

Università degli studi di Catania

Corso di Laurea Magistrale in Informatica

Compilatori A.A. 2014-15

Cantarella Danilo, Randazzo Filippo, Siragusa Sebastiano

Un mailParser in C per il parsing di file di testo

Introduzione

Lo scopo del progetto è stato quello di realizzare un parser per l'estrazione automatica di emails valide da file di testo.

In particolare, si è realizzato un back-end per la gestione completa di tutto il processo, dall'analisi della directory alla popolazione e manipolazione del database.

Per lo sviluppo del software è stato utilizzato il linguaggio C per la gestione del parsing dei file di testo e un database mySQL per la memorizzazione e la gestione di tutte le emails estratte.

Il codice sorgente è rilasciato sotto una licenza GNU GPL v3 ed è disponibile su GitHub all'indirizzo <https://github.com/Flyer-90/mailParser>.

Funzionamento

Il software presenta 8 diverse funzionalità, accessibili attraverso un menu rapido. Queste permettono di processare intere directory di file oppure solamente un singolo file, di operare sul database per ricercare emails, inserirne o cancellarne, di esportare tutto il database in file di testo e di effettuare il ping dei domini per trovare le emails valide o meno.



Al primo avvio del software, viene richiamata la funzione **initDB()** per effettuare un controllo al fine di verificare la presenza in locale del database **mailParser**. Se non esiste, questo viene creato ed inizializzato con la creazione di tutte le tabelle necessarie.

Viene creata una tabella per ogni possibile carattere contenuto in una mail valida. Ciò per permettere un'ordinamento delle emails e quindi una più semplice gestione di esse per operazioni future.

Operazione di subscribe

La funzionalità di processing di una directory permette di analizzare ed estrarre tutte le emails valide a partire da un path esistente. Sono supportati file in formato txt, doc, rtf, e csv.

La funzione **processDirectory**, in particolare, prende in input dall'utente un path valido che viene parsato attraverso la funzione **readDir** per leggere la lista di tutti i file presenti nella directory ed analizzare dunque quelli supportati.

Per ogni file supportato viene richiamata la funzione **processFile** che prende in input il path assoluto del file e un flag **mode**.

Il file viene aperto in modalità lettura per essere letto riga per riga, fino alla EOF. Ogni riga viene splittata per spazio, virgola e punto e virgola, ricavando dunque una serie di token. Per ognuno di questi token viene quindi richiamata la funzione **matchMail** che si occupa di verificare se esso rappresenta un'email sintatticamente valida o meno.

Viene utilizzata la funzione **regexec** per una validazione delle emails tramite le espressioni regolari. L'espressione regolare è stata definita in base alle specifiche dei documenti RCF 5322 ed RCF 5321.

Quindi, una volta applicata l'espressione regolare, se c'è matching, viene estratta dal token la stringa corrispondente alla email trovata e viene avviata una fase di validazione del dominio della email.

Viene applicata una seconda espressione regolare, contenente tutti i domini validi. Viene inoltre eseguito un'ulteriore controllo per evitare falsi positivi generati da eventuali matching su sottostringhe (es: dominio gmail.come)

Se l'email supera tutti i controlli, viene dunque valutata come sintatticamente valida e viene inserita nel database attraverso una query sql.

Grazie all'uso del database è stato gestito in automatico il controllo delle emails duplicate. Questo perchè il campo email è stato utilizzato come chiave primaria.

Una mail può essere considerata valida ma con un warning, con il risultato di essere inserita all'interno di una speciale tabella **warning**. La segnalazione di warning avviene quando l'email, e in particolare la parte del name, è molto lunga. Questo serve ad identificare quelle emails che magari hanno avuto problemi di codifica, con il risultato di essere state salvate in formato esadecimale.

Infine, il valore del flag **mode** (1 per la modalità di inserimento, 2 per la modalità di cancellazione) permette di fare entrambe le funzionalità di inserimento e cancellazione dal database, corrispondenti dunque alle funzionalità di subscribe e unsubscribe dalla mailing list.

Se invece si vuole aggiungere un solo file alla mailing list, è possibile processarlo specificando direttamente il path completo. Su tale file verrà eseguito lo stesso processo applicato in precedenza a tutti i file della directory.

E' infine possibile inserire manualmente anche una sola email.

Operazione di unsubscribe

Dopo aver popolato il database attraverso le operazioni viste in precedenza, è possibile effettuare delle cancellazioni per eseguire l'unsubscribe di indirizzi email dalla mailing list.

Questa operazione viene eseguita sfruttando la funzione **matchMail** descritta in precedenza. In particolare, viene settato il flag **mode** a 2, in modo tale da modificare le query di insert con nuove query di delete.

E' possibile inoltre cancellare manualmente anche una sola email.

Grazie all'uso del database è risultato semplice gestire eventuali tentativi di cancellare emails non presenti nel database.

Export del database su file di testo

Questa operazione permette di esportare l'intero database su dei file di testo (in formato .txt) da massimo 500 righe ciascuno.

L'export viene fatto all'interno di una cartella output, create automaticamente se non esiste, che contiene a sua volta una cartella per ogni tabella presente all'interno del database.

In particolare, per ognuna delle tabelle presenti, viene fatta una query per leggere tutte le emails contenute in essa. Si ottiene dunque una lista ordinata alfabeticamente di tutte le emails, che sarà così scritta su file. Vengono scritte al massimo 500 emails per ogni file, che vengono creati a runtime secondo necessità all'interno della cartella corrispondente alla tabella in questione.

