LABORATORIO DI SISTEMI OPERATIVI

Corso di Laurea in Ingegneria Informatica

A.A. 2021/2022

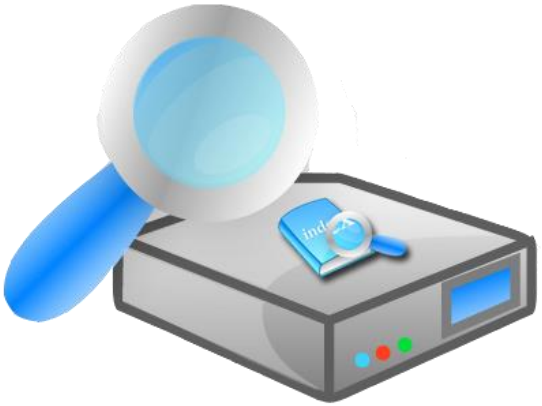
Ing. Domenico Minici


domenico.minici@unifi.it

ESERCITAZIONE 4

Strumenti per la gestione dei file

RICERCA DI FILE



find

- Strumento molto potente per trovare file
  - La sintassi è relativamente complessa
  - Permette di effettuare la ricerca combinando dei test sulle proprietà dei file:
    - Filename
    - File type
    - Owner (user e/o group)
    - Permessi
    - Timestamp
  - Le ricerca non è influenzata dal contenuto del file
  - E’ possibile eseguire comandi (actions) sui file trovati

find

```
find [path1 path2…] [espressione]
```

- Path
  - E’ possibile specificare uno o più percorsi (path) separati da spazio. La ricerca verrà effettuata solo nei percorsi specificati
- Espressione
  - Descrive come vengono trovati i file, e quali azioni devono essere eseguite su di essi

find – espressioni

- Le espressioni sono composte da una sequenza di elementi:
  - **Test**
    Valutazione di una proprietà dei file, può ritornare true o false
  - **Azioni**
    Azioni da effettuare sui file "trovati" (ad esempio eseguire un comando). Ritornano true se hanno successo.
  - **Opzioni globali**
    Influenzano l'esecuzione di test o azioni. Ritornano sempre true.
  - **Opzioni posizionali (positional options)**
    Influenzano solo le azioni o i test che seguono. Ritornano sempre true.
- Gli elementi di una espressione sono collegati da **operatori**
  - -o indica OR, -a indica AND
  - Se nessun operatore è specificato, l'utilizzo dell'operatore AND è implicito per collegare due espressioni
  - ! può essere usato per negare un espressione (NOT)

find – test

- `-name pattern`
  - Ricerca basata sul nome del file (non sul path!)
  - pattern può includere i metacaratteri *, ? oppure le parentesi []
    - Esempio 'm[ao]re' porta a trovare 'mare' e 'more'
  - E' necessario scrivere i pattern fra apici per evitare che la shell "espanda" i metacaratteri:
    - `echo '*'`
    - `echo *`
- `-type [dfl]`
  - Tipo di file (d directory, f regular file, l symbolic link)

find – test

- `-size [+-]n[ckMG]`
  - Ricerca basata sulla dimensione del file
  - Il prefisso `[+-]` indica se il file deve essere maggiore o minore della dimensione specificata
  - n indica la quantità di spazio occupata dal file
  - `[ckMG]` indica l'unità di misura utilizzata, rispettivamente byte, kilobyte, megabyte, gigabyte

find – test

- `-user user`
  - Il file appartiene a user?
  - L'utente può essere specificato come username o UID
- `-group group`
  - group è il group owner?
  - Il gruppo può essere specificato come group name o GID
- `-perm [-/]mode`
  - Test basato sui permessi del file (modalità ottale o simbolica)

`mode`	i permessi devono essere esattamente quelli specificati
`-mode`	almeno i permessi indicati devono essere presenti
`/mode`	almeno uno dei permessi indicati deve essere presente

find – azioni

- `-delete`
  - I file trovati vengono eliminati
  - Ritorna true in caso di successo (i file sono stati eliminati senza errori)
  - Attenzione! Se scriviamo –delete prima dei test, verranno eliminati tutti i file!

