Sistemi operativi

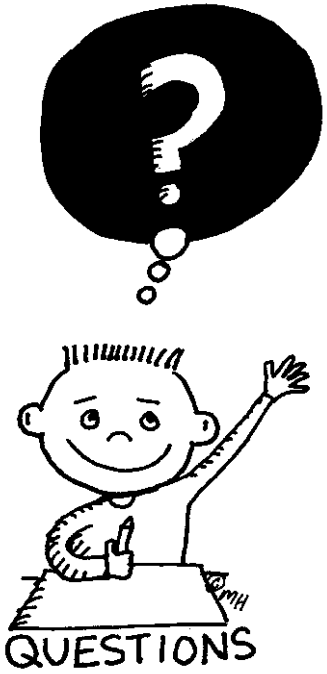
Corso di Laurea Triennale in Ingegneria Informatica

Lezione 4

- Cenni a Xserver
- Patch, diff
- Tar, gzip, bzip
- Find, grep

Fabio Del Vigna - fabio.delvigna@for.unipi.it

Domande sulle lezioni passate?



Soluzione esercizi passati 1/3

- **Login con root**
  - `adduser -s` **(pippo)**
  - `vipw` **: aggiungere la riga (es.):**
  - `pluto::1002:1002::0:0:Pluto:/tmp/pluto:/bin/sh`
  - **Attenzione: scegliere come ID di utente e gruppo un valore non in uso**
  - `vi /etc/group` **: aggiungere la riga (es.): (**`vigr`**)**
  - `pluto:*:1002:`
  - **Attenzione: scegliere come ID di gruppo lo stesso indicato nel punto 1.**
  - `mkdir /tmp/pluto`
  - `cp /etc/skel/* /tmp/pluto`
  - `chown -R pluto:pluto /tmp/pluto`
  - `chmod -R u+rwx,g+rx,o+rx /tmp/pluto`
  - **loggarsi con pluto e cambiare password (comando** `passwd`**).**

Soluzione esercizi passati 2/3

- **pluto** non può creare file nella home di **pippo** con i permessi default. Quindi, dato che non appartengono allo stesso gruppo, bisogna aggiungere il permesso in scrittura per la sezione **others**
  - `chmod o+w /home/pluto`
  - `chmod o+w /home/pippo`
- Una soluzione migliore consiste nel definire un gruppo a cui appartengono **pluto** e **pippo** e assegnare tale gruppo alle home dei due utenti e abilitare il diritto di scrittura (**group owner**) sulle due cartelle.
  - `su root` (**oppure logout, login con root**)
  - `vi /etc/group` : **aggiungere la riga (es.):**
  - `floppyusr:*:1003:pluto`

Soluzione esercizi passati 3/3

- **Le nuove impostazioni sui gruppi diventano effettive al nuovo login, come si può osservare dalla seguente sequenza di comandi:**
  - `id`
  - **logout**
  - **login con pluto**
  - `Id`
- **L'output prodotto dal comando `id` è diverso nei due casi (solo nel secondo viene mostrato il gruppo al quale pluto è stato aggiunto).**
- `su root` **(oppure logout, login con root)**
  - `mkdir /tmp/floppy`
  - `chown :floppyusr /tmp/floppy`
- `chmod g+rwx,o-rwx /tmp/floppy`

X Window System

- creato dal MIT nel 1984.
- fornisce l'ambiente e i componenti di base per le interfacce grafiche:
  - disegno e spostamento delle finestre sullo schermo
  - l'interazione con il mouse e la tastiera
- **non** gestisce l'interfaccia grafica utente o lo stile grafico delle applicazioni
  - questi aspetti sono gestiti direttamente da ogni singola applicazione
- trasparenza di rete

X Window System (2)

- X usa un modello client/server:
  - il server X comunica con gli altri programmi (client)
  - Il server accetta richieste per output grafici (finestre) e input dall'utente (dalla tastiera, il mouse o dal touchscreen).
- Il server X può essere:
  - un programma di sistema che controlla l'output video di un PC
  - un componente hardware dedicato
  - un'applicazione che mostra dati su una finestra di un altro sistema grafico.

