

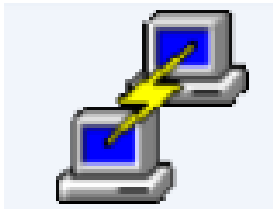
# Accesso ai PC del laboratorio

- Cliccare in basso a sinistra su Altro Utente
- Viene proposto il dominio AULE
- Utilizzare, a seconda del laboratorio:
  - Nome utente: lezione\_MLAB1
  - Nome utente: lezione\_MLAB2
- Funziona in entrambi i laboratori anche il seguente, ma presenta un'interfaccia con funzionalità ridotte
  - Nome utente: lezione
- La password dovrebbe essere già nota o verrà indicata al momento

# Il client putty

- Client ssh liberamente utilizzabile e scaricabile  
([www.chiark.greenend.org.uk/~sgtatham/putty/download.html](http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html))
- Sui PC del laboratorio si trova in:  
`C:\Programmi\MATLAB\R2019a\toolbox\idelink\foundation\hostapps\putty.exe`

- Icona:



# Utilizzo di putty

- Tra le molte opzioni di configurazione di putty sono utili:
  - Window - Translation: per selezionare il character set (scegliere UTF-8)
  - Window - Behaviour: selezionare System menu appears on ALT, in modo da poter far apparire la finestra di configurazione con ALT durante la sessione
  - Window - Appearance per cambiare il font
- L'uso base e' molto semplice: richiede solo di specificare:
  - nome DNS o indirizzo IP, protocollo da utilizzare (SSH), numero di porta
  - ad ogni gruppo è assegnata una coppia nome DNS-porta, corrispondente ad una diversa macchina virtuale
- Col bottone Open si accede alla fase di login (controllare la correttezza di tutti i settaggi: nome, numero porta, protocollo SSH)

# Fase di login

- Quando si effettua il login per la prima volta da una certa postazione sulla VM viene presentata una domanda relativa allo scambio di chiavi tra VM e postazione: rispondere Sì
- Vengono richiesti nome utente e password
- Se il login va a buon fine, viene presentato il prompt dei comandi (tipicamente il carattere \$) eventualmente preceduto da alcune righe di messaggio iniziale
- La directory corrente nella quale si trova l'utente dopo la fase iniziale di login è predefinita per ciascun utente e viene denominata home directory dell'utente
- L'utente può iniziare ad interagire col sistema tramite la linea di comando (in inglese detta CLI : Command Line Interface)

# Regole di utilizzo

- Le virtual machine (VM) sono dedicate al solo svolgimento di esercitazioni dei corsi dell'area Sistemi Operativi, sia durante le ore di laboratorio sia in modalità self-service con accesso remoto
- Le risorse di memorizzazione e di computazione di ciascuna VM sono limitate ma sufficienti alla realizzazione di semplici compiti: gli studenti devono limitare il carico delle VM allo stretto necessario
- Non sono autorizzati: download di dati/software per scopi non didattici (lo spazio è comunque molto limitato), utilizzo delle VM per navigazione web (comunque possibile in modalità solo testo) per scopi non didattici, invio mail per scopi non didattici, etc.
- Le attività svolte dalle VM sono soggette ai normali controlli da parte degli amministratori di sistema: ogni abuso verrà sanzionato con provvedimenti disciplinari
- In caso di problemi tecnici inviare una segnalazione e-mail a: [pietro.baroni@unibs.it](mailto:pietro.baroni@unibs.it)

# Note minime su uso di puTTY: sessioni di lavoro multiple

- E' possibile aprire più connessioni alla stessa macchina in finestre separate
- Questo può essere utile per vari scopi, ad esempio comodo per leggere documentazione in una finestra e agire sulla base di tale documentazione nell'altra finestra

