org All L1541 Git-master (Org +2)
Quit

knwl/dotfiles/dotemacs.org 1541:0 10:12 LF UTF-8 Org (+2) mas
Use + - ⌘ for further adjustment

Andiamo adesso a vedere alcuni (pochi) use-case personali di utilizzo di Emacs.

INTERFACCIAMENTO OS

TERMINAL EMULATOR

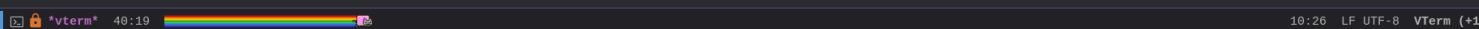
Emacs offre le tipiche funzionalità di un terminal emulator tramite vari pacchetti (**vterm**, **term-mode**, **shell-mode**, **eshell**).

```
              _` 
              `0` 
              `00` 
              `+000: 
              `+0000: 
              `+00000: 
              `+000000: 
              `/:-`+0000+: 
              `/+++++=====+` 
              `/+++++=====+` 
              `/++000000000000/` 
              `/000SSSS0++0SSSSS0+` 
              .0SSSSSS0-` ``/0SSSSSS+` 
              -0SSSSSS0. :SSSSSS0. 
              :0SSSSSS. OSSSS0+++. 
              /0SSSSSSS/ +SSSS000/-` 
              /0SSSSSS0+/-` -:/+0SSSS0+-` 
              `+SS0+:-`      `-/+0S0: 
              `++`          `-/+/` 
              `.` 

leo@archlinux 
----- 
OS: Arch Linux x86_64 
Kernel: 5.10.32-1-lts 
Uptime: 1 hour, 8 mins 
Packages: 1404 (pacman) 
Shell: bash 5.1.4 
Resolution: 1920x1080, 1366x768 
WM: i3 
Theme: Adwaiata [GTK2/3] 
Icons: Adwaiata [GTK2/3] 
Terminal: emacs 
CPU: Intel i7-2600K (8) @ 3.800GHz 
GPU: NVIDIA GeForce GTX 970 
Memory: 1980MiB / 7931MiB 

PREPARING TO SYNC... 
===== 
Syncing Knwl and Dropbox/org_files 
===== 
Unison 2.51.2 (ocaml 4.11.0): Contacting server... 
Looking for changes 
Reconciling changes 
Nothing to do: replicas have not changed since last sync. 
===== 
Syncing public and Dropbox/org_files 
===== 
Unison 2.51.2 (ocaml 4.11.0): Contacting server... 
Looking for changes 
Reconciling changes 
Nothing to do: replicas have not changed since last sync. 
===== 
DONE SYNCING 
[leo@archlinux ~]$ whoami 
leo 
[leo@archlinux ~]$ 


```

FILE EXPLORER

Tramite **dired** invece è possibile spostare, creare, modificare, eliminare e rinominare file e cartelle.

```
/home/leo/repos/TBD/test:  
total used in directorye 44 available 334.3 GiB  
drwxr-xr-x 11 leo users 4096 24 apr 10.52 .  
drwxr-xr-x  8 leo users 4096 24 apr 10.52 ..  
drwxr-xr-x  2 leo users 4096 24 apr 10.52 dir_junk_1  
D drwxr-xr-x  2 leo users 4096 24 apr 10.52 dir_junk_2  
drwxr-xr-x  2 leo users 4096 24 apr 10.52 dir_junk_3  
* drwxr-xr-x  2 leo users 4096 24 apr 10.52 dir_junk_4  
drwxr-xr-x  2 leo users 4096 24 apr 10.52 dir_junk_5  
drwxr-xr-x  2 leo users 4096 24 apr 10.52 dir_junk_6  
drwxr-xr-x  2 leo users 4096 24 apr 10.52 dir_junk_7  
drwxr-xr-x  2 leo users 4096 24 apr 10.52 dir_junk_8  
drwxr-xr-x  2 leo users 4096 24 apr 10.52 dir_junk_9
```

EMAIL MANAGER

È anche possibile gestire le email tramite vari pacchetti
(**mu4e**, **gnus**, **notmuch**).

```
* mu4e - mu for emacs version 1.4.5

Basics

* [j]ump to some maildir
* enter a [s]earch query
* [C]ompose a new message

Bookmarks

* [bu] Unread messages      (0/0)
* [bt] Today's messages     (0/0)
* [bw] Last 7 days
* [bp] Messages with images (0/414)

Misc

* [;]Switch context
* [U]pdate email & database

* [N]ews
* [A]bout mu4e
* [H]elp
* [q]uit

Info

* database-path        : /home/leo/.cache/mu/xapian
* maildir                : /home/leo/Maildir
* in store              : 7824 messages
* personal addresses    : leonardotamiano95@gmail.com, leo95.shop@gmail.com, leo95.game@gmail.com, leo95.dev@gmail.com,
leotwork@protonmail.com, leonardo.tamiano@alumni.uniroma2.eu
```

Per chi fosse interessato a come configurare mu4e, è
presente un video sul canale.



