L'ambiente UNIX/Linux

I filtri

Stefano Quer
Dipartimento di Automatica e Informatica
Politecnico di Torino

Filtri

- In UNIX/Linux un **filtro** è un comando che
  - Riceve il proprio input da standard input
  - Lo manipola (lo filtra) secondo determinati parametri e opzioni
  - Produce il suo output su standard output
- Sostanzialmente sono comandi che
  - Permettono un qualche tipo di manipolazione di testi

Filtri

- Filtri più comuni
  - awk, cat, cut, compress, grep, head, perl, sed, sort, tail, tr, uniq, wc
- Alcuni di questi comandi
  - Sono discretamente complessi
    - grep, sort
  - Sono dei linguaggi di scripting completi
    - sed, awk
  - Spesso
    - Utilizzano **espressioni regolari**
    - Sono utilizzati come comandi in **pipe** (ovvero utilizzati con l'operatore **|**) con altri

cut

```
cut [opzioni] file
```

- Rimuove sezioni specifiche di ogni riga del file indicato
- Esempi

```
cut -f 1,3 file.txt

cut -f 1-3,5-6 -d " " foo.txt
```

Seleziona i campi 1 e 3 di tutte le righe del file (delimitatore = tab)

Selezione i campi da 1 a 3 (1, 2 e 3) e da 5 a 6 del file, specificando che i campi sono delimitati da uno spazio

cut

Opzioni			
Formato		**Significato**	**Effetto**
Compatto	**Esteso**		
-c LIST	--characters=LIST		Seleziona solo i caratteri di posizione indicata
-f LIST	--fields=LIST		Indica la lista dei campi da selezionare (separati da virgola) Formato: n (=n), -n (≤n), n- (≥n), n1-n2 (≥n1 && ≤n2) Esempi: 3, -3, 3-, 3-5
-d DELIM	--delimiter=DELIM		Usa DELIM per dividere i campi (il delimitatore di default è la tabulazione)

Esercizio

- Si riporti il comando UNIX per effettuare quanto indicato, utilizzando eventuali ridirezioni e pipe
  - Visualizzare quante caratteri sono presenti in tutti i file di estensione ".txt" (senza visualizzare informazioni aggiuntive)

Soluzione

- Si riporti il comando UNIX per effettuare quanto indicato, utilizzando eventuali ridirezioni e pipe
  - Visualizzare quante caratteri sono presenti in tutti i file di estensione ".txt" (senza visualizzare informazioni aggiuntive)

-exec: applica wc (word count) al **contenuto** di tutti i file (\{}) rintracciati con la find

```
find . -name "*.txt" -exec wc \{} \; | cut -f 3 -d " "
```

Seleziona il terzo campo di wc nell'ipotesi (in generale falsa) che i campi siano separate da spazi singoli
(wc → numero righe, stringhe, caratteri e nome file)

tr

```
tr [opzioni] set1 [set2]
```

- Copia lo standard input nello standard output effettuando le sostituzioni oppure le cancellazioni specificate
- Va utilizzato ridirigendo il suo input con l'output di altri comandi

tr

Opzioni			
Formato		**Significato**	**Effetto**
Compatto	**Esteso**		
-c, -C	--complement	Complementa insieme	Utilizza il complement del set_1
-d	--delete	Cancella caratteri	Cancella i caratteri indicate nel set_1
-s	--squeeze-repeats	Elimina ripetizioni	Sostituisce ogni sequenza di un carattere ripeturo incluso nel set_2 con una occorrenza singola dello stesso carattere

All'interno di set_1 e set_2
\num = carattere di codice ASCII num
\n = newline
\\ = backslah

tr

- Esempi

```
tr -d abcd < file.txt

cat file.txt | tr ab BA

echo ciao | tr ia IA

echo cciiaaooccciiiaaaooo | tr -s oaic
```

Visualizza su standard output le righe di file.txt in cui sono stati eliminati i caratteri a, b, c, d

Visualizza su standard output le righe di file.txt in cui 'a' è stato sostituito con 'B' e 'b' con 'A'

Visualizza cIAo su standard output

Visualizza ciaociao su standard output

uniq

```
uniq [options] [inFile] [outFile]
```

- Riporta oppure elimina le righe ripetute nel file in ingresso
  - Formato
  - Richiede che il file sia ordinato
  - Senza opzioni elimina le righe duplicate

uniq

Opzioni			
Formato		Significato	Effetto
Compatto	Esteso		
-c	--count		Stampa il numero di ripetizioni prima della riga
-d	--repeated		Visualizza solo le righe ripetute
-f N	--skip-fields=N		Ignora i primi N campi per il confronto
-I	--ignore-case		Case insensitive

uniq

- Esempi

```
uniq --count file.txt
uniq -c file.txt

uniq -d a.x
```

Elimina le righe duplicate visualizza le rimanenti inserisce il numero di volte che compaiono