# Note minime su uso di puTTY:

## Copy e Paste

- **Copy:** all'interno di una finestra puTTY se si seleziona col mouse una porzione di testo questa è anche automaticamente copiata  
N.B. in una finestra puTTY, Ctrl-C (così come altre combinazioni di caratteri analoghe) viene interpretato come una digitazione per la linea di comando, con i propri effetti, e non dà luogo a Copy
- **Paste:** all'interno di una finestra puTTY si può incollare del testo precedentemente copiato premendo il tasto destro del mouse (oppure Shift-Ins)
- Sono possibili Copy e Paste tra finestre puTTY (che si comportano come sopra) e altre finestre Windows (che si comportano come solito)

# Il comando man

- Fornisce informazioni di varia natura su vari aspetti del sistema
- Per iniziare: `man man`
- Distribuzioni diverse talvolta presentano man pages diverse sullo stesso argomento
- Vi sono anche siti di man pages:  
p.e. <http://man7.org/linux/man-pages/index.html>
- **Confrontare contenuto di `man man` con pagina relativa a comando `man` sul sito [man7.org](http://man7.org)**



# Sintassi dei comandi

- Tipica sintassi di un comando:  
`nomecomando opzioni argomenti_comando`
- Il nome comando può riferirsi ad un file eseguibile a se' stante o a un comando incorporato (builtin) nella shell (la documentazione dei builtin si trova nella manpage della shell stessa: `man bash`)
- La sezione SYNOPSIS del man sintetizza tutti i possibili usi di un comando
- Le opzioni sono (tipicamente) precedute da `-` (forma breve) o `--` (forma lunga o verbose), alcune opzioni richiedono propri argomenti
- Gli argomenti rappresentano l'oggetto del comando
- La specifica della sintassi del comando evidenzia le parti opzionali comprese tra `[ ]` e quelle ripetibili piu' volte seguite da `...`
- Premere `q` per uscire da man

# Decifrare la SYNOPSIS

```
man [-acdfFhkKtwW] [--path] [-m system] [-p string]
[-C config_file] [-M pathlist] [-P pager]
[-S section_list] [section] name ...
```

- Una parte dello standard POSIX describe la sintassi da usare nella SYNOPSIS

[http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1\\_chap12.html](http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1_chap12.html)

- Tuttavia non è detto che ciò che si trova in pratica rispetti sempre esattamente lo standard
- I lucidi seguenti danno alcune indicazioni utili a orientarsi con la SYNOPSIS

# Decifrare la SYNOPSIS

```
man [-acdfFhkKtwW] [--path] [-m system] [-p string]  
[-C config_file] [-M pathlist] [-P pager]  
[-S section_list] [section] name ...
```

- Le parti tra parentesi quadre sono facoltative
- I . . . indicano che quanto precede può essere ripetuto
- Quindi l'uso più semplice del comando toglie le parti facoltative e le possibili ripetizioni è:

# Decifrare la SYNOPSIS: uso minimale

**man** name

In generale nella SYNOPSIS:

- il testo in grassetto è da utilizzare così com'è (in questo caso **man**)
- il testo non in grassetto (e sottolineato) indica parametri formali da sostituire di volta in volta con valori appropriati (in questo caso name sarà sostituito di volta in volta con il nome del comando di cui si vogliono leggere le pagine di manuale)

Esempi di uso minimale:

```
man man
```

```
man ls
```

```
man rm
```

# Decifrare la SYNOPSIS: ripetizioni

**man** name ...

Dovunque nella SYNOPSIS ci siano ... l'elemento che precede può essere ripetuto più volte

Esempio di uso con ripetizione:

```
man man ls rm
```

mostra nell'ordine le pagine di manuale di `man`, `ls`, e `rm`

# Decifrare la SYNOPSIS: opzioni

```
man [-acdfFhkKtwW] [--path] [-m system] [-p string]  
[-C config_file] [-M pathlist] [-P pager]  
[-S section_list] [section] name ...
```

- Un comando ha tipicamente molti comportamenti possibili che possono venire selezionati specificando una o più opzioni
- La specifica di un'opzione è di solito preceduta da almeno un -
- Tradizionalmente ci sono:
  - opzioni in forma breve specificate con un singolo carattere (precedute da un singolo trattino)
  - opzioni in forma estesa specificate con intere parole (precedute da un doppio trattino)
- In molti casi (ma non nella synopsis di `man` sopra riportata) ci sono opzioni specificabili in entrambi i modi
- Per esempio molti comandi hanno un'opzione di aiuto invocabile indifferentemente con `-h` e con `--help`

