

Università degli Studi di Catania

CdLM in Informatica Magistrale

Corso di Compilatori A.A. 14-15

Federico Vindigni

Relazione finale di progetto

M@ilicious

The simple and smart email parser

Introduzione

Mailicious è un semplice ma potente programma che consente all'utente di creare, modificare ed esportare mailing list.

La punta di diamante di Mailicious è il suo parser, molto intelligente e flessibile, che unito ad un DBMS creato ad hoc permette di riconoscere ed importare molto velocemente qualsiasi email presente in un file plain text.

Il programma, sebbene dotato di un'asciutta ed efficiente interfaccia a riga di comando, si propone come un back-end da essere utilizzato in altri progetti software.

L'intero programma è scritto in ANSI C, con un leggero utilizzo della libreria POSIX. Questo, oltre a garantire prestazioni sempre al top, permette facilmente il porting nei sistemi più diffusi.

Il codice sorgente nella sua totalità è rilasciato sotto licenza GNU GPL v3 ed è disponibile su GitHub. Per scaricarlo, contribuire al suo sviluppo o fare una fork visita <http://github.com/federicovindigni/mailicious>.

Struttura del software

Il software è composto da tre moduli: il parser (parser.c), il gestore delle mailing list (emfs.c) e l'interfaccia a riga di comando (mailicious.c).

I file utils.c e env.h contengono routine, variabili e costanti di supporto al programma.

Ciò che segue è una descrizione a grandi linee dei moduli. Per il funzionamento dettagliato fare riferimento ai commenti presenti nel codice.

Il parser è il fiore all'occhiello del programma. Esso è in grado di riconoscere le email all'interno di un file plain text senza fare supposizioni sulla formattazione del suo contenuto. Attuando una analisi su tre livelli delle email (lessicale, sintattica e semantica) il parser garantisce un'ottima percentuale di affidabilità. Ma la sua vera forza resta però l'altissima configurabilità. È possibile infatti modificare ogni singolo parametro del parser affinché sia più selettivo o viceversa più permissivo, consentendo ad esempio di ignorare gli errori singoli all'interno di un'email. Per fare ciò vedere il file /bin/conf.

Il gestore delle mailing list si occupa di organizzare e gestire le mailing list all'interno della memoria di massa. Le mailing list sono salvate nella cartella /bin/db. Ogni mailing list è composta da tre file: uno con estensione .dat che contiene le email; uno .ind che contiene gli indici di accesso al file .dat, per velocizzare le operazioni di lettura/scrittura; uno .fds che contiene gli slot di memoria liberi all'interno del file .dat.

Il gestore fornisce al parser e all'interfaccia utente le primitive necessarie per operare sulle mailing list.

L'interfaccia utente è a riga di comando e può essere usata in due modi differenti in base alle necessità. Si può accedere alla console del programma digitando `./mailicious` potendo così eseguire più comandi di seguito, oppure con `./mailicious command arg1 arg2..` il programma eseguirà quanto indicato alla fine terminerà.

Quest'ultima modalità è utile per eseguir Mailicious da un programma esterno.

Per informazioni sui comandi di Mailicious digitare `./mailicious help`.

Compilazione ed utilizzo del software

Nella directory di Mailicious si trovano le cartelle `/src`, `/bin` e `/doc`, che contengono il codice sorgente, il programma eseguibile e la documentazione.

Per utilizzare il programma bisogna compilarlo dai sorgenti. Per fare ciò da terminale spostarsi su `/src` e digitare `make` (se si utilizza un compilatore diverso da `gcc` modificare la variabile `CC` nel `Makefile` con il nome del compilatore usato). Se il codice verrà compilato correttamente il programma `mailicious` verrà posizionato in `/bin`.