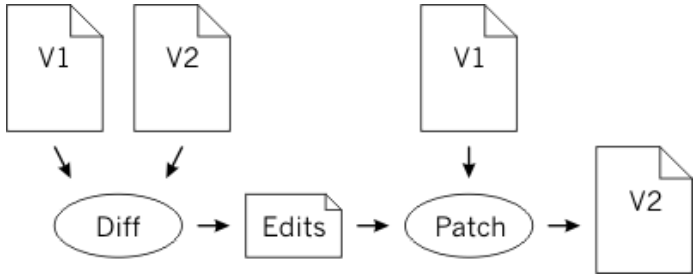
X Window System (3)

- Il protocollo di comunicazione tra server e client opera in modo trasparente rispetto alla rete:
  - entrambi possono risiedere sulla stessa macchina o su altre, anche con architetture e sistemi operativi differenti
  - il server ed il client possono anche comunicare in modo sicuro attraverso la rete sfruttando un tunnel cifrato (SSH).

Patch, Diff e Make

- Nei sistemi Unix sono disponibili strumenti per lo sviluppo di progetti software:
  - Make, per costruire programmi a partire da un insieme di sorgenti
  - Patch e Diff, per confrontare e modificare (patchare) file testuali
  - Programmi per gestire lo sviluppo concorrente di codice (SVN, CVS, Git e simili)

Diff e Patch


V1
V2
V1
Diff
Edits
Patch
V2


EASY DEPLOY THIS WEEK
JUST A SMALL SINGLE-PATCH DIFF
memegen.com

Patch

- Si utilizza per modificare codice in maniera automatizzata.
- Esempio di uso:
  - Un software, sotto forma di codice sorgente, non gira su un sistema .
  - Le modifiche sono distribuite sotto forma di “patch” al codice originale.
  - Con patch si può patchare il codice originale.

Esempio di uso patch

`patch < file.patch` → patch “unified”

`patch file.txt < file.patch`

`patch -R < file.patch`

Attenzione ai livelli (gerarchia cartelle) del file di patch che si ha nel caso si applichino patch a intere cartelle (opzione -p[level]).

Diff 1/2

- Diff serve per confrontare file.
- Confronto “manuale” visivo
- Generazione file patch.

Diff 2/2

- **`diff file1 file2`**
  - differenze (riga per riga) tra i due file

```
(~/src/mutt/mutt)> cvs diff -r3.22 init.h | colordiff
Index: init.h
===================================================================
RCS file: /home/roessler/cvs/mutt/init.h,v
retrieving revision 3.22
retrieving revision 3.25
diff -r3.22 -r3.25
206a207,208
>     ** .dt %C   .dd charset
>     ** .dt %c   .dd reqiures charset conversion (n or c)
216a219
>     ** .dt %T   .dd graphic tree characters
590a594,600
>     { "forward_edit",    DT_QUAD, R_NONE, OPT_FORWEDIT, M_YES },
>     /*
>     ** .pp
>     ** This quadoption controls whether or not the user is automatically
>     ** placed in the editor when forwarding messages.  For those who always want
>     ** to forward with no modification, use a setting to ``no''.
>     */
1271c1281
<     ** If ``yes'', always attempt to verify PGP/MIME or S/MIME signatures.
---
>     ** If ``yes'', always attempt to verify PGP or S/MIME signatures.
(~/src/mutt/mutt)>
(~/src/mutt/mutt)>
```

Esempio di uso diff

- `diff -u file_orig.c file_nuov.c > orig.patch`
- `diff -ruN cart_orig/ cart_nuov/ > orig.patch`

- L'opzione -u crea il file patch in formato unificato (unified).

