



Laboratorio di Sistemi e Reti

Comandi per l'utilizzo base della SHELL Linux

Classe 3^A - Indirizzo Informatica

Emanuele Bertolero

Istituto Tecnico Industriale "Don Luigi Orione"



```

/bin/bash
( Today is National Existential Ennui )
( Awareness Day. )
-----
o
o

  o.o |
  :./ |
(( )) |
(( )) |
(( ))=|
(( ))

metalx1000@taka ~$ man pv
metalx1000@taka ~$ echo "This is some text"
This is some text
metalx1000@taka ~$ echo "This is some text"
This is some text
metalx1000@taka ~$ echo "This is some text"|pv -qL 10
This is some text
metalx1000@taka ~$ echo "This is some text"|pv -qL 10
This is some text
metalx1000@taka ~$ echo "This is some text"|pv -qL 20
This is some text
metalx1000@taka ~$ echo "This is some text"█iv -qL 100
This is some text
metalx1000@taka ~$ █
```



Comandi

I comandi per la gestione del **filesystem**:

- cd**: Cambia directory
- ls**: Elenco dei file e delle directory
- mkdir**: Crea una nuova directory
- rmdir**: Rimuove una directory vuota
- touch**: Crea un nuovo file vuoto
- rm**: Rimuove un file o una directory
- mv**: Sposta o rinomina file e directory
- cp**: Copia file e director

I comandi **base**:

- sudo**: Eseguire comandi con privilegi di superutente
- apt**: Trovare ed installare pacchetti nuovi
- clear**: Ripulire lo schermo della console
- chmod**: Modificare i permessi di accesso



CD

Il comando "**cd**" (Change Directory) consente di **spostarsi** tra le directory del sistema, utilizzato quindi per **cambiare** la directory corrente in una specifica directory di destinazione.

Sintassi

```
cd [directory]
```

Dove **directory** può essere:

- **..** (punto punto): la directory corrente al livello superiore (directory genitore);
- **~** (tilde): la directory home dell'utente corrente;
- **/** (barra): la directory radice del sistema;
- **nome_directory**: per spostarsi all'interno della directory corrente.



LS

Il comando "**ls**" viene utilizzato per **visualizzare** il contenuto di una directory specifica. Fornisce un **elenco** dettagliato o semplice dei file e delle directory.

Sintassi

```
ls [opzioni] [directory]
```

Dove **opzioni** può essere:

- **-l**: Elenco dettagliato con: i permessi, il proprietario, la dimensione, ecc
- **-a**: Mostra anche i file nascosti (che iniziano con un punto);
- **-h**: Mostra le dimensioni dei file in un formato leggibile;
- **-t**: Ordina i file e le directory per data e ora di modifica.

Dove **directory** può essere:

- Vuota (si intende la corrente)
- Percorso specifico



MKDIR

Il comando "**mkdir**" (Make Directory) permette di creare una o più directory all'interno della directory corrente o in una specifica directory di destinazione.

Sintassi

```
mkdir [opzioni] directory
```

Dove **opzioni** può essere:

- **-p**: Crea anche le directory genitore se non esistono. Ad esempio per creare directory in cascata una dentro l'altra;
- **-m**: Imposta i permessi di accesso delle nuove directory secondo le opzioni specificate;
- **-v**: Mostra un output verboso, visualizzando ogni directory creata.

Dove **directory** è il nome che vogliamo dare alla directory



RMDIR

Il comando "**mkdir**" (Remove Directory) consente di eliminare directory vuote dalla directory corrente o da una specifica directory di destinazione.

Sintassi

```
rmdir [opzioni] directory
```

Dove **opzioni** può essere:

- **-p**: Rimuove anche le directory genitore se diventano vuote dopo la rimozione della directory specificata.
- **-v**: Mostra un output verboso, visualizzando il messaggio di conferma per ogni directory rimossa.

Dove **directory** è il nome della directory che vogliamo rimuovere che ricordiamo deve essere vuota, infatti è comodo per evitare la rimozione accidentale di directory contenenti file importanti.



TOUCH

Il comando "**touch**" consente di creare nuovi file vuoti o di aggiornare la data di accesso e di modifica di file esistenti senza modificare il contenuto dei file.

Sintassi

```
touch [opzioni] nome_file
```

Dove **opzioni** può essere:

- **-a**: Aggiorna solo la data di accesso del file;
- **-m**: Aggiorna solo la data di modifica del file;
- **-c**: Non crea nuovi file se non esistono già;
- **"-t timestamp"**: Imposta una data e un'ora specifica per il file nel formato "CCYYMMDDhhmm.ss".

Dove **nome_file** è il nome del file che vogliamo creare



RM

Il comando "**rm**" (Remove) permette di rimuovere file e directory in modo permanente dal sistema.

