

Come avviare la VM

1. Far partire l'explorer di Windows e recarsi sul disco `C:\`
2. Recarsi in `C:\VM\LUbuntu`
 - Attivare le estensioni nomi file tramite View > Show > File Name Extensions
3. Il file `LUbuntu.vmx` contiene le informazioni sull'avvio della VM (per quanto riguarda VMWare Workstation/Player). Eseguire il file.
 - Se necessario selezionare VMWare Workstation
4. Su VMWare Workstation, premere "Power on this virtual machine"
 - Premere OK se necessario
 - Nel caso appaia che il profilo sia stato spostato o copiato, selezionare "Copied".
5. Eseguire il login dell'account studente con password "studente"

Come spegnere la macchina: **NON cliccare sulla X di VMWare Workstation**; utilizzare il menù di `lubuntu > leave > shutdown`.

Perchè? Siccome Linux usa fortemente il disco, il rischio di perdere file di sistema spegnendo il PC improvvisamente è elevato. Pertanto bisogna spegnere la macchina tramite `shutdown`.

Il terminale utilizzato è `qTerminal`, avviabile da start o con `CTRL+ALT+T`; tale terminale è basato su `bash`: in caso si avesse un'altro tipo di `bash`, eseguire il comando `bash`.

Per chiudere una `bash`, usare `CTRL-D` o il comando `exit`.

`/etc/whoami`

Il file `/etc/whoami` contiene informazioni su tutti gli utenti del sistema.

Ogni riga rappresenta uno specifico utente, specificato nel seguente modo:

```
cat /etc/whoami
```

```
<username>:x:<uid>:<groupid>:<description>:<home_dir>:<op_location>
```

`/etc/shadow`

Il file `/etc/shadow` non può essere letto dagli utenti normali (`others` hanno permessi 0); va eseguito come `root` o utilizzando `sudo`.

Ogni riga di questo file contiene le password crittografate dei vari utenti; quando un utente effettua il login tramite password, essa viene convertita tramite lo stesso algoritmo di hash e poi comparata con la hash presente in `etc/shadow`; se sono uguali il login va a buon fine

/etc/group

Il file `/etc/group` contiene informazioni relative ai gruppi di sistema e agli utenti che vi appartengono.

Ogni riga rappresenta uno specifico gruppo, specificato nel seguente modo:

```
cat /etc/group

<groupname>:x:<groupid>:(users_in_group)
```

/usr/bin/passwd

Il file `/usr/bin/passwd` permette di cambiare le password degli utenti. Questo file è particolare: al posto di `x`, nei permessi utente relativi all'esecuzione, c'è una `s`; ciò **significa che l'utente che esegue tale file genera un processo che ha i permessi del proprietario effettivo del file**. Per esempio, se `studente` avvia `/usr/bin/passwd/`, durante l'esecuzione di tale file `studente` avrà i permessi del proprietario, ossia `root`.

Il proprietario del processo e il proprietario del file sono quindi due cose profondamente diverse.

Esecuzione di comandi da CLI

I comandi built-in sono eseguibili da ogni posizione della shell. Nel caso invece di esecuzione di files, bisogna procedere in maniera diversa.

Ci sono 3 modi per eseguire comandi da linea di comando:

- Tramite **percorso assoluto**: `/home/vulpi/Desktop/ciao`
- Tramite **percorso relativo**: se la directory è `/home/pippo`, allora `../vulpi/Desktop/ciao`
- Solo **tramite il nome del file**, a patto che sia specificato nella `PATH`.

Subshells & Variabili

Una subshell è una shell generata da un'altra shell; la prima si chiama shell figlia, la seconda shell padre. Essa viene creata in più casi:

- Processi in background
- Comandi raggruppati / concatenati
- **Esecuzione di script**

Nell'ultimo caso viene creata una shell figlia basata dal **primo commento presente** nel file di script. Se per esempio è presente `#!/bin/bash` viene eseguita una `bash` figlia.

Ciascuna bash ha directory (inizialmente ereditata dal padre) e variabili personali. Ogni subshell ottiene una **copia** delle variabili d'ambiente del padre, ma **non eredita le variabili locali**.

- Variabili d'ambiente: sono variabili sempre utili e che vengono trasmesse **COME COPIA** da padre a figlio: alcune sono `$PATH` , `$HOME` , `$SHELL` , ...
- Variabili locali: sono le variabili che sono state create dall'utente in uno specifico momento. **Non vengono trasmesse da padre a figlio.**

Il fatto che le variabili d'ambiente vengano passate come copia è fondamentale: infatti le shell figlie non possono modificare le variabili d'ambiente del padre.

E' possibile trasformare una variabile locale a una variabile d'ambiente tramite il comando `export` .

E' possibile visualizzare tutte le variabili d'ambiente tramite il comando `env` .

E' possibile eliminare una variabile tramite `unset` .

Il comando `set` mostra tutte le variabili e le funzioni della shell.

Utilizzare `source` seguito dal nome di uno script farà in modo di eseguire tale script all'interno della stessa shell da cui il comando è stato avviato, **senza creare una shell figlia**; questo può essere utile per modificare variabili locali/d'ambiente direttamente nella shell padre.

Bisogna fare attenzione ad eseguire script tramite `source` perché i comandi verranno eseguiti direttamente sulla shell padre.

`/home/.bashrc`

Questo è il file di configurazione della bash. Al suo interno sono presenti delle dichiarazioni di variabili e vari comandi che **vengono eseguiti non appena viene creato un terminale**.