Viene creata inoltre una speciale cartella warning dove vengono scritte tutte le emails che sono state segnalate come warning e quindi inserite nell'apposita tabella. In questo modo è possibile visionare quelle emails che potrebbero essere state corrotte a causa di problemi di codifica.

Ping dei domini

Questa funzionalità permette di effettuare il ping di tutti i domini delle emails trovate, in modo da verificare quali, oltre ad essere sintatticamente valide, sono anche effettivamente funzionanti.

Attraverso la funzione **initTableDomains** viene popolata la tabella domains del database.

Da ogni emails presente nel database viene estratto il dominio ed inserito in questa tabella. Quando il dominio è già stato inserito, viene aumentato il relativo contatore, in modo tale da sapere esattamente quante emails hanno quel particolare dominio.

La prima volta questa operazione viene fatta in automatico. Dalla volta successiva in poi, viene chiesto all'utente se intende aggiornare la tabella dei domini. Questa operazione può essere utile nel caso in cui siano state inserite emails con nuovi domini non ancora presenti nella tabella e quindi non ancora pingati.

Per ogni dominio l'operazione di ping viene eseguita con un timeout massimo di 5 secondi, oltre la quale il dominio viene considerato non raggiungibile. Dopo ogni ping viene fatta una query di update per inserire le informazioni sul tempo di risposta o per settare eventualmente quel dominio come non raggiungibile. Questo comunque non significa che l'email non è valida, poichè molti domini, se pur esistenti, non rispondono all'operazione di ping.































































Caso concreto

Per il testing del software, è stata utilizzata una directory contenente 110 files per un totale di 1,7 MB di testo.

Attraverso l'uso della funzione di parsing di una directory, si hanno i seguenti risultati:

```
Processing file listmembers_web.txt...
Processing file 4-40.rtf...
Processing file emboss.txt...
Processing file four_batch.txt...
Processing file wyss_email.txt...
Processing file mailing-list7.rtf...
Processing file BD_MAILING_ECMI_Conjunta1-2-3_.txt...
Processing file cfp-mailing-list2.txt...
Have been processed 110 files.
Have been inserted 19695 email.
Execution time: 18.685028 seconds
```

Sono state dunque inserite nel database 19695 emails, con un tempo totale di esecuzione di circa 18 secondi.

		 Struttura	 SQL	 Cerca	 Query da esempio	 Esporta	 Importa		
<input type="checkbox"/>	a		 Mostra	 Struttura	 Cerca	 Inserisci	 Svuota	 Elimina	1,657
<input type="checkbox"/>	b		 Mostra	 Struttura	 Cerca	 Inserisci	 Svuota	 Elimina	883
<input type="checkbox"/>	c		 Mostra	 Struttura	 Cerca	 Inserisci	 Svuota	 Elimina	999
<input type="checkbox"/>	d		 Mostra	 Struttura	 Cerca	 Inserisci	 Svuota	 Elimina	1,012
<input type="checkbox"/>	e		 Mostra	 Struttura	 Cerca	 Inserisci	 Svuota	 Elimina	570
<input type="checkbox"/>	f		 Mostra	 Struttura	 Cerca	 Inserisci	 Svuota	 Elimina	560
<input type="checkbox"/>	g		 Mostra	 Struttura	 Cerca	 Inserisci	 Svuota	 Elimina	855
<input type="checkbox"/>	h		 Mostra	 Struttura	 Cerca	 Inserisci	 Svuota	 Elimina	571
<input type="checkbox"/>	i		 Mostra	 Struttura	 Cerca	 Inserisci	 Svuota	 Elimina	407
<input type="checkbox"/>	j		 Mostra	 Struttura	 Cerca	 Inserisci	 Svuota	 Elimina	1,219

L'operazione di export del database ha prodotto 103 files (ognuno contenente al massimo 500 emails) con un tempo di esecuzione di soli 0,098 secondi.

```
Exporting database to files...
Have been exported 19695 email in 65 files in a total of 37 folders.
Execution time: 0.098730 seconds
```

Ad esempio, per la tabella 'a' che conteneva 1657 emails, sono stati prodotti 4 files a_0.txt, a_1.txt, a_2.txt e a_3.txt.

Il ping dei domini è invece un'operazione che richiede ovviamente molto più tempo di esecuzione, circa 19500 secondi, ossia 315 minuti.

Dalle tabelle delle emails vengono estratti un totale di 5220 domini.

I risultati dell'operazione di ping sono i seguenti:

- dei 5220 domini pingati, 1491 hanno risposto entro il timeout;
- il tempo medio di risposta è di 112,88 ms;
- su un totale di 19695 emails, 9123 hanno il dominio che ha risposto al ping.

Conclusioni

Il software riesce a gestire molto bene una grande quantità di emails, supportando quasi tutti i formati di file di testo.

La complessità di ogni funzione è $O(n)$ dove n corrisponde al numero di emails presenti. Tutte le operazioni disponibili sono dunque eseguite in tempi rapidi, ad eccezione della funzione di ping che richiede maggior tempo a causa del timeout impostato a 5 secondi. Invece, operazioni come l'inserimento, la cancellazione o la ricerca di una singola email hanno complessità $O(1)$.

Per eventuali suggerimenti o per segnalare qualsiasi problema, i nostri contatti sono i seguenti:

- Cantarella Danilo - cantarella.danilo@gmail.com
- Randazzo Filippo - filippo.rnd@gmail.com
- Siragusa Sebastiano - seby.siragusa@gmail.com