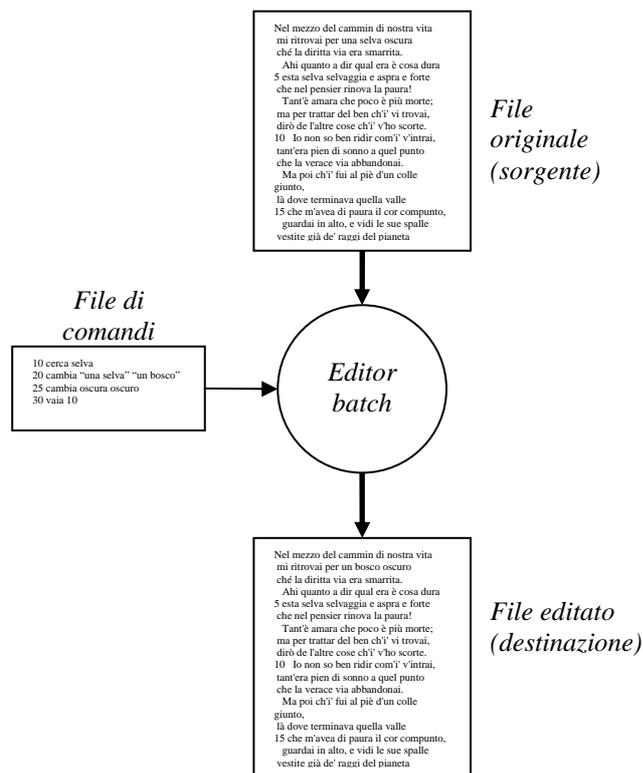


Laboratorio di Linguaggi di Sistema – a.a. 2005/2006 – Progetto finale

Il problema

Si vuole realizzare un editor di file testuali che funzioni in modalità batch. Un editor batch svolge funzioni analoghe a quelli dei comuni editor interattivi, ma legge una serie di comandi di editing da applicare al testo da un file, anziché da tastiera. Ciò consente di applicare dei comandi di editing in maniera automatica, ripetitiva, e potenzialmente ad un numero molto grande di file senza che l'utente debba compiere manualmente le operazioni di editing. Il modello di editing è il seguente:



L'editor batch legge una serie di istruzioni da un file di comandi, e li applica a uno o più file di origine, producendo uno o più file editati, ottenuti dagli originali attraverso l'applicazione delle istruzioni. Le istruzioni sono numerate, e appaiono nel file di comandi in un ordine qualunque; l'esecuzione segue l'ordine della numerazione (non necessariamente quello con cui le istruzioni compaiono nel file). Le istruzioni operano sui file sorgente una riga alla volta. Quando sono presenti più file sorgente, essi vengono elaborati uno alla volta, ripetendo l'applicazione dei comandi dall'inizio con ogni nuovo file; per ciascuno dei file di origine, indicati sulla riga di comando, i risultati vengono scritti in un file con lo stesso nome, ma con l'aggiunta del suffisso ".new". Quando tutti i file sono stati elaborati, l'esecuzione termina; l'editor deve concludere l'elaborazione stampando su *stderr* alcune statistiche sulle operazioni compiute (numero di righe lette, numero di righe scritte, numero di comandi interpretati, ed eventualmente altri a scelta dello studente).

In generale, il formato di una riga del file di comandi è il seguente:

nn *istruzione* *parametri*

in cui *nn* è il numero assegnato al comando in questione, *istruzione* è la parola chiave che identifica l'istruzione (una lista completa viene data nel seguito), e *parametri* sono 0 o più parametri necessari all'istruzione. I parametri

sono separati da spazio, e possono essere usate le virgolette per delimitare stringhe contenenti spazi. *Avanzato***: si implementi un sistema di codici di escape simile ai vari "\n", "\t", "\"" del C per permettere l'espressione di stringhe arbitrarie all'interno del file di comandi.

In ogni istante, l'editor mantiene una *riga corrente*, proveniente dal file di input, a cui vengono applicati i comandi; alcuni comandi modificano il contenuto della riga corrente, mentre altri rendono corrente una riga diversa (e tipicamente successiva) del file.

L'elenco completo delle istruzioni, con i relativi parametri, è il seguente:

- **prossima** – rende corrente la riga successiva; se la riga corrente era l'ultima del file corrente, l'esecuzione dei comandi per questo file termina.
- **cerca stringa** – rende corrente la prima riga successiva che contiene la *stringa* data; se nessuna riga successiva contiene la stringa data, l'esecuzione dei comandi per questo file termina.
- **cambia stringa1 stringa2** – sostituisce tutte le occorrenze di *stringa1* nella riga corrente con *stringa2*.
- **traslitera stringa1 stringa2** – traslitera i caratteri di *stringa1*, sostituendo ogni occorrenza nella riga corrente di ciascuno dei caratteri in *stringa1* con il carattere in *stringa2* che occupa la stessa posizione. Se *stringa2* è più corta di *stringa1*, e quindi per qualche carattere non è disponibile la versione trasliterata, il carattere in questione viene semplicemente cancellato.
- **setrovi stringa comando** – se la riga corrente contiene almeno un'istanza di *stringa*, esegue il *comando*.
- **vaia nr** – continua l'esecuzione dall'istruzione numerata *nr*, o dalla prima istruzione numerata con un numero maggiore di *nr* se non esiste un'istruzione numerata *nr*.
- **premetti stringa** – inserisce *stringa* all'inizio della riga corrente.
- **appendi stringa** – aggiunge *stringa* alla fine della riga corrente.
- **cancella** – elimina la riga corrente, che quindi non viene copiata nel file editato; rende corrente la riga successiva (come per il comando **next**).
- **inserisci stringa** – inserisce una riga contenente la *stringa* data nel file editato; non cambia la riga corrente.
- **finfile** – termina l'esecuzione dei comandi sul file corrente; tutte le righe del file rimanenti, se ve ne sono, vengono copiate nel file editato senza modifiche. Se non ci sono altri file da elaborare, l'esecuzione termina.
- **fine** – termina l'esecuzione dei comandi nel file corrente, come per **finfile**, e copia senza altre modifiche tutti i file rimanenti.

- **muori** – termina l'esecuzione in maniera forzata; il file corrente non viene completato, e i file successivi non vengono elaborati.

*Avanzato**:* lo studente può aggiungere a questa lista altre istruzioni che ritenga utili, senza alterare la modalità di funzionamento dell'editor batch. Per esempio: comandi per concatenare due righe del file, comandi per spezzare una riga in due in corrispondenza di una stringa data, ricerche di semplici pattern, ecc.

Alcuni esempi

Mostriamo alcuni esempi di file di comandi, con una descrizione informale della loro funzione. Le istruzioni sono qui date in ordine per chiarezza, ma si ricordi che nel file possono apparire in un ordine qualunque (l'ordine di esecuzione è comunque quello della numerazione).

10 traslittera "aeio" "uuuu" 20 prossima 30 vaia 10	Trasforma il file in sardese, sostituendo tutte le vocali del file con "u".
10 cerca "crisi" 20 cancella 30 vaia 10	Elimina dal file tutte le righe che contengono la parola "crisi" (attenzione: anche come parte di una parola più lunga!)
10 setrovi " crisi " cancella 20 prossima 30 vaia 10	Elimina dal file tutte le righe che contengono la parola "crisi" (attenzione: non riconosce "crisi" ad inizio o fine riga!)
10 inserisci "/* Copyright 2006 */" 20 inserisci "/* Myself & co. */" 30 finefile	Inserisce due righe di commenti in cima ai file indicate sulla riga di comando
5 cambia romanzo contratto 10 setrovi FIRMATO goto 20 15 prossima 17 vaia 5 20 prossima 30 inserisci "Silvio" 40 finefile	Dato un file di testo, cambia tutte le occorrenze di "romanzo" che precedono la prima occorrenza di "FIRMATO" con "contratto". Quando si trova una occorrenza di "FIRMATO", viene inserita una riga (<u>dopo</u> quella contenente "FIRMATO") contenente la parola "Silvio", e si passa al file successivo senza ulteriori modifiche al file corrente.
5 cerca printf 10 premetti "if (debug)" 15 vaia 5	Inserisce un "if (debug)" prima di tutte le righe che contengono una chiamata a printf() (presumibilmente, all'interno di un file sorgente C).

Sintassi

L'editor batch viene invocato con uno o più parametri; il primo parametro (obbligatorio) è il nome del file contenente le istruzioni di editing; i parametri successivi sono i nomi dei file da elaborare. Se è presente un solo parametro, l'editor opera su *stdin*, inviando il file editato su *stdout*.

Gestione degli errori

Tutti gli errori, sia di sistema (fallimento nell'apertura dei file, nella lettura o scrittura, mancanza di memoria, ecc.) che interni (istruzioni errate o impossibili da eseguire) devono essere segnalati da appropriati messaggi su *stderr*. Quando possibile, l'esecuzione deve continuare anche in presenza di errori (per esempio, passando al file successivo se non è possibile aprire un file),

Tempi e modalità di consegna

L'elaborato deve essere consegnato entro il 31 gennaio 2006 (eventuali motivate eccezioni possono essere concordate con il docente) e deve essere costituito da una stampa del codice sorgente, adeguatamente formattato e commentato, da una stampa con esempi di esecuzione, e da una breve relazione scritta (2-3 pagine) che descrive il progetto stesso. Il codice sorgente deve essere anche inviato via email al docente (gervasi@di.unipi.it) contestualmente alla consegna della parte cartacea. Si raccomanda di usare un Makefile contenente le dipendenze e i comandi necessari per la compilazione; alternativamente, si indichino i comandi di compilazione usati nella mail di accompagnamento al codice. Si raccomanda inoltre fortemente di usare il comando UNIX `a2ps`, o altro equivalente, per fare una stampa formattata del codice.

Prima della consegna verificare sulla pagina web del corso (<http://islp.di.unipi.it/LLS05>) l'eventuale presenza di istruzioni più aggiornate riguardo alle modalità e ai tempi di consegna.