- `-exec command ;`
  - Esegue il comando specificato sul file considerato (se ha superato i test precedenti)
  - Tutti gli argomenti specificati dopo command vengono considerati come argomenti del comando, fino al carattere ';'
  - La stringa '{}' è utilizzata per indicare il nome del file attualmente processato
  - Il comando viene eseguito a partire dal percorso di partenza
    - Utilizzare -execdir per eseguire il comando a partire dal path del file trovato

find – esempi

```
find path -name 'prova*' ! -type d
```

- Ricerca i file il cui nome inizia con 'prova' e che NON sono directory
- E' necessario utilizzare gli apici oppure il carattere di escape '\' prima di * per "proteggerlo" dalla shell

```
find path ! -name '*.csv' -size +50M
-execdir ls -l {} \;
```

- Cerca i file con
  - Estensione diversa da ".csv"
  - Dimensione superiore a 50 Megabyte
- Ai file trovati viene applicato il comando ls -l
  - E' necessario usare il carattere di escape '\' per "proteggere" \; dalla shell

find – esempi

`find path -perm -664`

- Ricerca file per cui valgono almeno questi permessi
  - Lettura e scrittura per owner e group owner
  - Lettura per gli altri
- Vengono trovati anche file con permessi in più oltre a questi (ad esempio se anche gli altri utenti hanno permesso in scrittura)

`find path -perm /u=w,g=w`

- Cerca i file che possono essere scritti da almeno uno fra owner e group owner

locate

`locate [options] file1…`

- Ricerca il/i file specificato/i
- Sfrutta un database aggiornato periodicamente dal sistema
  - L'aggiornamento del database può essere forzato con il comando updatedb (richiede privilegi di root)

- find vs locate
  - locate è più semplice da utilizzare e più veloce
  - find è un comando standard presente in tutti i sistemi Unix/Linux
  - find dà sempre risultati aggiornati (non dipende dall'aggiornamento di un database)
  - find permette di definire test e azioni

RICERCA DI TESTO NEI FILE

grep

- Il comando grep (general regular expression print) permette di cercare in uno o più file di testo le linee (righe) che corrispondono ad espressioni regolari o stringhe letterali

```
grep [opzioni] [-e] modello [-e modello2…] file1
   [file2…]
```

  - Se si vuole specificare più di un modello (stringa o espressione regolare) si deve utilizzare –e prima di ciascun modello (incluso il primo)

grep – opzioni

Opzione	Significato
-i	Ignora le distinzioni fra maiuscole e minuscole
-v	Mostra le linee che non contengono l'espressione
-n	Mostra il numero di linea
-c	Riporta solo il conteggio delle linee trovate
-w	Trova solo parole intere
-x	Linee intere

grep – espressioni regolari

- E' possibile specificare dove la stringa/espressione deve trovarsi all'interno di un riga
  - `'^'` l'espressione deve trovarsi ad inizio riga
  - `'$'` l'espressione deve trovarsi in fondo alla riga

- Esempi:
  - `'^stringa'` Righe che iniziano con 'stringa'
  - `'stringa$'` Righe che terminano con 'stringa'
  - `'^stringa$'` Righe che contengono solo 'stringa'
  - `'^$'` Righe vuote

grep – espressioni regolari

- Le parentesi quadre permettono di definire set di caratteri ammessi
- Esempio:

```
grep '1[23]:[0-5][0-9]' file
```

  - Il primo carattere deve essere '1'
  - Il secondo può essere '2' o '3'
  - Il terzo deve essere ':'
  - Il quarto deve essere una cifra tra '0' e '5'
  - Il quinto fra '0' e '9'

grep – espressioni regolari

- `'.'` indica qualsiasi carattere

  `'...cept'` Riconosce sia ‘accept che ‘except’

- `'*'` indica che l’espressione che precede può essere ripetuta zero o più volte

  `'[A-Za-z]*'` Riconosce zero o più caratteri alfabetici

- `'\'` è il carattere di escape

  `'^[0-9].*\.$'` Riconosce righe che iniziano con una cifra e terminano con un punto

ARCHIVIAZIONE E COMPRESSIONE

Archiviazione e compressione

- Il comando tar (Tape ARchive) permette di archiviare/estrarre una raccolta di file e cartelle

```
tar modalità[opzioni] [file1…]
```

