Introduzione a Linux e Bash

Linux è un Sistema Operatino basato su UNIX creato da Linus Torvalds.
Oggi il **64% dei server** mondiali gira su qualche variante di UNIX o Linux.

In questa lezione vedremo nel dettaglio alcuni comandi della shell di Linux.

MAN

Comando presente nella maggior parte dei sistemi oerativi unix e unix-like. Ci permette di visualizzare le **Manual Pages**, cioè le pagine di manuale dei singoli comandi.

> **Un pò di storia:**
>
> Pubblicato il 3 Novembre del 1971, ma reso disponibile solo con la settima edizione di UNIX nel 1979, lo *UNIX Programmer's Manial* è stato un grande passo in avanti nelle distribuzioni UNIX.
> Oggigiorno ogni applicazione a riga di comando di UNIX ha una pagina man, l'assenza indica solitamente scarsa qualità del software.
> Alcuni progetti, come Debian, si preoccupano di scrivere pagine per quei comandi che ne sono privi.

Ogni pagina è un documento esplicativo di un comando, funzione o file che descrive im maniera sintetica l'argomento, inoltre le varie pagine sono raggruppate in sezioni omogenee per tipo di argomento trattati.

```
man nome_pagina
```

AWK

awk/mawk è un interprete per il linguaggio di programmazione AWK che può essere usato per manipolare dati, prendere parti di testo, implemetare algoritmi.

> **mawk è la nuova versione.**

Un programma AWK è una sequenza di istruzioni {azioni} e di funzioni.
I programmi corti possono essere scritti direttamente su linea di comando racchiusi in ' ' in modo tale da evitare interpretazioni dalla shell.

```
awk opzioni 'criteri di selezione {azione}' file di input > file di output
```

Un esempio semplice del funzionamento di AWK è il seguente:

Abbiamo il seguente file di testo:

```
ajay account manager 45000
sunil account impiegato 25000
varun account manager vendite 50000
amit account manager 47000
tarun peon vendite 15000
deepak account impiegato vendite 23000
sunil peon vendite 13000
satvik direttore acquisti 80000
```

Se volessimo prendere le righe in cui ci stanno solo i manager e scirverli su un nuovo file di testo possiamo scrivere questo sulla linea di comando:

```
$ awk '/manager/ {print}' dipendente.txt > manager.txt
```

Ovviamente può essere usato anche per molte altre funzioni.

CAT

Possiamo usarlo per concatenare e per stampare file sulla shell:

```
cat file.txt
```

Stampa il file.txt sulla linea di comando, mentre:

```
cat file1.txt file2.txt > file_concat.txt
```

Prenderà i due file alla destra dell'operatore e li unirà in unico file, quello alla sinistra dell'operatore.

CUT

Rimuove delle sezione da ogni riga di un file.

```
cut OPTION file.txt
```

Option è semplicemente il tipo di dati che vogliamo prendere dal file.txt

Esempio:

```
cat -b 1,2,3 dipendente.txt
```

Prenderà i primi tre byte dal testo dipendente.txt, l'ouput sarà:

```
aja
sun
var
ami
tar
dee
sun
sat
```

DIFF

Serve a comparare due file.

```
diff OPTION file1.txt file2.txt
```

GREP

Cerca per dei pattern all'interno dei filex.

```
grep OPTION file1.txt
```

Le opzioni in questo caso sono delle espressioni regolari, il comando stamperà le righe che soddisfano l'espressione, per esempio:

```
grep '[s]' file1.txt
```

Stamperà dal file1.txt tutte le linee che cominciano con 's'.

HEAD & TAIL

Serve a stampare le prime 10 linee di un file.

```
head OPTION file1.txt
```

Anche qui possiamo usare delle opzioni che fanno delle funzioni specifiche.

Tail invece ha il compito opposto:

```
tail OPTION file1.txt
```

LESS

Less è un programma simile a more, ma che permette di eseguire spostamenti all'interno del file sia in avanti che all'indietro. Oltre a ciò *less* non deve leggere l'intero file in ingresso prima di visualizzarlo, quindi è in grado di avviarsi più velocemente di altri editor di testo come *vi*.

```
less file1.txt
```

OD

Il **comando od (octal dump)** in Linux è uno strumento versatile utilizzato per visualizzare il contenuto dei file in vari formati, con l'ottale predefinito. Questo comando è particolarmente utile per il debug degli script, l'esame dei file binari o la visualizzazione di dati non leggibili dall'uomo come il codice eseguibile.

```
od [OPTION] file1.txt
```

Se avessi il seguente file:

```
100
101
102
103
104
105
```

E faccessi:

```
od -b file1.txt
```

Otterrei la rappresentazione ottale del file anteceduto da un offset.

```
0000000 061 060 060 012 061 060 061 012 061 060 062 012 061 060 063 012
0000020 061 060 064 012 061 060 065 012
```

```
0000030
```

SED

E' uno stream editor e serve a fare delle trasformazioni di testo basilari su uno stream in input.

```
sed [OPTIONS] 'SCRIPT' [INPUTFILE...]
```

Un esempio di quello che può fare è: sostituire una parola con un'altra.
Se ho questo file di testo:

```
linux is great os. unix is opensource. unix is free os.
learn operating system.
linux linux which one you choose.
linux is easy to learn.unix is a multiuser os.Learn unix .unix is a
powerful.
```

Ed eseguissi questo comando:

```
sed 's/unix/linux/' prova.txt
```

Andrei a sostituire tutte le prime occorrenze della parola unix di ogni riga con la parola linux.

SORT

Serve a fare il sorting di un determinato file:

```
sort [OPTION] file.txt
```

Le OPTION ci permettono di fare il sorting in maniera diversa.

SPLIT

Serve a dividere file in pezzi:

```
split [OPTION] file.txt
```

Possiamo anche usarlo senza OPTION, in questo caso il file.txt verrà spezzato in tanti file composti da 1000 righe ognuno.

TR

Traduce o elimina caratteri:

```
tr [OPTION] string1 [string2]
```

Esempio:

```
cat prova.txt | tr [:upper:] [:lower:]
```

Ci permette di stampare il file prova.txt dopo aver modificato tutte le lettere da maiuscolo a minuscolo.

UNIQ

Serve per filtrare o omettere linee del file preso in considerazione.

```
uniq [OPTION] [INPUT[OUTPUT]]
```

Nella sua versione stadard, cioè senza OPTION, elimina le linee duplicate.

WC

Serve a contare parole, linee e lettere:

```
wc [OPTION] [FILE]
```

esempio:

```
wc -c prova.txt
```

Stamperà il numero di caratteri presenti nel file prova.txt

TAR

Permette di comprire file in UNIX sotto un unico formato tar:

```
tar [opzioni] [file-archivio] [file o directory da archiviare]
```

Esempio:

```
tar cvf archivio.tar prova.txt
```

Ci permette di creare un archivio, archivio.tar, con all'interno il file prova.txt.