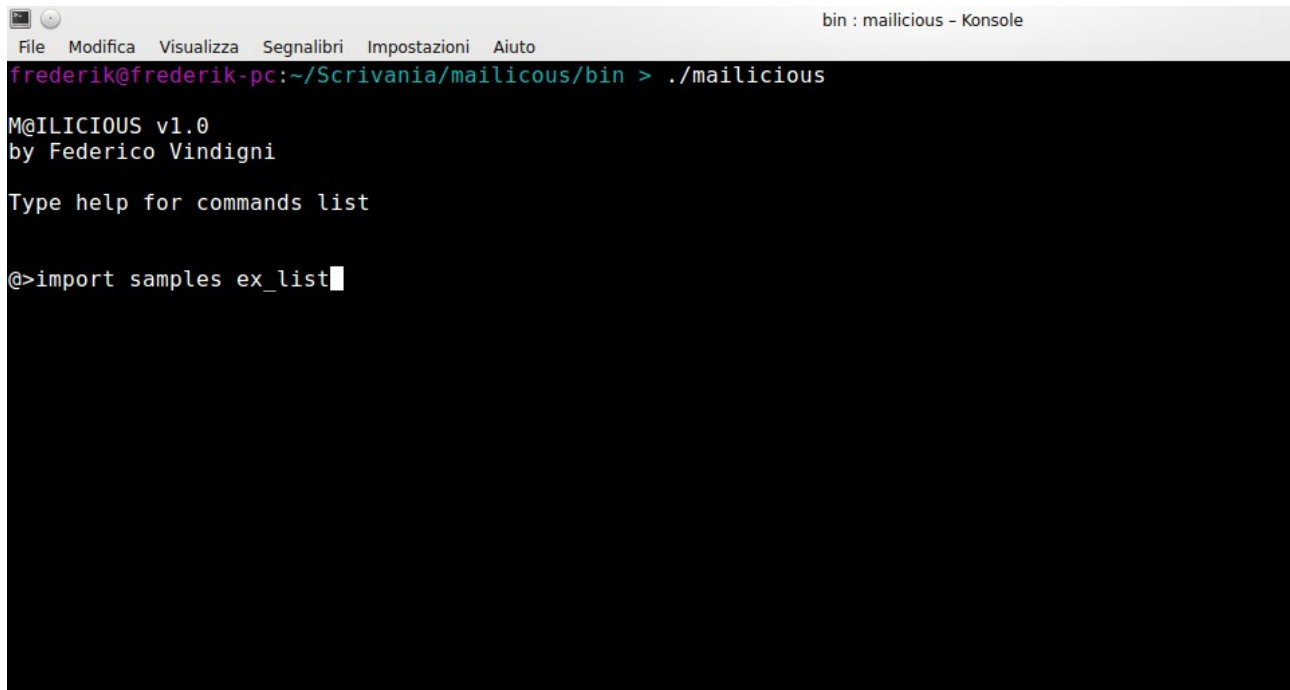
Per eseguire il programma spostarsi nella cartella `/bin` e lanciare `./mailicious` oppure `./mailicious command args1 arg2 ...` (leggi sopra la due modalità di utilizzo).

I comandi accettati da Mailicious sono i seguenti (per i dettagli sui loro argomenti digita `./mailicious help`):

- `import` : prende in input un file o una cartella ne fa il parsing e crea una nuova mailing list con email trovate.
- `export` : prende in input una mailing list e crea una cartella dove salva le email in vari file in base alla lettera di inizio dell'email.
È possibile specificare il numero di email che ogni file può contenere (in questo caso avremo più file per ogni lettera).
- `search` : dice se una data email è presente o meno in una mailing list.
- `update` : modifica un'email di una mailing list.
- `delete` : elimina un'email o una insieme di email lette da un file da una mailing list.
- `remove` : cancella una mailing list dal database.
- `ping` : effettua il ping dei domini di una mailing list dando in output il tempo di risposta di ogni dominio, la risposta più veloce, quella più lenta e la risposta media. I risultati vengono salvati in un file.

Esempio di utilizzo

In /bin abbiamo una cartella samples contenente 111 file di vari formati (.txt, .rtf, etc.) per una dimensione totale di 1,7MB. Vogliamo creare una mailing list con tutte le email presenti in questi file.