Sintassi

```
rm [opzioni] file/directory
```

Dove **opzioni** può essere:

- **-r** o **-R**: Rimuove in modo ricorsivo i file e le directory;
- **-f**: Forza l'eliminazione dei file senza richiedere conferme;
- **-i**: Chiede conferma prima di eliminare ogni file o directory;
- **-v**: Visualizza il nome di ogni file o directory eliminati.

Dove **file/directory** è il nome del file o della directory che elimineremo in modo permanente senza possibilità di recupero.



MV

Il comando "**mv**" (Move) consente di **spostare** file e directory da una posizione all'altra all'interno del sistema di file o di rinominarli.

Sintassi `mv [opzioni] sorgente destinazione`

Dove **opzioni** può essere:

- **-i**: Chiede conferma prima di sovrascrivere un file esistente;
- **-v**: Mostra un output verboso, visualizzando il nome di ogni file o directory spostato.

Dove **sorgente** è il nome del file o della directory che vogliamo spostare

Dove **destinazione** è la nuova posizione che daremo al file o directory



CP

Il comando "**cp**" (Copy) consente di creare **copie** di file e directory, consentendo di **duplicare** e replicare il contenuto all'interno del sistema di file.

Sintassi

```
cp [opzioni] sorgente destinazione
```

Dove **opzioni** può essere:

- **-r** o **-R**: Copia in modo ricorsivo le directory e il loro contenuto;
- **-i**: Chiede conferma prima di sovrascrivere un file esistente;
- **-v**: Mostra un output verboso, visualizzando il nome di ogni file o directory copiato.

Dove **sorgente** è il nome del file o della directory che vogliamo copiare

Dove **destinazione** è la posizione dove vogliamo creare suddetta copia



SUDO

Il comando "**sudo**" (SuperUser Do) consente agli utenti autorizzati di eseguire comandi che richiedono privilegi elevati, come l'amministrazione di sistema, la gestione dei pacchetti e la modifica di file di configurazione.

Sintassi

```
sudo comando
```

Quando si esegue un comando con "sudo":

1. il sistema **verifica** se l'utente ha i permessi **necessari** per eseguirlo
2. se l'utente è autorizzato viene richiesto di **immettere** la propria **password** (la password non viene mostrato per motivi di sicurezza)
3. il comando viene **eseguito** con i privilegi di amministratore.

Paragonabile a



Esegui come amministratore

su Windows.



APT

Il comando "**apt**" (Advanced Package Tool) semplifica la gestione dei pacchetti software nel sistema, consentendo agli utenti di installare facilmente nuovi software, aggiornare le versioni esistenti e rimuovere software non più necessario.

Sintassi

```
apt [opzioni] comando [pacchetto]
```

I possibili **comandi** sono:

- **update**: aggiorna la lista dei pacchetti disponibili contattando i repository configurati e scaricando le informazioni più recenti
- **install**: aggiunge nuove funzionalità, applicazioni o librerie al sistema operativo scaricando e installando i pacchetti specificati
- **remove**: rimuove e disinstalla applicazioni o librerie specificate dal sistema



CLEAN

Il comando “**clear**” è utile per eliminare il testo precedente visualizzato nel terminale, consentendo di iniziare con una schermata vuota.

Sintassi

```
clear
```



CHMOD

Il comando “**clear**” è utilizzato per modificare i permessi di accesso dei file e delle directory nel sistema Linux.

Sintassi

```
chmod opzioni permessi file/directory
```

Dove **opzioni** può essere:

- **-R**: Applica la modifica dei permessi in modo ricorsivo a una directory e ai suoi contenuti.
- **--help**: Mostra una breve descrizione del comando "chmod" e delle sue opzioni disponibili.

Dove **file/directory** è il file/directory ai quali vogliamo cambiare permessi

Dove **permessi** è l'assegnazione dei permessi (*vedi slide successiva...*)



I permessi (1)

Il sistema dei permessi di accesso utilizza una notazione con tre cifre per rappresentare i permessi. Ci sono diverse modalità per scrivere i permessi su Linux. Ecco alcuni dei modi comuni:

- **Notazione simbolica:**

Questo metodo utilizza una combinazione di lettere e simboli per rappresentare i permessi. Le lettere più comuni sono: "**r**" (read) per il permesso di lettura, "**w**" (write) per il permesso di scrittura, "**x**" (execution) per il permesso di esecuzione

- **Notazione numerica:**

Questo è il metodo più comune e utilizza una sequenza di numeri da 0 a 7 per rappresentare i permessi. Ogni cifra corrisponde a un insieme di permessi per il proprietario, il gruppo e gli altri utenti.



I permessi: notazione simbolica

La notazione **simbolica** rappresenta i permessi di accesso utilizzando una combinazione di **lettere** e simboli.