# Decifrare la SYNOPSIS: opzioni con propri argomenti

```
man [-acdfFhkKtwW] [--path] [-m system] [-p string]  
[-C config_file] [-M pathlist] [-P pager]  
[-S section_list] [section] name ...
```

- Alcune opzioni richiedono propri argomenti altre no
- Ad esempio l'opzione **-M** richiede un argomento `pathlist` (che si usa se si vuole specificare una lista di percorsi nel file system dove cercare le pagine di manuale)
- Le opzioni **acdfFhkKtwW** e l'opzione **--path** non hanno propri argomenti

# Decifrare la SYNOPSIS: opzioni combinabili

```
man [-acdfFhkKtwW] [--path] [-m system] [-p string]  
[-C config_file] [-M pathlist] [-P pager]  
[-S section_list] [section] name ...
```

- Normalmente più opzioni sono combinabili tra loro
- In particolare per le opzioni che non hanno argomenti propri la combinazione viene specificata mettendo in sequenza più caratteri dopo il trattino
- Ad esempio:

man -aDw man equivale a man -a -D -w man

N.B. Non è detto che tutte le combinazioni specificabili siano praticamente utili o sensate



# Decifrare la SYNOPSIS: opzioni non combinabili

- In alcune pagine man per indicare che due opzioni sono mutuamente esclusive tra loro si usa la barra verticale |
- Ad esempio

```
grep [options] [-e PATTERN | -f FILE] [FILE...]
```

indica che le opzioni **-e** e **-f** con i rispettivi argomenti vanno usate in alternativa

# Decifrare la SYNOPSIS: argomenti facoltativi del comando

```
man [-acdfFhkKtwW] [--path] [-m system] [-p string]  
[-C config_file] [-M pathlist] [-P pager]  
[-S section_list] [section] name ...
```

Oltre alle opzioni un comando può avere suoi argomenti facoltativi

Nel caso di `man`, [section] indica un argomento facoltativo che può precedere name

Ad esempio

```
man 7 man
```

cerca le pagine di manuale relative a `man` nella sezione 7 anzichè nella sezione 1