Archiviazione e compressione

Fabio Del Vigna - fabio.delvigna@for.unipi.it

Archiviazione - `tar`

- tar = Tape ARchive
- Un file `tar` è una raccolta di file e/o directory
- `tar [azione][switch] [archivio] [file]`
- `tar` compresso (`.tgz` o `.tar.gz`) è diventato standard per il passaggio di dati tra sistemi Unix.

`tar` - Opzioni

- azione
  - c : crea
  - x : estrae
  - t : visualizza il contenuto di un archivio
- switch
  - v : verbose
  - z : comprime con gzip
  - j : comprime con bzip2
  - f : file
    - come ultima opzione
  - n : comportamento non ricorsivo

`tar` - Creazione

- `tar -cvf archivio.tar sorgente`
- `tar -czvf archivio.tar.gz ~/archivio/*`
- `archivio.tar`: file da creare
- `sorgente`: contenuto dell'archivio

`tar` - Visualizzazione

- `tar -tvf filename.tar`

- Elenca il contenuto di filename.tar

`tar` - Estrazione

- `tar -xvf archivio.tar destinazione`
- `tar -xzvf archivio.tar.gz`
- Non rimuove l'archivio, ma crea copie del suo contenuto

gzip and gunzip

- Programma di compressione
  - `gzip archivio archivio.gz`
- `gzip`
  - Utility di compressione
- `gunzip`
  - Utility di decompressione

`bzip2` and `bunzip2`

- `bzip2`
  - Utility di compressione
- `bunzip2`
  - Utility di decompressione

- **`bzip2 archivio archivio.bz2`**

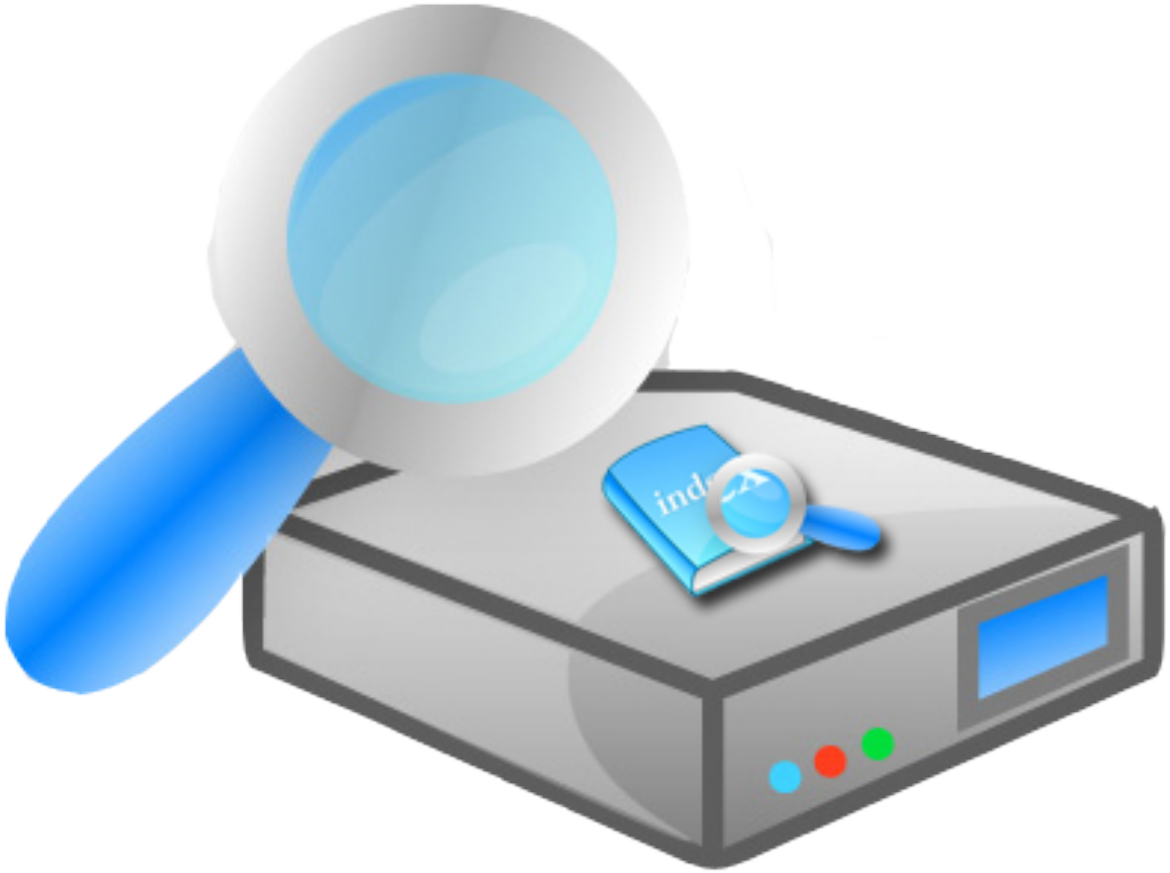
- Rapporto di compressione migliore di `gzip`