Visualizzza le sole righe ripetute

basename

```
basename nome [estensione]
```

- Elimina il direttorio (path) e suffisso (estensione) da un nome di file

basename

- Esempi

```
> basename /home/quer/current/file.txt
file.txt

> basename /home/quer/current/file.txt ".txt"
file

> basename /home/quer/current/file.txt .txt
file

> basename /home/quer/current/file.txt txt
file.
```

sort

```
sort [opzioni] [file]
```

- Ordina i file in input in ordine alfabetico

sort

Opzioni			
Formato		Significato	Effetto
Compatto	Esteso		
-b	--ignore-leading-blanks		Ignora gli spazi iniziali
-d	--dictionary-order		Considera solo spazi e caratteri alfabetici
-f	--ignore-case		Trasforma caratteri minuscoli in maiuscoli (case insesitive)
-I	--ignore-case		Case insensitive
-n	--numeric-sort		Confronta utilizzando un ordine numerico
-r	--reverse		Ordine inverso

sort

Opzioni			
Formato		Significato	Effetto
Compatto	Esteso		
-k c1[,c2]	--key=c1[,c2]		Ordina sulla base dei soli campi selezionati
-m	--merge		Merge file già ordinati (senza riordinare)
-o=f	--output=f		Scrive l'output nel file f invece che su stdandard output

sort

- Esempi

```
sort file.txt
```

Ordina le righe del file file.txt interpretandole come sequenza di caratteri ASCII

```
cat file1.txt file2.txt | \
  sort -r –k 1,3 -f
```

Concatena i file file1.txt e file2.txt e ordina in ordine decrescente le righe dei due file utilizzando i campi 1,2 e 3 e ignorando la differenza tra lettere maiuscole e minuscole

Esercizio

- Si riporti il comando UNIX per effettuare quanto indicato, utilizzando eventuali ridirezioni e pipe
  - Visualizzare tutti i file del direttorio corrente ordinando le righe per dimensione crescente del file

Soluzione

- Si riporti il comando UNIX per effettuare quanto indicato, utilizzando eventuali ridirezioni e pipe
  - Visualizzare tutti i file del direttorio corrente ordinando le righe per dimensione crescente del file

Output di ls -la

```
total 28
drwxr-xr-x 1 quer quer  512 Nov 12 10:17 .
drwxr-xr-x 1 root root  512 Sep 26 16:08 ..
-rw------- 1 quer quer 1669 Oct  8 22:23 .bash_history
-rw-r--r-- 1 quer quer  220 Sep 26 16:08 .bash_logout
...
```

```
ls –la | sort –n –k 5
```

Permangono i direttori ".", ".." e la riga "total"

grep

- Global Regular Expression Print
  - Cerca nel contenuto dei file di ingresso le righe che hanno un "match" con il pattern fornito e le visualizza su standard output
- Il comando esiste in diverse versioni
  - grep
    - Versione standard
  - egrep, fgrep, rgrep
    - egrep equivale a "grep –E"
    - Usa Extended RE nel pattern

grep

```
grep [options] pattern [file]
```

- Opzioni principali

Opzioni			
Formato		Significato	Effetto
Compatto	Esteso		
-e PATTERN	--regexp=PATTERN		Specifica i pattern da ricercare Permette di specificare pattern multipli
-B N	--before-context=N		Prima di ciascun match stampa N righe (oltre alla riga in cui si è trovato il match). Inserisce un separatore (--) dopo ogni insieme stampato.

grep

Opzioni			
Formato		**Significato**	**Effetto**
Compatto	**Esteso**		
-A N	--after-context=N		Dopo ciascun match stampa ancora N righe (oltre alla riga in cui si è trovato il match). Inserisce un separatore (--) dopo ogni insieme stampato.
-H	--with-filename		Stampa il nome del file per ogni match
-I	--ignore-case		Case insensitive
-n	--line-number		Stampa il numero di riga del match
-r, -R	--recursive		Procede in maniera ricorsiva sul sottoalbero
-v	-inverse-match		Stampa solo le righe che non fanno match

grep

- Esempi

```
grep abc file.txt

grep -e "l." -e a file.txt

grep -H -A 4 abc file.txt
```

Stampa tutte le righe del file che contengono "abc"

Stampa tutte le righe che contengono una 'l' seguita da un altro carattere qualsiasi, oppure contengono una a

Stampa tutte le righe che contengono la stringa "abc", stampa tali righe e le 4 successive, facendole precedere dal nome del file

Esercizio

- Si riporti il comando UNIX per effettuare quanto indicato, utilizzando eventuali ridirezioni e pipe
  - Visualizzare tutti i file del direttorio corrente ordinando le righe per ora di creazione decrescente

Soluzione

- Si riporti il comando UNIX per effettuare quanto indicato, utilizzando eventuali ridirezioni e pipe
  - Visualizzare tutti i file del direttorio corrente ordinando le righe per ora di creazione decrescente

```
total 28
drwxr-xr-x 1 quer quer  512 Nov 12 10:17 .
drwxr-xr-x 1 root root  512 Sep 26 16:08 ..
-rw------- 1 quer quer 1669 Oct  8 22:23 .bash_history
-rw-r--r-- 1 quer quer  220 Sep 26 16:08 .bash_logout
...
```

Output di ls -la

```
ls -la | \
  grep -v -e "total" -e "\.$" -e "\.\.$" | \
  sort -n -r -k 8
```