Come gestire le emails in Emacs (Mu4e)

PROGRAMMAZIONE

La flessibilità di Emacs offre svariate occasioni per supportare e facilitare le tipiche task svolte da un programmatore.

GIT INTERFACE

Il pacchetto **magit**, ad esempio, offre una comoda ed intuitiva interfaccia a **git**, il famoso software di versionamento distribuito.

```
Head:    master Update
Push:    local/master Update

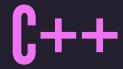
Untracked files (13)
Unstaged changes (12)
modified  dotfiles/.compton.conf
Modified  dotfiles/.i3blocks.conf
@@ -5,6 +5,10 @@ separator=true
separator_block_width=15
markup=pango

+[covid19]
+label=VAC
+interval=once
+
# Disk usage
#
# The directory defaults to $HOME if the instance is not specified.
modified  dotfiles/emacs_bookmarks
modified  languages/bash/ppa
modified  languages/latex/cv/auto/cv.el
modified  languages/latex/cv/cv.log
modified  languages/latex/cv/cv.pdf
modified  languages/latex/cv/cv.tex
modified  languages/python/utils/index_generator.py
modified  notes/languages.org
modified  notes/os.org
modified  notes/tools.org

Recent commits


```

◆ 🔒 magit: knwl 30:0 🎨 10:41 LF UTF-8 Magit (+2)



Ogni linguaggio di programmazione poi ha la sua particolare modalità.

```
1 #include <iostream>
2 #include <experimental/filesystem>
3 #include <fstream>
4 #include <string>
5 #include <vector>
6 #include <unordered_map>
7 #include <htmlcxx/html/ParserDom.h>
8 #include <algorithm>
9 #include <locale>
10 #include <codecvt>
11
12 using namespace std;
13 namespace fs = std::experimental::filesystem;
14
15 // -----
16 // Filesystem library
17
18 string getFileExtFromString(const string &s) {
19     size_t i = s.rfind('.', s.length());
20     if (i != string::npos) {
21         return(s.substr(i+1, s.length() - i));
22     }
23 }
24     return("");
25 }
26
27 string getFileExtensionFromDirEntry(const fs::directory_entry &d) {
28     return (d.path().extension());
29 }
30
31 string getFilenameFromDirEntry(const fs::directory_entry &d) {
32     return (d.path().filename());
33 }
34
35 // -----
36 // htmlcxx library
37
38 void walk_tree(tree<htmlcxx::HTML::Node> const &dom, stringstream &output, bool print=true) {
39     tree<htmlcxx::HTML::Node>::iterator it = dom.begin();
40
41     if (print && (!it->isTag()) && (!it->isComment())) {
42         output << it->text() << "\n";
43     }
44
45     for (unsigned i = 0; i < dom.number_of_children(it); i++) {
46         bool print_child = (it->isTag()) &&
47             ((it->tagName() == "p") ||
48              (it->tagName() == "i") ||
49              (it->tagName() == "a") ||
50              (it->tagName() == "span") ||
51              (it->tagName() == "b"));
52         walk_tree(dom.child(it, i), output, print_child);
53     }
54 }
55
56 // -----
57 // inverted index
```

```
1 pse/src/indexer.cpp 23:0 10:43 LF UTF-8 C++/l master ✓
Compile command: g++ -lhtmlcxx -lcss_parser_pp -lcss_parser indexer.cpp -lstdc++fs -o indexer
```

```
* mode: compilation; default-directory: "~/repos/pse/src/" -*-
Compilation started at Sat Apr 24 10:43:02
g++ -lhtmlcxx -lcss_parser_pp -lcss_parser indexer.cpp -lstdc++fs -o indexer
Compilation finished at Sat Apr 24 10:43:04
```

```
2 * compilation* 1:0 All
[leo@archlinux src]$ ls
indexer indexer.cpp js main.cpp out.txt
[leo@archlinux src]$ ls
indexer indexer.cpp js main.cpp out.txt
[leo@archlinux src]$ ./indexer > out.txt
[leo@archlinux src]$
```

```
[1] 1:bash*
```

```
3 * vterm* 6:21 All
```

```
LF UTF-8 VTerm (+1)
```

PYTHON (JUPYTER NOTEBOOKS)

Con l'avvento di nuove tecnologie è possibile scrivere del nuovo codice elisp per lavorare con queste nuove tecnologie sempre all'interno di Emacs.