Archivi - **path**

- I file vengono memorizzati col **path**
- Non vengono effettuati controlli di overwrite

- `-C nome_directory`: estrae i dati nella directory specificata
- Se non si specifica la destinazione si usa la directory corrente ed il path nell'archivio

Ricerca di file



Fabio Del Vigna - fabio.delvigna@for.unipi.it

find

- Ricerca all'interno di percorsi i file secondo le condizioni indicate

- **Le condizioni sono legate all'apparenza esterna e non al contenuto**

find

- `find [percorso…] [espressione]`

- **percorso…**
  - insieme di percorsi separati da spazi
- **espressione**
  - insieme di test e azioni separati da operatori

Espressioni

`[opzione…] [condizioni]`

- Opzione
  - Altera il comportamento del programma
- Condizioni
  - Espressioni con risultati logici

- Per concatenare più condizioni si usano gli operatori booleani
- default è l'AND logico (-and)

Condizioni

- **-name schema**
  - TRUE se il nome del file corrisponde
  - si possono utilizzare i caratteri * e ?
- **-size [+|-]n[b|c]**
  - TRUE se la dimensione del file è uguale (maggiore + o minore -) a n unità di spazio
- **-type t**
  - TRUE se il file è del tipo specificato (d,f,l)

Condizioni

- **-user utente**
  - TRUE se la risorsa appartiene a utente
- **-group gruppo**
  - TRUE se appartiene al gruppo ”gruppo”
- **-perm [-/+]modello**
  - TRUE se i permessi corrispondono esattamente al modello specificato (in forma ottale o simbolica)

Operatori booleani di `find`

- **(espressione)**
  - Precedenza nell'esecuzione dei test
- **!espressione**
  - nega un'espressione
- **espressione [-and] espressione**
  - AND logico tra espressioni (può essere omesso)
- **espressione -or espressione**
  - OR logico tra espressioni

Usare i caratteri di escape per proteggere dall'espansione della shell

Azioni di `find`

- Operazioni da compiere per ogni risultato della scansione.
  - **-print**
    - stampa i nomi dei file trovati
  - **-exec comando [{}] [;|+]**
    - esegue **comando** sui risultati della ricerca
    - esegue una istanza del **comando** per ogni file trovato (la riga terminata da ;)
    - con {} i file trovati vengono passati a **comando** come lista (la riga terminata da +)

Esempi `find`

```
find . –name prova\* -print
```

Cerca i nomi che iniziano con prova

```
find / -name “lib*” -print
```

Ricerca in / i nomi iniziano per lib

“\” protegge i metacaratteri

Esempi `find`

```
find /home –name "pro*" ! –type d
```

- Escluse le directory
- Ricerca a partire da /home i nomi che iniziano con pro
- Virgolette usate per evitare che la shell trasformi pro* in qualcosa di diverso

