

Esercizio: Lavorare con File di Testo nella CLI

Parte 1 Editor di Testo Grafici

Passo 2 Aprire SciTE dal Terminale

- Nota che mentre SciTE è aperto in primo piano, la finestra del terminale utilizzata per avviarlo è ancora aperta in background. Inoltre, nota che la finestra del terminale utilizzata per avviare SciTE non visualizza più il prompt. **Perché il prompt non viene mostrato nel terminale?**

Il prompt non viene mostrato nel terminale perché il terminale è impegnato con quel processo.

Parte 2 Editor di Testo da Riga di Comando

- Nano si avvierà e caricherà automaticamente il file di testo space.txt. Sebbene il testo possa sembrare troncato o incompleto, non lo è. Poiché il testo è stato creato senza caratteri di ritorno a capo e l'a capo automatico (line wrapping) non è abilitato per impostazione predefinita, nano sta visualizzando una lunga riga di testo. **Quale carattere usa nano per rappresentare che una linea continua oltre i bordi dello schermo?**

Il carattere che usa nano per rappresentare che una linea continua oltre i bordi dello schermo è "\$".

Parte 3 Lavorare con i File di Configurazione

Passo 1 Localizzare i File di Configurazione

- **Perché i file di configurazione delle applicazioni utente sono salvati nella directory home dell'utente e non sotto /etc con tutti gli altri file di configurazione a livello di sistema?**

Perché se i file di configurazione delle applicazioni utente fossero sotto /etc con tutti gli altri file di configurazione a livello di sistema, ogni utente dovrebbe avere permessi di root per modificare le proprie preferenze, il che sarebbe un enorme rischio per la sicurezza. Salvandoli nella directory home, l'utente ha pieni permessi di lettura e scrittura sui propri file di configurazione senza necessitare di privilegi elevati.

Passo 2 Modificare e Salvare i File di Configurazione

- **Anche la finestra del terminale che era già aperta ha cambiato colore da verde a rosso? Spiega.**

No, la finestra del terminale che era già aperta non ha cambiato colore perché il file di configurazione ".bashrc" è salvato nella home directory dell'utente.

Passo 2 Modificare i File di Configurazione per i Servizi

- Dopo aver aperto con successo la homepage di nginx, guarda il messaggio di connessione nella finestra del terminale. **A cosa si riferisce il messaggio di errore?**

Il messaggio di errore si riferisce al file di configurazione del server web nginx.

- Puoi verificare se il server nginx è effettivamente spento pulendo prima la cronologia recente nel browser web, poi chiudendo e riaprendo il browser web, e infine andando alla homepage di nginx all'indirizzo 127.0.0.18080. **Appare la pagina web?**

No, la pagina web non riappare.

- Ricorda, poiché il file è memorizzato sotto /etc, avrai bisogno dei permessi di root per modificarlo. **Puoi modificare il file /etc/nginx/custom_server.conf con SciTE? Descrivi il processo di seguito.**

Sì, con SciTE è possibile modificare il file custom_server.conf. Essendo il file memorizzato sotto /etc, sul terminale eseguo il comando sudo su per avere i permessi di root e modificare il file eseguendo poi il comando sudo scite /etc/nginx/custom_server.conf. Per applicare le modifiche al server web Nginx, dovrò riavviare il servizio eseguendo il comando sudo systemctl restart nginx.

BONUS Prendere Familiarità con la Shell Linux

Parte 1 Basi della Shell

Passo 1 Accedere alla Riga di Comando

- Una pagina man fornisce informazioni dettagliate su un dato comando e tutte le sue opzioni disponibili. **Elenca alcune sezioni incluse in una pagina man.**

EXAMPLES: fornisce esempi pratici di come utilizzare il comando in scenari comuni. Questa sezione è spesso molto utile per capire l'applicazione pratica del comando.

OPTIONS: elenca e descrive in dettaglio tutte le opzioni (flag) che possono essere utilizzate con il comando, spiegando il loro comportamento e il loro effetto.

- **Qual è la funzione del comando `cp`? Quale comando useresti per trovare maggiori informazioni sul comando pwd? Qual è la funzione del comando pwd?**

Il comando cp (abbreviazione di "copy") è utilizzato per copiare file e directory. Permette di creare una copia esatta di un file o di una directory da una posizione all'altra nel filesystem. Il comando pwd (abbreviazione di "print working directory") viene utilizzato per stampare il nome della directory di lavoro corrente. Mostra il percorso completo della directory in cui ti trovi attualmente nel terminale.

