Comando cat

- Il comando cat visualizza il contenuto di un file.
- Provare i seguenti comandi:
- cat testo1.txt
- cat testo2.txt
- cat testo1.txt testo2.txt testo3.txt
- cat testo*
- cat -n testo* (numera righe)
- cat -b testo (numera righe non vuote)

Caratteri speciali

- Per poter indicare più files alla volta si usano dei caratteri speciali.
- Il carattere * sta per qualsiasi sequenza di caratteri
  *c ->qualsiasi file che finisce per c come per esempio
  testo.doc
  prova.c
  c
- Il carattere ? Sta per esattamente un carattere qualsiasi. Quindi cat testo?.txt mostrera i files: testo1.txt testo2.txt e testo3.txt ma non testo99.txt

Ridirezione dell'output

- Molte volte è utile avere il risultato dell'esecuzione di un programma in un file , in modo da poterlo recuperare successivamente. Con l'operatore > diciamo di scrivere il risultato in un file e non a video.
- Con > creiamo un file nuovo o sostituiamo uno esiteste.
- Con >> accodiamo il risultato ad un file esistente, in queto modo non perdiamo il contenuto precente del file.
- Provate il comando cat testo*.txt > tutto.txt

Comando less

È scomodo visualizzarli con cat i files molto grandi, meglio usare il “paginatore” interattivo less.

- less telefoni.txt : visualizza il file telefoni.txt
- Navigazione con tasti freccia e Pag. su e giù.
- (numero riga) e tasto g va sulla riga.
- G (maiuscolo) va sull'ultima riga.
- Ricercare una parola con / “parola”
- N ricerca la precedente corrispondenza (ricerca verso l'alto)
- n ricerca la successiva corrispondenza. (ricerca verso il basso)
- q per uscire.

Comandi head e tail

- head visualizza le prime righe di un file..
- tail visuallizza le ultime righe di un file.
- head -n 20 /var/log/Xorg.0.log visualizza le prime 20 righe
- tail -n 20 /var/log/Xorg.0.log visualizza le ultime 20 righe

Un semplice editor di testo nano

- Per modificare il file telefoni.txt
  nano telefoni.txt
- CTRL +O salva il file
- CTRL +X esce

Comando wc

- Il comando wc sta per word count. E serve a sapere il numero di righe parole e caratteri presenti in un file.
- Provare :
  wc telefoni.txt
  wc -l telefoni.txt

Facciamo lavorare insieme i programmi !

- Oltre a poter salvare il risultato di un programma su un file , si può fare elaborare il suo risultato immediatamente ad un altro programma.
- Tramite l'operatore | chiamato pipe (tubo in inglese) riversiamo il risultato di un programma (output) nell'ingresso di un altro programma (input)
- Ogni programma ha un compito ben specifico ma facendoli lavorare insieme possiamo ottenere il risultato voluto..

Facciamo lavorare insieme i programmi ! Esempio ls con wc

- Cosa vogliamo ? Contare i file con estensione .txt presenti nella directory corrente.
- ls -l *.txt li visualizza ma non li conta !
- Chiediamo aiuto a wc. Ogni file viene stampato su una riga. Quindi basta contare le righe per sapere il numero di files.
- ls -l | wc -l

  il risultato di ls lo diamo in pasto a wc che contando le righe ci dice il numero di files .txt presenti nella cartella.

Comando grep

- Il comando grep serve per ricercare del testo all'interno dei files.
  Provare :
  grep Mario telefoni.txt
  grep mario telefoni.txt
  grep -i Mario telefoni.txt (ricerca senza differenza fra maiuscole e minuscole)

  grep -i ma *.txt (cerca la stringa ma senza distinzionie fra maiuscole e minuscole in tutti i files il cui nome termina per .txt che sono nella directory corrente.

  grep -i c ma *.txt conta le corrispondenze trovate.

Comando ps

Il comando ps mostra i processi correntemente attivi.

- Ha molte opzioni ma quelle più utili sono le seguenti

  ps aux

  ax specifica di mostrare tutti i processi

  u è il tipo di visualizzazione da utilizzare (u sta per user oriented)

- PID e un codice che identifica univocamente un processo
- Molto utile utilizzare ps insieme a grep per trovare subito il programma che ci interessa. Esempio:

  ps aux | grep firefox

“Uccidere un programma cattivo”

- Se un programma va in una condizione anomala e non riusciamo più a chiuderlo correttamente possiamo terminarlo con il comando kill.
- La sintassi è : kill -9 “pid_del-programma” .
  Per sapere il pid usare ps.
- Perchè -9 ? In realtà kill può mandare vari messaggi ai programmi. Il -9 indica quindi il tipo di messaggio da inviare al programma .I tipi di messaggi si possono vedere con kill -l e hanno vari scopi. Kill -9 è il meno gentile di tutti ma funziona sempre!

“terminare un programma senza il pid”

- Se vogliamo terminare un programma senza conoscerne il pid , ma conoscendo solo il nome del suo eseguibile possiamo usare il programma killall
  killall -9 nomeProgramma
- ATTENZIONE: come il nome del comando ci fa capire, esso non identifica in maniera univoca un processo in esecuzione ma terminerà tutti i processi generati da quel file eseguibile.
  Per esempio se abbiamo due Gedit aperti, di cui uno bloccato ed uno no . Con il comando 'killall -9 gedit' termineremo entrambi, con eventuale perdita di dati.

“top Gestire i processi in maniera interattiva”

- top visualizza una breve sintesi delle risorse utilizzate dal sistema e la lista dei processi presenti.
- Tasto O per selezionare campo in base al quale effettuare l'ordinamento
- Tasto R per ordinamento inverso
- Tasto k per immettere il pid di un processo da uccidere
- Tasto h visualizza help
- Tasto q per uscire.
- Versione più intuitiva e moderna htop.