Ricerca con `locate`

`locate "nomefile"`

- Esegue una ricerca del file "nomefile" all'interno di un database
- Il database va popolato e tenuto aggiornato con il comando: **`updatedb`**

Ricerca nei file

grep

grep

- **grep [opzioni] "stringa" nome_file ...**
  - Cerca le righe contenenti **stringa**
- **grep "pippo" pluto**
  - Visualizza le parti del file `pluto` in cui compare la parola `pippo`.
- **grep pippo ***
  - Cerca la parola `pippo` in tutti i file della directory corrente.

Il comando `grep` è case sensitive.

grep

- È possibile utilizzare espressioni regolari per le ricerche (**g**eneralized **r**egular **e**xpression **p**rinter)
- `grep 'ri.*o' pluto`
- Cerca le stringhe che iniziano per `ri` e terminano con `o` all'interno del file `pluto`.
- `.*` indica 0(zero) o più caratteri qualunque

Opzioni `grep`

`-i` (ignore case)

ignora le distinzioni tra minuscole e maiuscole

`-v`

mostra le linee che NON contengono l'espressione

`-n`

mette il numero di riga davanti ad ogni riga che riporta

`-c`

riporta solo il conteggio delle linee

`-w`

verifica solo parole intere

`-x`

controlla le corrispondenze di linee intere

Metacaratteri (1 di 2)

- ^ Inizio riga

```
grep '^d' ls.out
```

- $ Fine riga

```
grep '\.c$' ls.out
```

- Aggiungere anche il metacarattere '\' perché anche il carattere '.' è un metacarattere
- '\' neutralizza il valore di metacarattere del carattere che lo segue
- Per citare letteralmente il carattere '\' è necessario quindi scriverlo due volte: '\\'

metacaratteri
jolly
wild-cards

Metacaratteri (2 di 2)

.	uno ed un solo carattere qualunque
*	zero o più occorrenze dell'espressione che lo precede

Esempi:

- Ricercare una riga costituita dalla sola stringa "riga completa":

```
'^riga completa$'
```

- Individuare tutte le righe vuote del file:

```
'^$'
```

Insiemi di caratteri

[s]