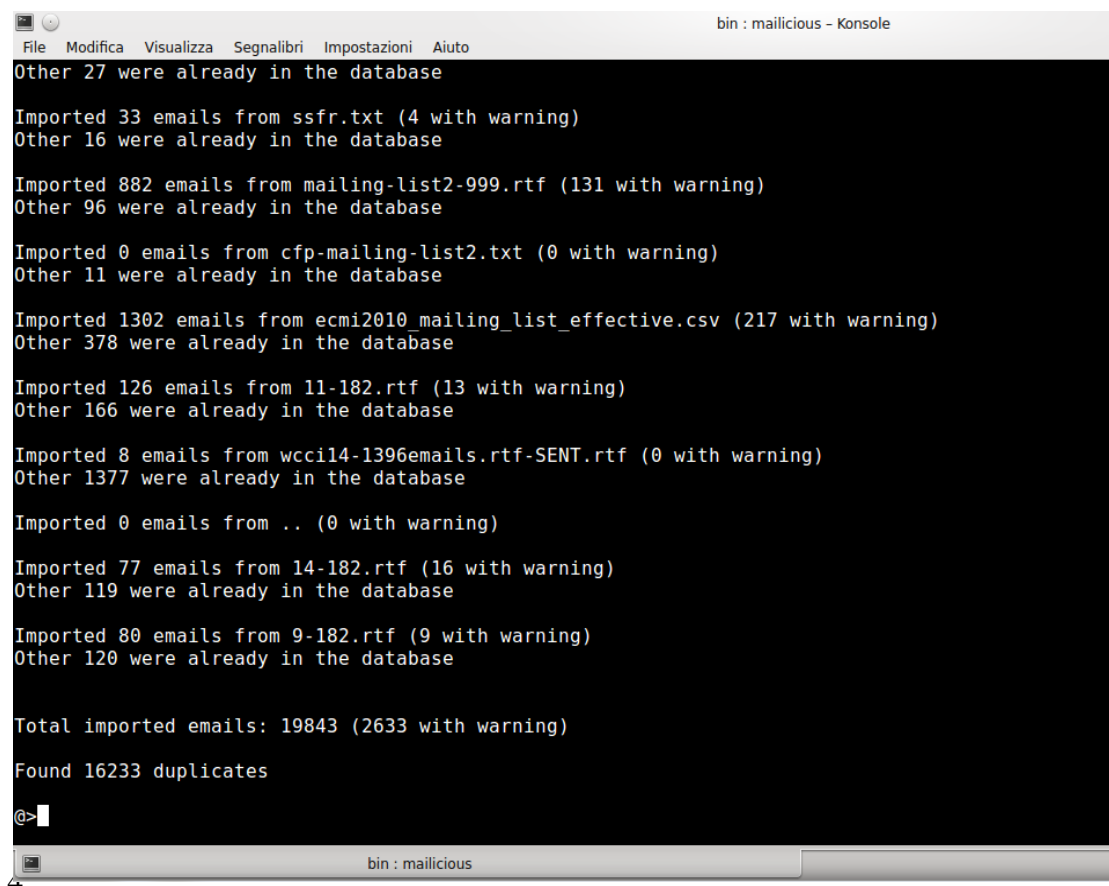
```
bin : mailicious - Konsole
File Modifica Visualizza Segnalibri Impostazioni Aiuto
frederik@frederik-pc:~/Scrivania/mailicious/bin > ./mailicious

M@ILICIOUS v1.0
by Federico Vindigni

Type help for commands list

@>import samples ex_list
```

L'output ottenuto è il seguente



```
bin : mailicious - Konsole
File Modifica Visualizza Segnalibri Impostazioni Aiuto
Other 27 were already in the database

Imported 33 emails from ssfr.txt (4 with warning)
Other 16 were already in the database

Imported 882 emails from mailing-list2-999.rtf (131 with warning)
Other 96 were already in the database

Imported 0 emails from cfp-mailing-list2.txt (0 with warning)
Other 11 were already in the database

Imported 1302 emails from ecmi2010_mailing_list_effective.csv (217 with warning)
Other 378 were already in the database

Imported 126 emails from 11-182.rtf (13 with warning)
Other 166 were already in the database

Imported 8 emails from wccil4-1396emails.rtf-SENT.rtf (0 with warning)
Other 1377 were already in the database

Imported 0 emails from .. (0 with warning)

Imported 77 emails from 14-182.rtf (16 with warning)
Other 119 were already in the database

Imported 80 emails from 9-182.rtf (9 with warning)
Other 120 were already in the database

Total imported emails: 19843 (2633 with warning)
Found 16233 duplicates

@>
```

Dopo questo comando è stata creata una lista di nome `ex_list` contenente 19843 email. Nella macchina usata per l'email (Intel core i3 con 4GB di RAM e sistema operativo Linux) il tempo richiesto per l'operazione è stato 2 secondi.