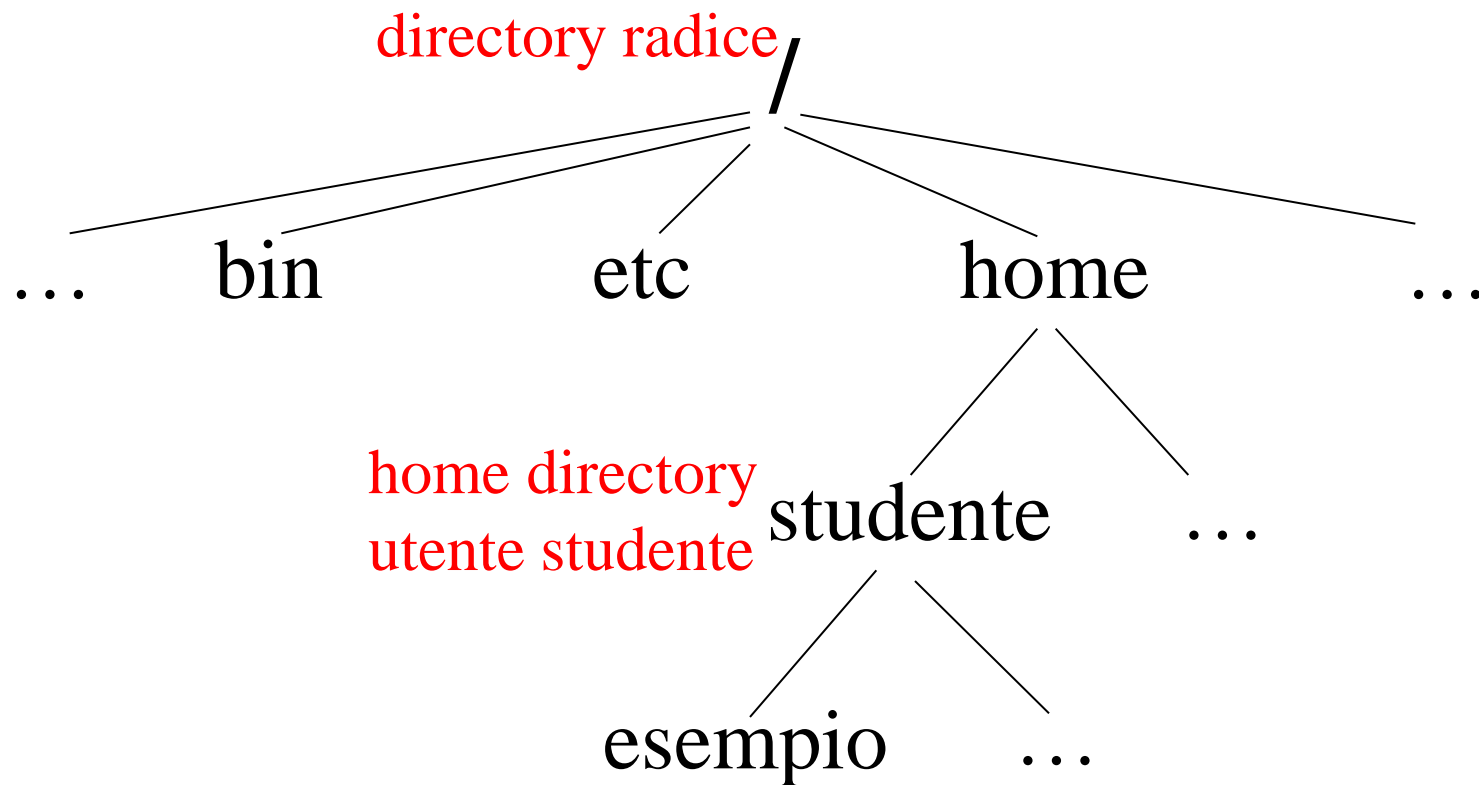
# RTFM (Read The Fu...nny Manual)

- L'abitudine alla lettura di manuali e documentazione è parte indispensabile nell'utilizzo professionale di qualunque sistema operativo
- Molti comandi forniscono inoltre un'opzione `-h` o `--help` per visualizzare informazioni sul loro utilizzo
- Non sempre le pagine man sono pensate per il principiante
- In tal caso può essere utile ricorrere a: `info nomecomando` che talvolta replica semplicemente le pagine man ma in altri casi fornisce una trattazione più approfondita e didattica rispetto alle pagine man troppo sintetiche
- In casi disperati: cercare su Internet

# L'ambiente info

- info fornisce un ambiente ipertestuale navigabile con interfaccia a caratteri: per imparare ad usarlo `info info` (NON ADESSO)

# Struttura del file system



- La home directory è la posizione dell'albero in cui ci si trova (directory corrente) quando ci si collega
- Muovendosi su e giù per l'albero si cambia la directory corrente