Già incluso nel precedente

Elimina i direttori ".", ".." e la riga "total"

Esercizio

- Si riporti il comando UNIX per effettuare quanto indicato, utilizzando eventuali ridirezioni e pipe
  - Visualizzare tutte le righe dei file di estensione ".txt" che contengono una stringa palindroma
    - Di 3 caratteri (e.g. "aba")
    - Di 5 caratteri (e.g. "abcba")

Soluzione

- Si riporti il comando UNIX per effettuare quanto indicato, utilizzando eventuali ridirezioni e pipe
  - Visualizzare tutte le righe dei file di estensione ".txt" che contengono una stringa palindroma
    - Di 3 caratteri (e.g. "aba")
    - Di 5 caratteri (e.g. "abcba")

Basic Reg Exp

```
grep -e "\(.\).\1" *.txt

grep -e "\(.\)\(.\).\2\1" *.txt
```

Extended Reg Exp (per 5 caratteri)

```
grep –extended-regexp -e "(.).\1" *.txt
grep –E -e "(.)(.).\2\1" *.txt
```

Esame del 29.06.2015

Esercizio

- Si riportino i comandi UNIX per effettuare quanto indicato, utilizzando eventuali ridirezioni e pipe
  - Nel direttorio "/home/foo" cercare i file con il nome che inizia con il carattere "L" e estensione "txt". In questi file ricercare al presenza della stringa "laib". Visualizzare il nome del file e l'intera riga in cui tale stringa viene rintracciata.

Esame del 29.06.2015

Soluzione

- Si riportino i comandi UNIX per effettuare quanto indicato, utilizzando eventuali ridirezioni e pipe
  - Nel direttorio "/home/foo" cercare i file con il nome che inizia con il carattere "L" e estensione "txt". In questi file ricercare al presenza della stringa "laib". Visualizzare il nome del file e l'intera riga in cui tale stringa viene rintracciata.

```
find /home/foo -name "L*.txt" -exec \
  grep -H "laib" '{}' \;
```

Esame del 29.06.2015

Esercizio

- Si riportino i comandi UNIX per effettuare quanto indicato, utilizzando eventuali ridirezioni e pipe
  - Trovare tutti i file di estensione "txt" nel direttorio "/home" memorizzati tra il livello di profondità 3 (incluso) e il livello di profondità 5 (incluso) dell'albero dei direttori e che siano leggibili. Di questi modificare il proprietario in "ugo".

Esame del 29.06.2015

Soluzione

- Si riportino i comandi UNIX per effettuare quanto indicato, utilizzando eventuali ridirezioni e pipe
  - Trovare tutti i file di estensione "txt" nel direttorio "/home" memorizzati tra il livello di profondità 3 (incluso) e il livello di profondità 5 (incluso) dell'albero dei direttori e che siano leggibili. Di questi modificare il proprietario in "ugo".

```
find /home -mindepth 3 -maxdepth 5 \
  -name "*.txt" -readable \
  -exec chown "ugo" '{}' \;
```

Esame del 29.06.2015

Esercizio

- Si riportino i comandi UNIX per effettuare quanto indicato, utilizzando eventuali ridirezioni e pipe
  - Per ogni file di estensione "txt" presente nel direttorio corrente ricavare il nome e il numero di caratteri presenti nel file. L'elenco venga ordinato in base al numero di caratteri in ordine numerico inverso e memorizzato nel file "stat.txt".

Esame del 29.06.2015

Soluzione

- Si riportino i comandi UNIX per effettuare quanto indicato, utilizzando eventuali ridirezioni e pipe
  - Per ogni file di estensione "txt" presente nel direttorio corrente ricavare il nome e il numero di caratteri presenti nel file. L'elenco venga ordinato in base al numero di caratteri in ordine numerico inverso e memorizzato nel file "stat.txt".

Oppure –k 1

```
find . -name "*.txt" \
  -exec wc -c '{}' \; | sort -rn -k 1,1 > stat.txt
```

Esame del 29.06.2015

Esercizio

- Si riportino i comandi UNIX per effettuare quanto indicato, utilizzando eventuali ridirezioni e pipe
  - Un'applicazione C è formata da main.c, f1.c, f2.c e main.h. Scrivere un Makefile con due target
    - Il primo sia in grado di compilare l'applicazione denominando l'eseguibile myapp
    - Il secondo rimuova eventuali file temporanei e sposti l'eseguibile nel direttorio "/user/bin"

Esame del 29.06.2015

Soluzione

- Si riportino i comandi UNIX per effettuare quanto indicato, utilizzando eventuali ridirezioni e pipe
  - Un'applicazione C è formata da main.c, f1.c, f2.c e main.h. Scrivere un Makefile con due target.
    - Il primo sia in grado di compilare l'applicazione denominando l'eseguibile myapp.
    - Il secondo rimuova eventuali file temporanei e sposti l'eseguibile nel direttorio "/user/bin"

<tab>

```
compile: main.c f1.c f2.c
        gcc -o myapp main.c f1.c f2.c
install:
        rm *.tmp
        cp myapp /user/bin
```