Supponiamo ora di volere esportare la mailing list.

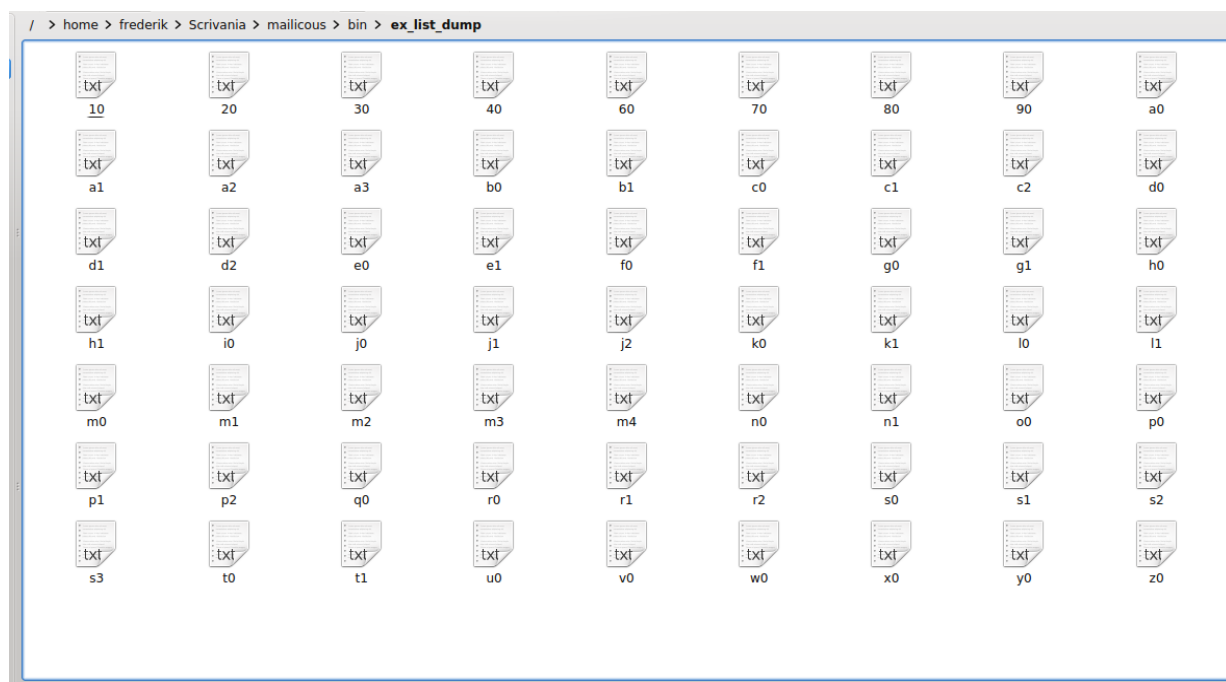
```
bin : mailicious - Konsole
File Modifica Visualizza Segnalibri Impostazioni Aiuto
frederik@frederik-pc:~/Scrivania/mailicious/bin > ./mailicious

M@ILICIOUS v1.0
by Federico Vindigni

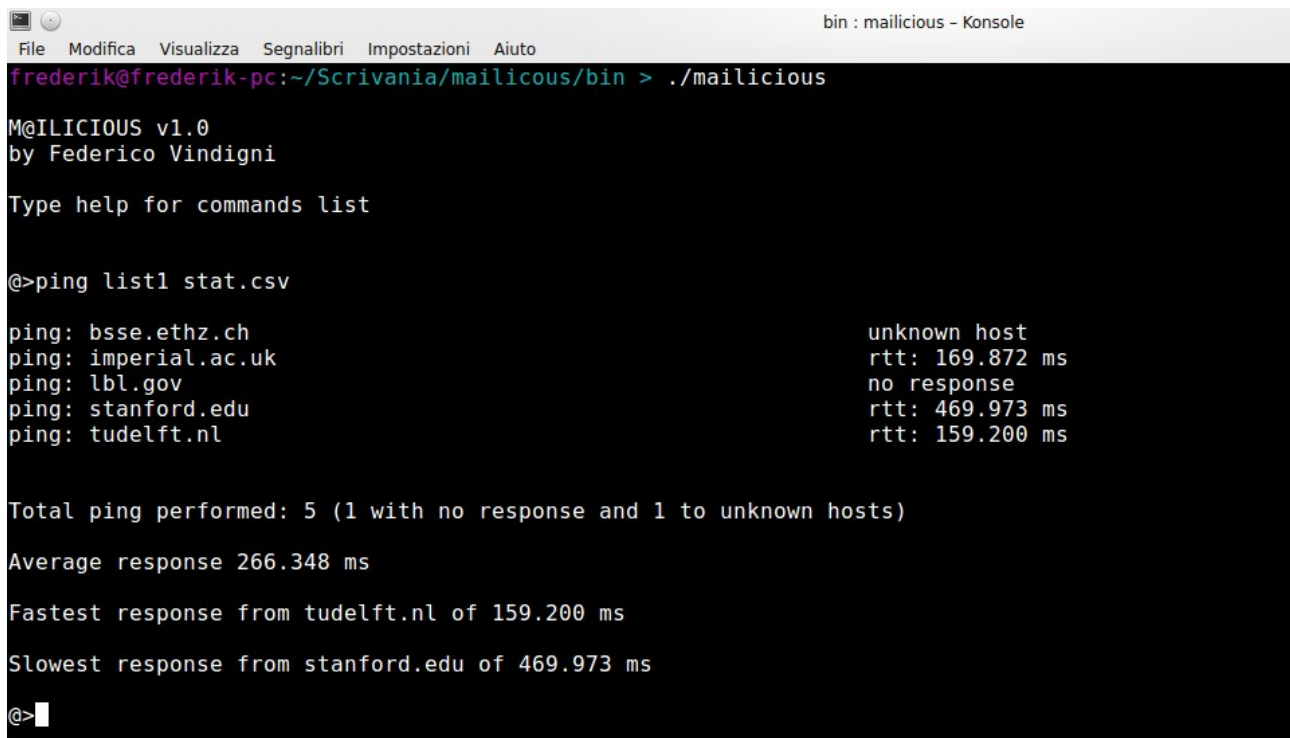
Type help for commands list

@>export ex_list ex_list_dump 500 ;\n
```