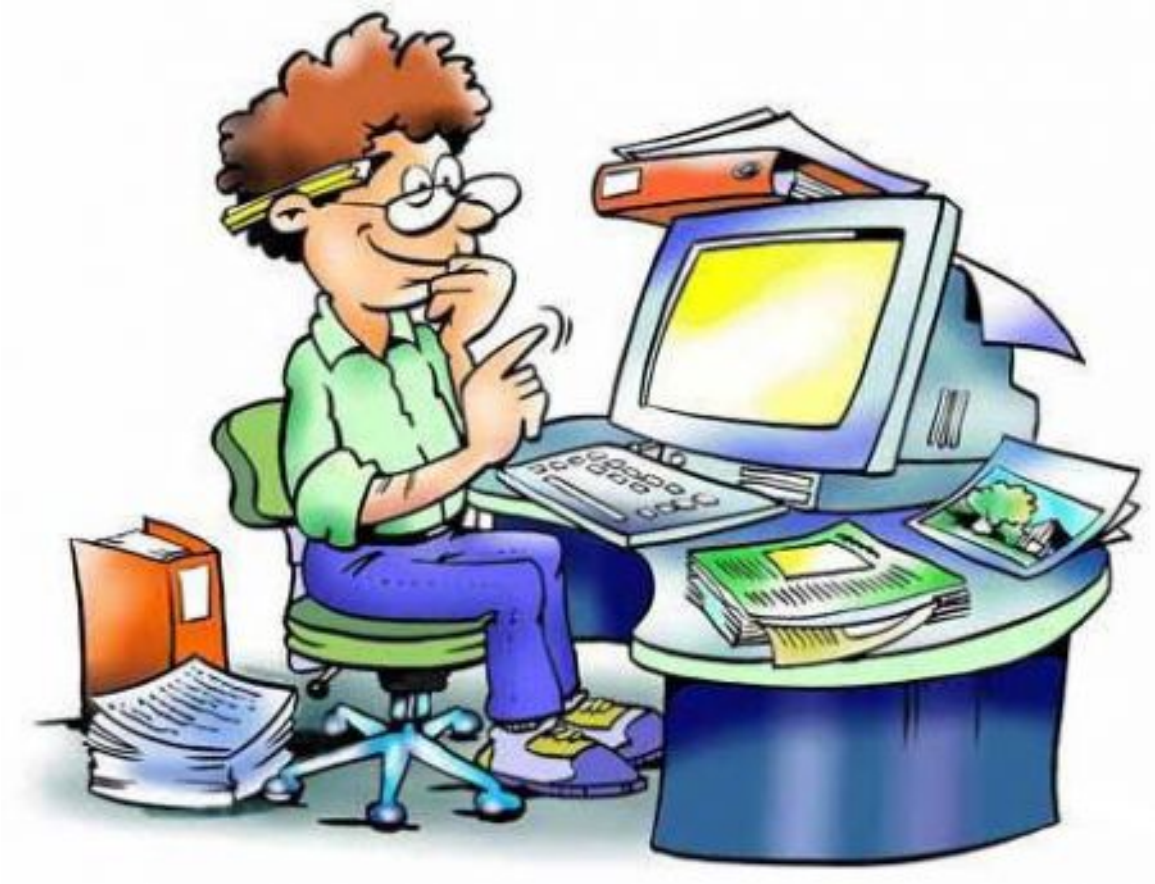
- '[' e ']' sono metacaratteri
- 's' elenco di caratteri ammessi.
- L'insieme '[s]' soddisfa UN SOLO qualunque carattere compreso nell'elenco s.
- Negli insiemi '[s]' si possono specificare intervalli di caratteri usando il carattere '–'

Esempio

```
grep '1[2,3]:[0-5][0-9]' ls.out
```

- il primo carattere della stringa cercata deve essere letteralmente un '1'
- il secondo può essere un '2' o un '3'
- il terzo deve essere letteralmente un ':'
- il quarto può essere '0' o '1' o '2' o '3' o '4' o '5'
- il quinto deve essere una cifra compresa tra '0' e '9' (estremi compresi).

Esercizio



Esercizio 1

- Creare due file di testo, con contenuto di almeno 5 righe (chiamarli argv1.sh e argv2.sh).
- Diff
  - Creare un file patch che permetta, se applicato a argv1.sh di ottenere lo stesso contenuto di argv2.sh
- Patch
  - Applicare la patch creata con diff al file argv1.sh
  - Rimuovere la patch da argv1.sh

Esercizio 2

- Archiviazione e compressione
  - creare nella `home` un archivio compresso in formato `gzip` di nome `config.tgz` contenente i file con estensione `conf` presenti nella cartella `/etc`
  - mostrare i file contenuti nell'archivio
  - decomprimere l'archivio con `gunzip`
  - estrarre i file contenuti con il comando `tar`
- Ricerca dei file
  - cercare dentro la cartella `/etc` tutti i file il cui nome contiene la stringa `sys` e la cui dimensione è superiore a `10 byte`
  - cercare nella `root` tutti i file che hanno il bit `SUID` o `SGID` attivo
  - concatenare e mostrare a video tutti i file (a partire dalla `root`) il cui nome contiene la stringa `tab`
- Ricerca nei file
  - cercare dentro la cartella `/etc` tutti i file che contengono la stringa `fstab`

Riferimenti

- http://www.x.org/ - Official site of the Xorg foundation
- http://www.linfo.org/x.html – introduction to X
- http://tools.ietf.org/html/rfc1198 – RFC