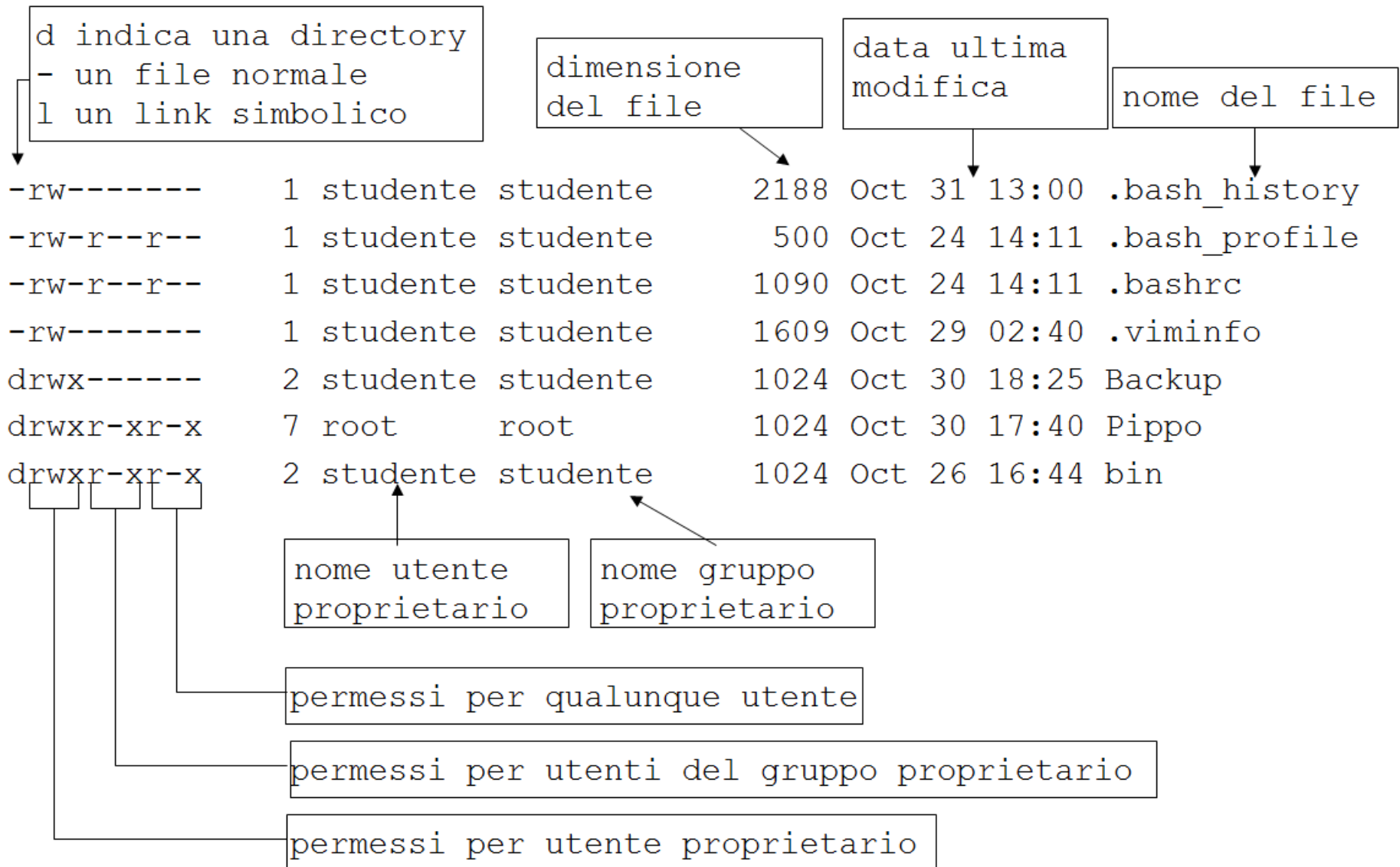
# Comandi base di navigazione sul filesystem

- `pwd` : visualizza la directory corrente
- `cd nomedir` : `nomedir` diventa la nuova directory corrente, se non si specifica `nomedir` si passa alla home directory dell'utente corrente. **Eseguire man cd.**
- valori speciali di `nomedir`:
  - `.` Directory corrente
  - `..` Directory superiore
  - `~` home directory dell'utente
- `ls` : senza argomenti, visualizza il contenuto della directory corrente, con argomenti: RTFM **Eseguire man ls o info ls**
- opzioni principali di `ls`:
  - `-a` visualizza anche i “file nascosti” (quelli il cui nome inizia con `.`)
  - `-l` visualizza informazioni estese sul contenuto della directory
  - `-h` visualizza dimensioni dei file in formato più leggibile

# Prova di utilizzo ls

- **Identificare il file di dimensione maggiore tra quelli contenuti nella directory esempio**
- **Identificare il file modificato piu' recentemente tra quelli contenuti nella directory esempio**

# Formato output di ls -al





# Nomi (pathname) assoluti o relativi

- All'interno del file system ogni file o cartella è identificato in modo univoco dal suo nome (pathname) assoluto che rappresenta il percorso per raggiungere quel file o cartella a partire dalla radice