The screenshot shows a Jupyter Notebook interface running in a terminal window. The left pane displays a list of notebooks:

```
Contents API 6.3.0 (http://127.0.0.1:8888)
| [Home] |
[New Notebook] [Resync] [Open In Browser]
Create New Notebooks Using Kernel:
(*) Python 3

----- All Opened Notebooks -----

Sort by: Name
In Order: Ascending

[Open] ----- [Delete] : 01_mnist_keras.ipynb          22/09/2020
[Open] ----- [Delete] : 02_loss.ipynb                 22/09/2020
[Open] ----- [Delete] : 03_overfitting.ipynb          22/09/2020
[Open] ----- [Delete] : 04_BetaBernoulli.ipynb        22/09/2020
[Open] ----- [Delete] : 05_regression.ipynb           22/09/2020
[Open] ----- [Delete] : 06_naivebayes.ipynb          22/09/2020
[Open] ----- [Delete] : 07_biasvar.ipynb              22/09/2020
[Open] ----- [Delete] : 08_linregr_samples.ipynb       22/09/2020
[Open] ----- [Delete] : 09_equiv_kern.ipynb            22/09/2020
[Open] ----- [Delete] : 10_equiv_predict.ipynb         22/09/2020
[Open] ----- [Delete] : 11_titanic.ipynb              22/09/2020
[Open] ----- [Delete] : 12_class_regr.ipynb           22/09/2020
[Open] ----- [Delete] : 13_lda.ipynb                  22/09/2020
[Open] ----- [Delete] : 14_perceptron.ipynb           22/09/2020
[Open] ----- [Delete] : 15_gda.ipynb                 22/09/2020
[Open] ----- [Delete] : 16_logistic.ipynb             22/09/2020
[Open] ----- [Delete] : 17_gda.lin-sk-cv.ipynb        22/09/2020
[Open] ----- [Delete] : 18_gda-sk-cv.ipynb             22/09/2020
[Open] ----- [Delete] : 19_logreg.ipynb               22/09/2020
[Open] ----- [Delete] : 20_logregrebf.ipynb            22/09/2020
[Open] ----- [Delete] : 21_linregr_gradient.ipynb      22/09/2020
[Open] ----- [Delete] : 22_iris.ipynb                 22/09/2020
[Open] ----- [Delete] : 23_softmax.ipynb              22/09/2020
[Open] ----- [Delete] : 24_MCMC.ipynb                22/09/2020
[Open] ----- [Delete] : 25_linregr_mc.ipynb           22/09/2020
[Open] ----- [Delete] : 26_ROC.ipynb                 22/09/2020
[Open] ----- [Delete] : 27_parzen.ipynb              22/09/2020
[Open] ----- [Delete] : 28_knn.ipynb                 22/09/2020
[Open] ----- [Delete] : 29_local_regr.ipynb           22/09/2020
[Open] ----- [Delete] : 30_gp.ipynb                  22/09/2020
[Open] ----- [Delete] : 31_svm_xor.ipynb              22/09/2020
[Open] ----- [Delete] : 32_gaussian_processes.ipynb    22/09/2020
[Open] ----- [Delete] : 33_textclassification.ipynb     22/09/2020
[Open] ----- [Delete] : 34_svm_custom_kernel.ipynb      22/09/2020
[Open] ----- [Delete] : 35_bp_bp.ipynb                22/09/2020
[Open] ----- [Delete] : 36_bp_bp.ipynb                22/09/2020
[Open] ----- [Delete] : 37_lstm.ipynb                 22/09/2020
[Open] ----- [Delete] : 38_ensemble.ipynb              22/09/2020
[Open] ----- [Delete] : 39_pca.ipynb                 22/09/2020
```