In questo modo verrà creata la cartella `ex_list_dump` con file contenenti al più 500 email separate ognuna da un punto e virgola e un ritorno a capo.



Supponiamo ora di voler fare il ping dei domini della mailing list chiamata list1 e mettere l'output sul file stat.csv



```
bin : mailicious - Konsole
File Modifica Visualizza Segnalibri Impostazioni Aiuto
frederik@frederik-pc:~/Scrivania/mailicious/bin > ./mailicious

M@ILICIOUS v1.0
by Federico Vindigni

Type help for commands list

@>ping list1 stat.csv

ping: bsse.ethz.ch                unknown host
ping: imperial.ac.uk             rtt: 169.872 ms
ping: lbl.gov                    no response
ping: stanford.edu               rtt: 469.973 ms
ping: tudelft.nl                 rtt: 159.200 ms

Total ping performed: 5 (1 with no response and 1 to unknown hosts)
Average response 266.348 ms
Fastest response from tudelft.nl of 159.200 ms
Slowest response from stanford.edu of 469.973 ms

@>
```

Conclusioni

Come mostrato Mailicious è uno strumento semplice quanto potente. Come già detto, il fatto di essere codificato esclusivamente in C rende la sua portabilità elevatissima non vincolando l'utente ad installare altro software.

La complessità delle operazioni è $O(n)$, con n numero di email, per tutte quelle operazioni che agiscono su più email (es: import, export). La complessità delle operazioni che agiscono su singole email (es: search, update) si attesta ad $O(1)$ finché le email in una lista non sono maggiori di circa 100000. Andando oltre la complessità tende a diventare logaritmica.

Per chiarimenti, suggerimenti o per segnalare un problema mandate un'email a federico.vindigni@gmail.com.