“Disk free ”

- Il comando df (disk free) serve per conoscere lo spazio libero sui dischi.
- df mostra lo spazio disco in kilobyte.
- df -h mostra le dimensioni con il multiplo più opportuno (M =Mega G=Giga T=Tera)

  h sta per human readable cioè facilmente leggibile dall'uomo.

“Disk Usage ”

il comando du serve per conoscere lo spazio effettivamente occupato dai vari files . Risulta molto utile per trovare i files grandi da cercare di eliminare in caso di mancanza di spazio.

- du mostra lo spazio occupato da ogni sotto cartella in kilobyte.
- du -h mostra la dimensione in: K=Kilo, M =Mega G=Giga, T=Tera
- du -h * mostra anche i singoli files

-S non conteggia lo spazio occupato dalle sotto cartelle

-c calcola il totale .

“Sort ”

sort serve per ordinare un file o l'output di un altro progamma in base ad una “colonna”, se non specificata prende la prima.

- I seguenti parametri determinano l'ordinamento:
  -b ordine alfabetico ignorando gli spazi
  -d ordine alfabetico considerando gli spazi
  -f ignora differenza maiuscole minuscole
  -r ordine inverso (dal più grande al più piccolo)
  -n ordine numerico
  -k indica la colonna per la quale ordinare -k1 la prima -k2 la seconda... ecc
  -u elimina le righe duplicate.
  -h ordina in base a human readable

“Ersercitazioni con sort ”

- Ordinare il file telefoni.txt per nome.
- Ordinare il file telefoni.txt per numero di telefono.
- Creare un nuovo file tel.txt ordinato per cognome.
- Vedere quali sono i files più grandi che si trovano sotto una qualsiasi cartella della propria home .
- Ordinare l'output di df in base alla percentuale d'uso.

Domande


Terminale
File Modifica Visualizza Terminale Vai Aiuto
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?
Domande ?


GULP

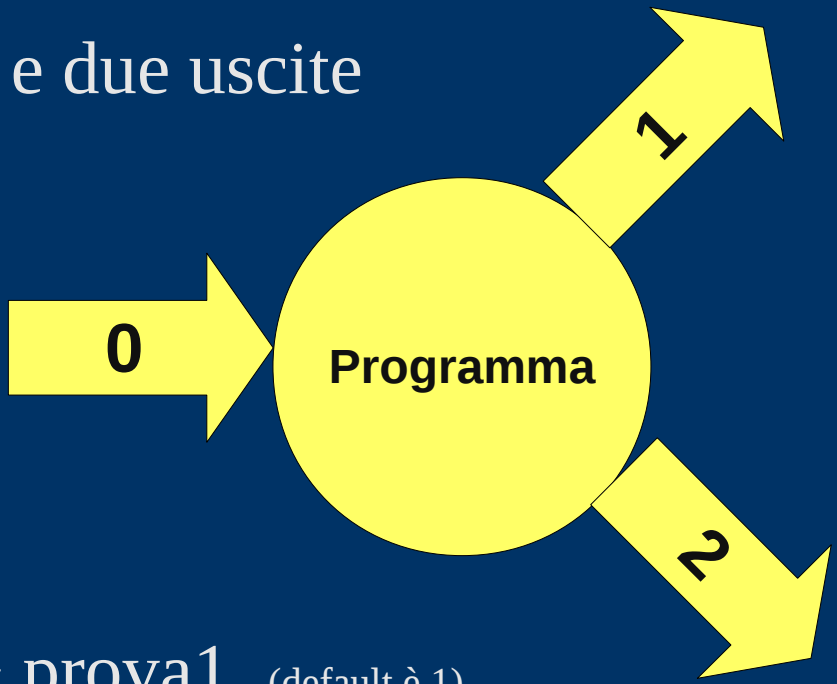
Bash *le ridirezioni*

- Un programma ha un ingresso e due uscite

  standard input tastiera 0

  standard error video 2

  standard output video 1



- Ridirezione >
- Provare ls “file inesistente” > prova1 (default è 1)
- Provare ls “file inesistente” 2 > prova2
- Provare ls “file inesistente” &> prova3

  > crea sostituisce >> accoda

“Il comando find”

- find -name '*stringa*' cerca tutti i files che nel nome contengono stringa e che sono contenuti in una qualsiasi cartella sotto a quella corrente.
- find -iname '*stringa*' cerca tutti i files che nel nome contengono stringa, SENZA distinzione di MAIUSCOLE-MINUSCOLE ,e che sono contenuti in una qualsiasi cartella sotto a quella corrente
- find -size +1M (trova i file più grandi di un mega) .

  find -mmin 10 modificati negli ultimi 10 minuti

  find -mtime 5 modificati negli ultimi 5 giorni
  (5*24 ore da ora ...mtime 0 da meno di un giorno)
-

“Espressioni regolari ”

simbolo	significato
. (punto)	Un carattere qualsiasi
[blf]	Uno dei carattere compreso fra le quadre In questo caso o b o l o f
[blf]*	Come sopra ma ripetuti da 0 a n volte
[A-Z]	Tutte le lettere maiuscole
[A-Za-z]	Tutte le lettere
^	Inizio riga
$	Fine riga
[:space:]	Qualsiasi spazio
[Nn]+	Qualsiasi sequenza di N o n ma Presente almeno una volta
^root	La parola root ad inizio riga
root$	La parola root a fine riga
[mM]ari[ao]	Cerco la parola mario o maria con o senza la lettera maiuscola