The right pane shows code cells and plots:

```
In [0]: %A [+] [python3]
50 plt.subplot(10,10,1+1)
51 plt.imshow([(255-x for x in X[img_idx[i]]], cmap='gray', interpolation='gaussian')
52 if class_value:
53     plt.title("%d".format(t[img_idx[i]]), fontsize = 16, color='b')
54 plt.gca().xaxis.set_major_locator(plt.NullLocator())
55 plt.gca().yaxis.set_major_locator(plt.NullLocator())
56 plt.axis('off')
57 plt.subplots_adjust(top=1)
58 plt.show()
59
60
61 In [0]:
62 def plotData(X, Y, c, npxel=28):
63     m, n = X.shape
64     image = np.array(X[c,:])
65     plt.figure(figsize=(6,6))
66     plt.imshow(image.reshape(npxel, npxel)), cmap='Greys', interpolation='quadratic'
67     plt.show()
68 In [0]:
69 def plotAccuracy(acc_history_train, acc_history_test):
70     plt.figure(figsize = (12,8))
71     plt.plot(acc_history_train, marker='o', markersize=5, label='Train')
72     plt.plot(acc_history_test, marker='o', markersize=5, label='Test')
73     plt.legend()
74     plt.gca().xaxis.set_major_locator(plt.NullLocator())
75     plt.show()
76
77 In [0]:
78 def save_model(m,filename):
79     model_json = m.to_json()
80     with open("/drive/My Drive/colab_data/"+filename+".json", "w") as json_file:
81         json_file.write(model_json)
82     # serialize weights to HDF5
83     m.save_weights("/drive/My Drive/colab_data/"+filename+".h5")
84     print("Saved model to disk!")
85
86 In [0]:
87 def load_model_weights(filename, model):
88     model.load_weights("/drive/My Drive/colab_data/"+filename+".h5")
89     print("Loaded weights from disk")
90     return model
91
92 In [0]:
93 def load_model(filename):
94     json_file = open("/drive/My Drive/colab_data/"+filename+'.json', 'r')
95     loaded_model_json = json_file.read()
96     json_file.close()
97     m = model_from_json(loaded_model_json)
98     # load weights into new model
99     m.load_weights("/drive/My Drive/colab_data/"+filename+".h5")
100    print("Loaded model from disk")
101
102 markdown:
103 Fissa il numero di classi pari a 10 (corrispondenti alle cifre 0,1,...,9) e leggi i dati, suddivisi in training e test set, $(X_{train}, t_{train}), (X_{test}, t_{test})$. Le matrici $X$ rappresentano la immagine mentre i valori $t$ classificano la relativa classe
```

Bottom status bar: 1 ein:notebooklist http://127.0.0.1:8888* 19:3 Top LF UTF-8 ein:notebooklist 2 ein: http://127.0.0.1:8888/01_mnist_keras.ipynb* 67:0 4% LF UTF-8 EIN[Py]

ORG-MODE

Tra tutte le funzionalità di emacs, **org-mode** (secondo me) è senza dubbio la migliore, in quanto esistono svariati use-cases in cui utilizzare file orgs.

I file scritti in org-mode sono semplici file di testo che vengono processati in modo dinamico da Emacs.

```
#+TITLE: Esempio file Org-Mode
#+AUTHOR: Leonardo Tamiano

* Outline 1
  Prova.

** Sub-Outline 1.1
*** Sub-Sub-Outline 1.1.1
*** Sub-Sub-Outline 1.1.2
** Sub-Outline 1.2
*** Sub-Sub-Outline 1.2.1
  Prova prova.

* Outline 2
** Sub-Outline 2.1
** Sub-Outline 2.2
  Prova prova.

*** Sub-Sub-Outline 2.2.1
*** Sub-Sub-Outline 2.2.2
**** Sub-Sub-Sub-Outline 2.2.2.1
```

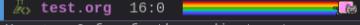
test.org 21:32 All
Use +,-,0 for further adjustment

```
#+TITLE: Esempio file Org-Mode
#+AUTHOR: Leonardo Tamiano

+ Outline 1
  Prova.

+ Sub-outline 1.1
  † Sub-Sub-Outline 1.1.1
  † Sub-Sub-Outline 1.1.2
+ Sub-outline 1.2
  † Sub-Sub-Outline 1.2.1
    Prova prova.

+ Outline 2
  † Sub-Outline 2.1
  †† Sub-Outline 2.2...
```

test.org 16:0 
Use +,-,0 for further adjustment

PERSONAL NOTES

Personalmente ho una collezione di file orgs in cui scrivo di tutto, dalle cose tecniche alle cose non-tecniche.