Esempi di nomi assoluti:

```
/home/studente/esempio
```

```
/home/studente/esempio/HelloWorld.Java
```

```
/bin/date
```

# Nomi (pathname) assoluti o relativi

- Dato che i nomi assoluti possono essere molto lunghi e scomodi per l'utente è possibile specificare nomi relativi (che non iniziano con /) e sono riferiti alla directory corrente.
- Per esempio se la directory corrente è `/home/studente`  
esempio verrà interpretato dal sistema come `/home/studente/esempio`  
e similmente `esempio/HelloWorld.Java` verrà interpretato come `/home/studente/esempio/HelloWorld.Java`
- Mentre se la directory corrente è `/bin`  
esempio verrà interpretato dal sistema come `/bin/esempio`  
e così via

# Nomi (pathname) assoluti o relativi

- Nel seguito dovunque si farà riferimento ad un nome di file si potrà usare indifferentemente un nome assoluto o uno relativo
- Le prime volte può capitare di fare un po' di confusione tra i due tipi di nomi se non si è abituati ad utilizzare la linea di comando
- Se succede qualcosa di inaspettato (in particolare il sistema ci dice che non esiste un file o cartella che sappiamo esistere) spesso va ricontrollato il nome che abbiamo digitato o la directory corrente

# Aiuti alla digitazione

- La digitazione sulla linea di comando e nomi è facilitata da due caratteristiche dell'interprete dei comandi
- **Autocompletamento:** se si inizia a scrivere un nome e poi si preme il tasto `Tab` il nome viene completato automaticamente (se il completamento è univoco) o viene esteso per quanto possibile se ci sono più possibili completamenti (e ripremendo `Tab` vengono mostrati tutti i completamenti possibili)
- **Navigazione comandi precedenti:** premendo la freccia in alto si percorre all'indietro la storia dei comandi precedentemente impartiti (e con la freccia in basso la si ripercorre in avanti fino all'ultimo)

# Visualizzazione di file

- `cat nomefile`: presenta a video il contenuto di `nomefile` ed esce
- `more nomefile`: presenta a video il contenuto di `nomefile` una pagina alla volta (premere spazio per avanzare di una pagina)
- `less nomefile`: presenta a video il contenuto di `nomefile` una pagina alla volta e permette scorrimento avanti e indietro (oltre allo spazio come sopra, si possono usare le freccette e PgUp/PgDown)
- In numerosi ambiti (`man`, `more`, `less`, ...) per effettuare la ricerca di una `stringa` nel testo si usa il comando `/:`  
`/stringa`

# Visualizzazione di file

- **Visualizzare il file `istruzioni.txt` presente nella home directory e seguire le sue indicazioni**

# Identità e autorizzazioni

- Per cambiare identità durante una sessione di lavoro, si usa il comando `su nomeutente`
- In assenza di `nomeutente`, `su` equivale a `su root`
- **Diventare `root` (il superutente)**
- In caso di crisi d'identità, il comando `whoami` presenta l'utente corrente
- **Verificare la propria identità**
- Per tornare nell'identità precedente: `exit` (esce dal sistema se si è nella “prima” identità)
- **Abbandonare i superpoteri e verificare il ritorno all'identità “normale”**

# Modifica proprietà e autorizzazioni di file

- Per modificare le autorizzazioni di un file:  
`chmod mode nomefile`
- La sintassi per specificare il mode non è immediata: RTFM  
(esempi lucido successivo)
- Per cambiare utente proprietario e gruppo proprietario: comandi  
`chown` e `chgrp`



# Esempi chmod

- Due sintassi: con numeri ottali oppure ugoa+-rwx
- Con la prima sintassi è possibile specificare i 3 gruppi di bit rwx come tre numeri ottali:  
chmod 744 nomefile dà luogo a rwxr--r--  
chmod 555 nomefile dà luogo a r-xr-xr-x  
chmod 600 nomefile dà luogo a rw-----
- Con la seconda sintassi si specificano variazioni rispetto allo stato precedente tramite una o più espressioni di aggiunta/cancellazione di permessi. Ciascuna espressione ha la seguente forma:  
ambito+/-permessivariati:  
chmod go-r        toglie permesso r a gruppo e utente generico  
chmod ug+wx      aggiunge permessi wx a proprietario e gruppo  
chmod a+r        aggiunge permesso r a tutti i livelli