Sintassi `[utenti] [operatore] [permessi]`

Dove i **permessi** sono:

- **r**: Permesso di lettura
- **w**: Permesso di scrittura
- **x**: Permesso di esecuzione

Dove gli **operatori** sono:

- **+**: Aggiunge i permessi
- **-**: Rimuove i permessi
- **=**: Imposta esattamente

Dove gli **utenti** sono:

- **u**: Proprietario (user)
- **o**: Altri utenti (others)
- **g**: Gruppo (group)
- **a**: Tutti gli utenti (all)

Esempio `chmod u+rwx,g+rw,o-rwx file.txt`



I permessi: notazione numerica

La notazione **numerica** offre una modalità compatta e diretta per indicare i permessi utilizzando delle **cifre** da 0 a 7.

Sintassi `[cifra_user] [cifra_group] [cifra_others]`

Dove le **cifre** possono essere:

- 0**: Nessun permesso (---)
- 1**: Esecuzione (--x)
- 2**: Scrittura (-w-)
- 3**: Scrittura ed esecuzione (-wx)
- 4**: Lettura (r--)
- 5**: Lettura ed esecuzione (r-x)
- 6**: Lettura e scrittura (rw-)
- 7**: Lettura, scrittura ed esecuzione (rwx)

d	r	w	x	r	-	x	r	-	-
	read	write	exec	read	write	exec	read	write	exec
File type	Owner permissions			Group permissions			User permissions		
(directory)	4	2	1	4	2	1	4	2	1
	7			5			4		

Esempio `chmod 764 file.txt`



Esercizi

1. Spostati nella directory del tuo utente e creane una nuova chiamata «Uno» che a sua volta ne contiene una chiamata «Due», così via fino a creare la directory «Quattro», fallo utilizzando un unico comando. All'interno di «Quattro» crea un nuovo file di testo «ricerca.txt». Siccome vuoi sembrare un bravo studente che si avvantaggia con i compiti falsifica la data di creazione della ricerca al 4 Maggio 2023.
2. Duplica ora la tua ricerca nella Scrivania (Desktop) specificatamente in una cartella chiamata «Compiti» che però non esiste e non puoi creare precedentemente, fai il tutto in unico comando.
3. Spostati nella Scrivania e poi nella cartella «Compiti», modifica ora i permessi della «ricerca.txt» in modo che solo il proprietario possa leggere, scrivere ed eseguire il file, che il gruppo possa leggerlo e modificarlo, mentre gli altri utenti possano solo leggere. Scrivi il comando sia utilizzando la notazione numerica sia utilizzando la notazione simbolica.
4. Elimina con un unico comando la cartella «Uno», ma anche la sua sotto cartella «Due», la «Tre», la «Quattro» e pure i files che contiene. Siccome non ti fidi della bash chiedile di mostrare un messaggio di conferma ad ogni eliminazione con successo.
5. Utilizzando il comando "sudo", crea una nuova directory chiamata «Sistema» nella directory radice ("/"). Assicurati che solo l'amministratore del sistema possa scrivere nella directory «Sistema».
6. Installa il pacchetto «vim» che è un bellissimo editor di testo. Provalo ad avviare e verifica funzioni correttamente. Successivamente, rimuovi completamente il pacchetto «vim» dal sistema.
7. Scrivi (ma non eseguire) un unico comando che possa distruggere tutto il sistema operativo cancellando qualsiasi files o directory presente nell'intero disco.



Soluzioni

1. `mkdir -p ~/Uno/Due/Tre/Quattro && cd ~/Uno/Due/Tre/Quattro`
`touch ricerca.txt`
`touch -d "2023-05-04" ricerca.txt`
2. `cp ~/Uno/Due/Tre/Quattro/ricerca.txt ~/Scrivania/Compiti/ricerca.txt`
3. **(Notazione numerica)** `chmod 754 ~/Scrivania/Compiti/ricerca.txt`
(Notazione simbolica) `chmod u=rwx,g=rw,o=r ~/Scrivania/Compiti/ricerca.txt`
4. `rm -rfv ~/Uno`
5. `sudo mkdir /Sistema`
`sudo chmod 700 /Sistema`
6. `sudo apt update`
`sudo apt install vim`
`vim`
`sudo apt remove vim`
7. `sudo rm -rf /`



POLO TECNICO PROFESSIONALE INDIRIZZO INDUSTRIALE
SCUOLE DON ORIONE FANO

- ISTITUTO TECNICO INDUSTRIALE
- ISTITUTO PROFESSIONALE INDUSTRIA E ARTIGIANATO
- CENTRO DI FORMAZIONE PROFESSIONALE

Emanuele Bertolero

emanuele.bertolero@donorionefano.edu.it