| | |
|------------------------|---|
| org - agenda | <code>~/repos/public/notes/agenda.org</code> |
| org - anime | <code>~/repos/public/notes/anime.org</code> |
| org - beorg | <code>~/repos/public/notes/beorg.org</code> |
| org - books | <code>~/repos/public/notes/books.org</code> |
| org - finance | <code>~/repos/public/notes/finance.org</code> |
| org - languages | <code>~/repos/knwl/notes/languages.org</code> |
| org - movies | <code>~/repos/public/notes/movies.org</code> |
| org - notes | <code>~/repos/public/notes/notes.org</code> |
| org - os | <code>~/repos/knwl/notes/os.org</code> |
| org - scratch | <code>~/repos/public/notes/scratch.org</code> |
| org - slides | <code>~/repos/yt/archived/2-ox-reveal/content/slides.org</code> |
| org - tools | <code>~/repos/knwl/notes/tools.org</code> |
| org - tv_series | <code>~/repos/public/notes/tv_series.org</code> |
| org - youtube | <code>~/repos/yt/youtube.org</code> |

UNIVERSITY NOTES

Tutti i miei appunti dell'università sono scritti in org-mode.

```
#+TITLE: ML - Machine Learning
#+AUTHOR: Leonardo Tamiano

* Course Info...
* Log lectures (units)
+ Lecture 01 - Introduction to ML...
+ Lecture 02 - Fitting of Polynomial I...
+ Lecture 03 - Probability/Statistics Review...
+ Lecture 04 - Bayesian Statistics...
+ Lecture 05 - Information Theory...
+ Lecture 06 - Model Inference...
+ Lecture 07 - Fitting of Polynomial II...
+ Lecture 08 - Linear Regression...
+ Lecture 09 - Linear Classification I...
+ Lecture 10 - Linear Classification II
+ Lecture 11 - Probabilistic Classification I
+ Lecture 12 - Probabilistic Classification II
+ Lecture 13 - Probabilistic Classification III
+ Lecture 14 - Probabilistic Classification IV
+ Lecture 15 - Montecarlo Methods
+ Lecture 16 - Non Parametric Models
+ Lecture 17 - Kernel Regression
+ Lecture 18 - Gaussian Processes
+ Lecture 19 - Support Vector Machines I
+ Lecture 20 - Support Vector Machines II
+ Lecture 21 - Support Vector Machines III
+ Lecture 22 - Support Vector Machines IV
+ Lecture 23 - Neural Networks I
+ Lecture 24 - Neural Networks II
+ Lecture 25 - Deep Learning I
+ Lecture 26 - Deep Learning II
+ Lecture 27 - Deep Learning III
+ Lecture 28 - Decision Trees
+ Lecture 29 - Ensemble Methods
+ Lecture 30 - Principal Component Analysis
+ Lecture 31 - Singular Value Decomposition
+ Lecture 32 - Clustering I
+ Lecture 33 - Clustering II
+ Lecture 34 - Clustering III
* Log lectures (live)...
* Log Code...
```

```
| ↵ ML/../../../../repos/ml/ml_log.org 68:33 Top
| LF UTF-8 Org ⌂ master
| C-x-
```

```
#+TITLE: AR - Analisi di Reti
#+AUTHOR: Leonardo Tamiano

* Informazioni corso...
* Log lezioni
+ Lezione 01 - Introduzione...
+ Lezione 02 - Chiusura Triadica...
+ Lezione 03 - Partitionamento in Comunità I...
+ Lezione 04 - Partitionamento in Comunità II...
+ Lezione 05 - Stabilità in Reti Segnate I...
+ Lezione 06 - Stabilità in Reti Segnate II...
+ Lezione 07 - Reti di Informazioni I...
+ Lezione 08 - Reti di Informazioni II...
+ Lezione 09 - Cascate Informative I...
+ Lezione 10 - Cascate Informative II...
+ Lezione 11 - Popolarità...
+ Lezione 12 - Processi di Diffusione I...
+ Lezione 13 - Processi di Diffusione II...
+ Lezione 14 - Processi di Diffusione III...
+ Lezione 15 - Ricerca Decentralizzata I...
+ Lezione 16 - Ricerca Decentralizzata II...
+ Lezione 17 - Sistemi di Voto I...
+ Lezione 18 - Sistemi di Voto II...
+ Lezione 19 - Sistemi di Voto III...
```

```
| ↵ AR/../../../../repos/ar/ar_log.org 18:0 All
| LF UTF-8 Org ⌂ master
| CHILDREN
```