# Operazioni sul file system

- `mv` : rinomina o sposta file o directory
- `cp` : copia file (o directory, anche in modo ricorsivo con opzione `-R`)
- `mkdir` : crea una nuova directory
- `rm` : rimuove un file o directory
  - di norma viene chiesta conferma (opzione `-f` per non dover confermare)
  - di norma si può rimuovere una directory solo se vuota (usando comando `rmdir`)
  - si può usare il comando `rm -r nome_dir` per rimuovere ricorsivamente una directory e tutto il suo contenuto)

# Mettiamo il tesoro al sicuro

- Creare una nuova directory con un nome a propria scelta
- Spostare il file contenente la password di `root` in questa nuova directory
- Cambiare il nome al file contenente la password di `root`
- Cambiare i permessi al file contenente la password di `root` in modo che sia leggibile e modificabile solo dal proprietario
- Cambiare proprietario alla nuova directory che contiene il file in modo che sia di proprietà di `root`
- Cambiare permessi alla directory in modo che sia leggibile, modificabile ed “eseguibile” solo dal proprietario
- Fare una copia della directory e del suo contenuto all’interno di un’altra directory a propria scelta
- Cambiare proprietario da `root` a `studente` della directory copia e del suo contenuto

# Operazioni su file

- `file nomefile`: da' indicazioni sul tipo di contenuto di un file
- `head nomefile`: visualizza le prime righe di un file (per default 10, il numero è specificabile con l'opzione `-n`)
- `tail nomefile`: visualizza le ultime righe di un file (per default 10, il numero è specificabile con l'opzione `-n`)
- `grep stringa nomefile`: visualizza tutte le righe di `nomefile` che contengono una certa stringa (più in generale un certo pattern)
- `touch nomefile`: aggiorna la data di ultima modifica di `nomefile` (se `nomefile` non esiste lo crea vuoto)

# Uso di comandi su file

- Verificare il tipo di tutti i file presenti nella directory `~/esempio`
- Verificare quali file nella directory `~/esempio` contengono la stringa `Hello`
- Visualizzare le prime righe dei file Postscript e tex presenti nella directory `~/unaDirectory/latex`
- Visualizzare le ultime righe del file log presente nella directory `~/unaDirectory/latex`
- Eseguire il comando `touch` su un file esistente e su uno inesistente e verificarne gli effetti

# Editing: vi

`vi` è un text editor con interfaccia a caratteri di antica tradizione  
Su molti sistemi in realtà dando il comando `vi` se ne invoca una  
versione “modernizzata” denominata `vim` (`vi improved`)

**Dare il comando `vimtutor` per entrare in un breve tutorial**

In generale il comando per iniziare è: `vi nomefile`

`vi` ha due modalità di funzionamento

- la modalità “Command” (o “Normal”) nella quale ciò che viene digitato sulla tastiera è interpretato come comando (NON come testo da inserire)

- la modalità “Editing” (o “Insert”) nella quale ciò che viene digitato sulla tastiera è interpretato come testo da inserire (salvo caratteri speciali come frecce e backspace in `vim`)

La modalità iniziale è “Command”: per passare a “Editing” si usa di norma il comando `i` (altre possibilità sono nel tutorial)

Per tornare a “Command”: digitare ESC

# Editing: vi(m)

E' facile perdersi tra le due modalità e perdere contestualmente la pazienza

Le prime volte se ci si perde è meglio ripartir da capo: ESC per tornare in normal mode e :q! per uscire senza salvare

Una volta capito il meccanismo e presa confidenza, vi può diventare uno strumento di estrema produttività

Dopo i primi passi con vimtutor, può essere utile riferirsi a

<http://www.vim.org/>  
sito ufficiale del programma

[http://vimdoc.sourceforge.net/html/doc/usr\\_toc.html](http://vimdoc.sourceforge.net/html/doc/usr_toc.html)  
per documentazione e manualistica