  - La modalità specifica il modo in cui il comando deve operare (ad esempio creare un archivio, o estrarre un archivio già esistente)
  - Le opzioni permettono di fornire ulteriori dettagli sul comportamento di tar (ad esempio specificare la tecnica di compressione ed il nome dell'archivio)
  - La lista di file/cartelle indica quali file/cartelle devono essere archiviati o estratti (in base alla modalità)

Archiviazione e compressione

- Il formato del file creato dipende dalla compressione (eventualmente) utilizzata
  - `.tar` se non è stata utilizzata compressione
  - `.tar.gz` se l'archivio è stato compresso con gz
  - `.tar.bz2` se l'archivio è stato compresso con bzip2

tar – azioni

- Subito dopo il comando tar, deve essere specificata la modalità in cui operare

Simbolo modalità	Significato
A	Aggiungi file tar all'archivio
c	Crea un nuovo archivio
d	Trova le differenze fra l'archivio ed il file system
--delete	Cancella file dall'archivio
r	Aggiungi file all'archivio
t	Elenca i file di un archivio
u	Aggiungi file all'archivio, ma solo se differiscono dalla copia eventualmente già presente
x	Estrai file dall'archivio

tar – opzioni

- Le opzioni permettono di definire meglio il modo in cui il comando tar deve operare

Simbolo modalità	Significato
v	Verbose
z	Compressione con gzip
j	Compressione con bzip2
f	Permette di specificare il nome dell'archivio

- Consultare "man tar" per la lista completa delle opzioni

tar – esempi

`tar cvf archivio.tar percorso`

- Crea un archivio di nome "archivio.tar" con il contenuto di "percorso"
- Modalità verbose

`tar czf archivio.tar.gz percorso`

- Crea un archivio compresso "archivio.tar.gz"
- Usa la compressione gz

`tar tf archivio.tar`

- Mostra il contenuto di "archivio.tar"

`tar xvf archivio.tar file`

- Estrae "file" da "archivio.tar"
- Modalità verbose

gzip/gunzip e bzip2/bunzip2

- Se si devono comprimere file o archivi creati precedentemente con tar, è possibile utilizzare

`gzip file1 file2 …`

  - I file elencati vengono compressi e salvati in file con lo stesso nome ed estensione .gz. I file non compressi vengono eliminati

`gunzip file1.gz file2.gz …`

  - Estrae i file compressi specificati in file con lo stesso nome (senza l'estensione relativa alla compressione). I file compressi, dopo essere stati estratti, vengono eliminati

- `bzip2` e `bunzip2` utilizzano la stessa sintassi, ma comprimono/decomprimono con l'algoritmo bzip2

ESERCIZI

Esercizio 1 – find

- Trovare i file nella cartella `/usr/bin` che hanno il bit SUID attivo (`u=s`)
- Trovare tutte le cartelle contenute in `/etc` che hanno nel nome la stringa `'sys'`
- Trovare tutti i file con estensione `'.txt'` in `/usr/share/docutils` con dimensione superiore a 10 Kilobyte
  - Per ogni file trovato fare in modo che venga mostrato l'output di `ls -l` eseguito dal path in cui si trovano i file

Esercizio 2 – grep, espressioni regolari

- Trovare in `/etc/passwd` le righe che contengono `'studenti'`
- Trovare in `/etc/group` la riga che descrive il gruppo `'studenti'` (fare in modo che non vengano mostrate le altre righe contenenti la parola 'studenti')
- Cercare nel file GPL-3 (per trovarlo utilizzare locate) le righe contenenti una lettera minuscola fra parentesi tonde
- Trovare in `/etc/passwd` le righe relative a UID da 102 a 105

Esercizio 3 – archiviazione

- Nella propria directory home, creare una cartella es3
- Dentro es3, creare un archivio `conf.tar.bz2`, contenente tutti i file in `/etc` con estensione `.conf`, ed utilizzando la compressione bzip2
- Mostrare i file contenuti nell'archivio
- Estrarre tutti i file dall'archivio – dove vengono salvati?
- Cercare nel manuale l'opzione per estrarre un archivio in una directory specifica
- Creare una sottocartella output, ed estrarre il file `etc/resolv.conf` in output