BLOG

Il mio blog è gestito interamente da un singolo file org.

```
#+TITLE: Blog Content
#+STARTUP: content
#+AUTHOR: Leonardo Tamano
#+HUGO_BASE_DIR: ../
#+HUGO_AUTO_SET_LASTMOD: t
#+OPTIONS: author:nil ^:nil

+ About...
+ Favorites...
+ Reviews...
+ Posts
:PROPERTIES:...

+ DONE How to manage e-mails in Emacs with Mu4e :tech:emacs:...
+ DONE The Complete Beginners Introduction to Emacs - Part 1/3 - How I use Emacs :tech:emacs:...
+ DONE The Complete Beginners Introduction to Emacs - Part 2/3 - The Basics of Emacs
:tech:emacs:...
+ DONE Se questo è un uomo :life:books:...
+ DONE The Complete Beginners Introduction to Emacs - Part 3/3 - A Crash Course on Emacs-Lisp
:tech:emacs:...
+ DONE Sulla pigrizia :life:...
+ DONE The Importance of RSS :tech:university:...
+ DONE How I Learned the Power of the Command Line :life:htb:university:...
+ DONE How I Track my Expenses with Ledger :tech:finance:...
+ DONE On Doubly-Linked Lists, and how they are Implemented in the Linux Kernel
:tech:data_structures:programming:...
+ DONE Implementing a Crossword Solver in C++ :tech:programming:games:...
+ DONE La tregua :books:life:...
+ DONE Lettera pre COVID-19 :life:...

+ HTB Writeups
:PROPERTIES:...

+ DONE HTB Writeup - Bashed :tech:htb:...
+ DONE HTB Writeup - Nibbles :tech:htb:...
+ DONE HTB Writeup - Poison :tech:htb:...
+ DONE HTB Writeup - Valentine :tech:htb:...

+ Scratch...
+ Footnotes...
+ COMMENT Local Variables :ARCHIVE:...
```

```
cd ~
cv whoami favorites archive

Hello, whoever you are, and welcome to my blog (♥).

I'm a university student, and I enjoy writing about a bunch of different things, such as:

books data-structures emacs finance games htb life programming tech university

I also share all my university lecture notes, the code I write, and other stuff in my ppa, with the hope that it may help you in some way.

RSS YouTube Instagram Twitter Email

And now, some quotes:



The heart has its reasons which reason knows nothing of.  
Blaise Pascal



---



### HTB Writeup - Valentine



---



2021-03-29  
#tech #htb



Complete writeup of the Valentine machine in Hack The Box.



Read more →


```

+ Bonus † Inside heartbleed (CVE-2014-0160)

To understand better this vulnerability the following resources might help:

- [Commit in which it was fixed](#)
- [CVE-2014-0160](#)
- <https://www.exploit-db.com/exploits/32745>
- <https://www.exploit-db.com/exploits/32764>
- <https://stackabuse.com/heartbleed-bug-explained/>

By downloading the version of the code right before it got fixed, we can start to analyze the flaw. The files of interested are two, as is shown the commit: `ssl/d1_both.c` and `ssl/tl_lib.c`. Since however the `d1_both.c` is used for the `DTLS` implementation, which a version of TLS that uses the UDP transport protocol - as opposed to the traditional version of TLS that uses TCP - we will only check out `tl_lib1.c`. Let us first discuss the flawed version of the code, and at the end we will discuss how it was fixed.

The function of interest is `tls1_process_heartbeat()`. As soon as we enter in the function the server reads the number bytes that the client declared to have sent, and puts such number in the `payload` variable.

```
#begin_src c
#ifndef OPENSSL_NO_HEARTBEATS
int
tls1_process_heartbeat(SSL *s)
{
    // -- p points to record data from client
    unsigned char *p = &s->s3->rrec.data[0], *pl;
    unsigned short hbttype;
    unsigned int payload;
    unsigned int padding = 16; /* Use minimum padding */

    /* Read type and payload length first */
    hbttype = *p++;
    // -- read 2 bytes from p and put them in payload
    n2s(p, payload);
    pl = p;

    // -- now payload has length of data from client
}
#endif
```

If the message was an heartbeat request, the server then proceeds to allocate memory. The memory allocated depends on the amount of data specified by the client, that is on the variable `payload`.

```
#begin_src c
#include <openssl/ssl.h>
#include <openssl/bio.h>
#include <openssl/err.h>
#include <openssl/asn1.h>
#include <openssl/pem.h>
#include <openssl/x509.h>
#include <openssl/evp.h>
#include <openssl/asn1t.h>
#include <openssl/asn1.h>
#include <openssl/asn1t.h>
```

blog/content.org/content.org 9407:70 95%

LF UTF-8 Org ↵ master

Bonus

Inside heartbleed (CVE-2014-0160)