<http://help.ubuntu.com/community/VimHowto>  
per un HOWTO

# Prova di utilizzo vi

Saper creare/modificare un “Hello world” con vi è già un buon segno

**Quindi nel file `hello.c` contenuto nella directory `esempio` ora dovrete riuscire a:**

- modificare il testo p.e. far effettuare due `fprintf` distinte con i due saluti (usando copia e incolla), scrivere un altro nome anziché `Brescia`**
- sostituire `Hello` con `Ciao` (usando il comando di ricerca e sostituzione)**
- salvare il testo modificato in un file con un altro nome e uscire**

**Dovreste anche riuscire a creare da zero un nuovo file di nome “`gioia_vi_hai`” che contiene il seguente testo: “Sono contento di aver imparato vi”**



# Editing: mcedit

Ma se non siete contenti ...

“You may have decided you hate *vi* . So be it!”

*vi* è l'editor standard la cui presenza è garantita su qualunque UNIX, ma se non lo si sopporta si può tentare con `mcedit`

Si tratta di un editor “stile MS-DOS”, addirittura dotato di menu a tendina e di un help interno

I comandi impartibili sono visualizzati in basso e sono in corrispondenza con i tasti funzione (p.e. all'avvio, F1 apre l'help, F9 apre i menu a tendina, F10 esce, etc.)

“The editor is very easy to use and requires no tutoring”

**Provare a svolgere anche con `mcedit` i compiti specificati nel lucido precedente**

# "Compiti" a casa

- Prendere confidenza con la linea di comando ed i comandi sui file
- Creare una nuova directory di nome nuova dentro la directory `/home/studente/unaDirectory`
- Copiare dentro nuova tutti i file contenuti in `/home/studente/images/medie/`
- Rinominare `Tessuto.bmp` in `MioTessuto.bmp` e poi cambiarne i permessi in modo che sia leggibile e scrivibile solo dal proprietario
- Creare una directory di nome nuovissima dentro la directory `/home/studente/waltDisney`
- Copiare ricorsivamente la directory `/home/studente/unaDirectory` dentro `/home/studente/waltDisney/nuovissima`

# "Compiti" a casa

- Provare a utilizzare `man` (o altre risorse) per capire il comportamento di comandi articolati
- Capire come utilizzare `find` per farsi restituire la lista dei file modificati negli ultimi `X` giorni all'interno di una certa `DIRECTORY` e sue sottodirectory
- Capire come utilizzare `grep` per cercare una certa `STRINGA` in modalità case insensitive in tutti i file presenti all'interno di una certa `DIRECTORY` e sue sottodirectory mostrando per ciascuna occorrenza trovata il numero della linea del file dove essa appare

# "Compiti" a casa

- Se si è curiosi su un ambiente di documentazione alternativo a man e su come può essere fatto un ambiente ipertestuale senza interfaccia grafica provare `info`
- Usare `info info` per prendere confidenza con le modalità essenziali di navigazione dell'ambiente `info`. Verificare la padronanza acquisita provando ad usare `info man`

# "Compiti" a casa

- Rinforzare la pratica di editing di file di testo
- Creare una nuova directory di nome `testi` dentro la directory `/home/studente/esempio`
- Creare dentro `testi` un file di testo di nome `studenti.txt` contenente alcune righe strutturate tipo:  
`NOME , COGNOME , MATRICOLA`
- Effettuare ricerche e sostituzioni a piacimento all'interno del file (ad esempio cercare un certo cognome e sostituire tutte le sue occorrenze con un altro cognome) e salvare ogni volta la versione modificata con un altro nome
- Alla fine nella directory ci saranno alcuni file di testo con nomi e contenuti diversi (ma le cui righe sono strutturalmente simili)

# Lookahead strategy

- Le esercitazioni di laboratorio sono guidate e autocontenute e non richiedono di per sé lavoro aggiuntivo o preparatorio
- Dare un'occhiata ai lucidi dell'esercitazione successiva non può far male se si ha tempo