To understand better this vulnerability the following resources might help:

- [Commit in which it was fixed](#)
- [CVE-2014-0160](#)
- <https://www.exploit-db.com/exploits/32745>
- <https://www.exploit-db.com/exploits/32764>
- <https://stackabuse.com/heartbleed-bug-explained/>

By downloading the version of the code right before it got fixed, we can start to analyze the flaw. The files of interested are two, as is shown the commit: `ssl/d1_both.c` and `ssl/tl_lib.c`. Since however the `d1_both.c` is used for the `DTLS` implementation, which a version of TLS that uses the UDP transport protocol - as opposed to the traditional version of TLS that uses TCP - we will only check out `tl_lib1.c`. Let us first discuss the flawed version of the code, and at the end we will discuss how it was fixed.

The function of interest is `tls1_process_heartbeat()`. As soon as we enter in the function the server reads the number bytes that the client declared to have sent, and puts such number in the `payload` variable.

```
#ifndef OPENSSL_NO_HEARTBEATS
int
tls1_process_heartbeat(SSL *s)
{
    // -- p points to record data from client
    unsigned char *p = &s->s3->rrec.data[0], *pl;
    unsigned short hbttype;
    unsigned int payload;
    unsigned int padding = 16; /* Use minimum padding */

    /* Read type and payload length first */
    hbttype = *p++;
    // -- read 2 bytes from p and put them in payload
    n2s(p, payload);
    pl = p;
```

PRESENTATIONS

Questa stessa presentazione è stata generata a partire da un file org.

```
#+TITLE: Introduzione ad Emacs
#+AUTHOR: Leonardo Tamiano
#+OPTIONS: num:nil toc:2
#+REVEAL_THEME: cyberpunk

* Che cos'è Emacs?...
* Come utilizzo Emacs
#+REVEAL: split

Per quanto Emacs sia un tool estremamente potente, la configurazione iniziale lascia molto a desiderare, sia in termini di estetica che di funzionalità.

#+REVEAL_HTML: 
#+REVEAL: split

Investendo abbastanza tempo ed impegno però è possibile trasformarlo in ciò che si vuole.

#+REVEAL_HTML: <div>
#+REVEAL_HTML: <div class="column" style="float:left; width:40%">
#+REVEAL_HTML: 
#+REVEAL_HTML: </div>

#+REVEAL_HTML: <div class="column" style="float:left; margin-top:130px; margin-left:50px;">
$${\longrightarrow}$$
#+REVEAL_HTML: </div>

#+REVEAL_HTML: <div class="column" style="float:right; width:40%">
#+REVEAL_HTML: 
#+REVEAL_HTML: </div>
#+REVEAL_HTML: </div>

#+REVEAL: split

Andiamo adesso a vedere alcuni (pochi) use-case personali di utilizzo di Emacs.

+ Interfacciamento OS...
+ Programmazione...
+ Org-Mode...
* Perché imparare Emacs?... |
```

intro.emacs.org 270:0 11:38 LF UTF-8 Org
Mark set

INTRODUZIONE AD
EMACS
LEONARDO TAMIANO

Created: 2021-04-24 sab 11:37



PERCHÉ IMPARARE EMACS?

Imparare Emacs non è facile, in quanto richiede tempo ed impegno. Detto questo, sono tante e varie le ragioni per cui, secondo me, vale la pena provare ad imparare Emacs.

PRODUTTIVITÀ

In generale le potenzialità offerte da Emacs per aumentare la propria produttività sono tantissime, e sono tanti gli insegnamenti che è possibile estrarre dall'utilizzo di queste tecnologie.

Solo org-mode, una parte (molto importante) di Emacs, contiene al suo interno innumerevoli applicazioni, molte delle quali possono essere sviluppate solo con ingegno e creatività.

INTRODUZIONE A LISP

Dato che Emacs è un interprete del linguaggio elisp, per imparare (bene) Emacs dobbiamo anche imparare la famiglia di linguaggi LISP, che molto spesso non viene trattata molto all'università, ma che presenta molte caratteristiche tremendamente affascinanti.

CRESCITA PERSONALE

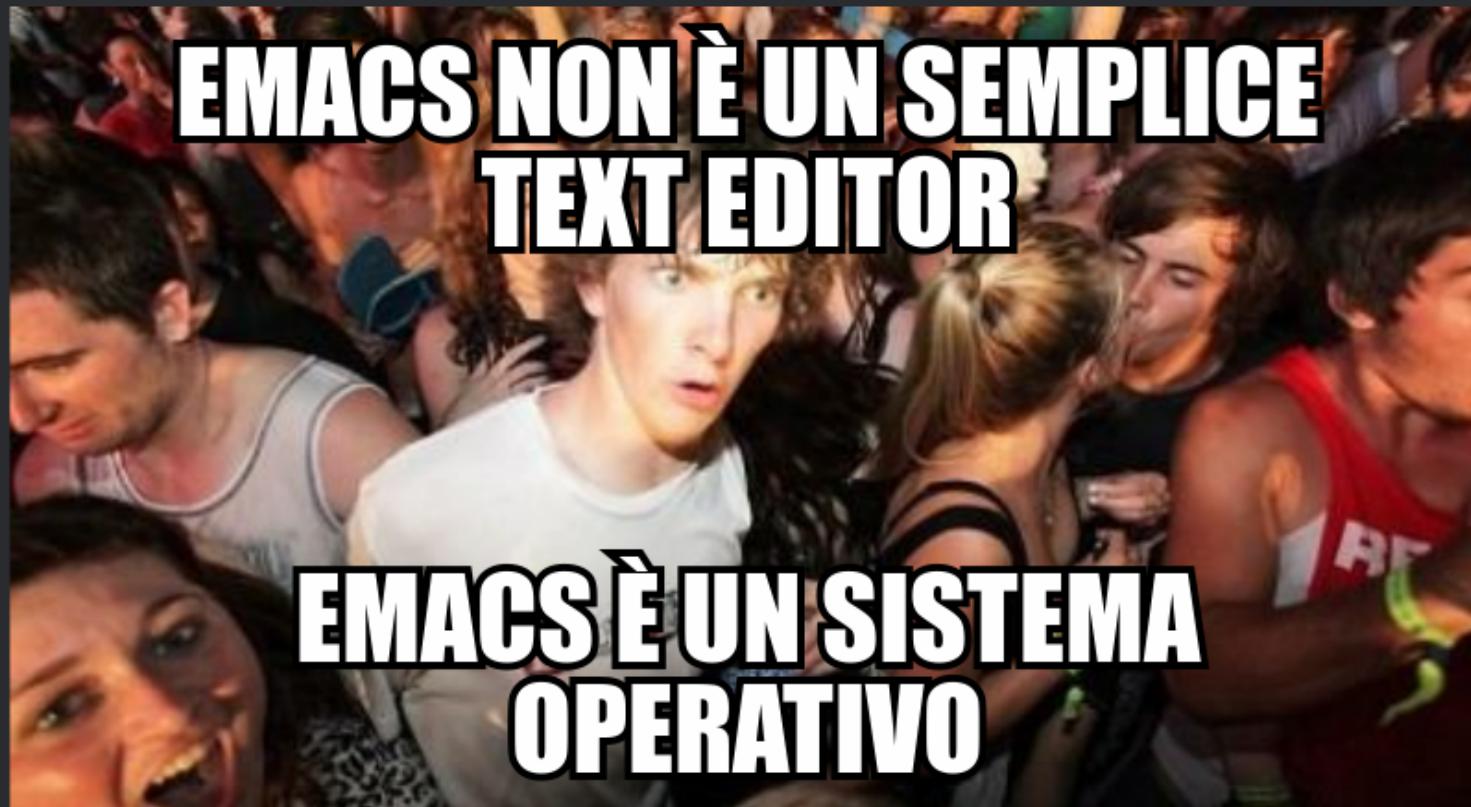
Più è difficile una determinata cosa, e più imparare quella cosa ci fa crescere, direttamente e indirettamente.

Sotto questo punto di vista, imparare Emacs apre la mente alle vere potenzialità offerte dal software, molte delle quali col passare degli anni stiamo perdendo di vista.

In ultima analisi, imparare Emacs è divertente, e apre un universo di possibilità ancora inesplorate.

Anche la possibilità di creare memes.

| | | | | | |
|--------|-------------------------------------|----|-------|--------|--------|
| Top | Emacs non è un semplice text editor | | | | |
| | 20 | 60 | white | middle | impact |
| Bottom | Emacs è un sistema operativo | | | | |
| | 20 | 60 | white | middle | impact |



✉️ 🔒 *meme* 3:18

1:08 LF UTF-8 Meme

email_sync: finished.