Passo 4: Redirigere gli Output

- Usa l'operatore > per redirigere l'output di echo a un file di testo invece che allo schermo. Nessun output è stato mostrato. **È previsto? Spiega.**

No, perché questo operatore permette di redirigere l'output di un comando verso una posizione diversa dalla finestra del terminale corrente.

- Usa di nuovo l'operatore > per redirigere un diverso output di echo al file di testo some_text_file.txt. Ancora una volta, usa il comando cat per visualizzare il contenuto del file di testo some_text_file.txt. **Cosa è successo al file di testo? Spiega.**

Se some_text_file.txt non esiste, lo crea. Se some_text_file.txt esiste già, ne cancella completamente il contenuto esistente (lo sovrascrive). Il contenuto originale del file, se ce

n'era uno precedente all'esecuzione del comando, è stato completamente sovrascritto dal nuovo messaggio.

Passo 5: Redirigere e Accodare a un File di Testo

- Simile all'operatore >, anche l'operatore >> permette di redirigere dati ai file. La differenza è che >> accoda i dati alla fine del file indicato, mantenendo intatto il contenuto corrente. Per accodare un messaggio a some_text_file.txt, esegui il comando sottostante. **Cosa è successo al file di testo? Spiega.**

Il nuovo messaggio è stato aggiunto alla fine del file, senza cancellare il contenuto precedente.

Passo 6: Lavorare con i file nascosti in Linux

- Usa ls -l per visualizzare i file memorizzati nella directory home dell'utente analyst. **Quanti file vengono visualizzati?**

Vengono visualizzati 3 file.

- Usa il comando ls -la per visualizzare tutti i file nella directory home di analyst, inclusi i file nascosti. **Quanti file in più vengono visualizzati rispetto a prima? Spiega.**
È possibile nascondere intere directory aggiungendo un punto prima del loro nome?
Ci sono directory nascoste nell'output di ls -la sopra? Fornisci tre esempi di file nascosti mostrati nell'output di `ls -la` sopra.

Sì, è possibile nascondere intere directory aggiungendo un punto "." prima del loro nome. Questa è una convenzione comune nei sistemi Unix-like (come Linux) per indicare file e directory che non dovrebbero essere visualizzati per impostazione predefinita dai comandi come ls, senza l'opzione -a.

Sì, ci sono directory nascoste nell'output di ls -la sopra. Possono essere identificate cercando le righe che iniziano con d, che indica una directory, e il cui nome inizia con un punto. Tre esempi sono, ".bash_history", ".bash_logout" e ".bash_profile".

Parte 2: Copiare, Eliminare e Spostare File

Passo 1: Copiare File

- Esegui il comando sottostante per copiare some_text_file.txt dalla directory home alla cartella cyops_folder2. **Quali sono i file sorgente e destinazione? (usa percorsi completi per rappresentare i parametri)**

File Sorgente: /home/analyst/some_text_file.txt

Destinazione: /home/analyst/cyops_folder2/

Passo 3: Spostare File e Directory

- Esegui il comando sottostante per spostare some_text_file.txt da /home/analyst/cyops_folder2 di nuovo alla directory home. **Quale comando hai usato per completare l'attività?**

mv cyops_folder2/some_text_file.txt .

- **Quali sono i vantaggi dell'utilizzo della riga di comando Linux?**

L'utilizzo della riga di comando (CLI - Command Line Interface) in Linux offre numerosi vantaggi, specialmente per amministratori di sistema, sviluppatori e utenti avanzati:

1. **Efficienza e Velocità:** Una volta che si acquisisce familiarità con i comandi, è spesso molto più veloce eseguire operazioni complesse o ripetitive rispetto a un'interfaccia grafica (GUI). Non è necessario navigare tra menu o finestre.
2. **Automazione e Scripting:** La CLI è essenziale per l'automazione. È possibile combinare più comandi in script (file di testo eseguibili) per automatizzare attività complesse, routine di backup, installazioni di software, ecc. Questo è fondamentale per la gestione di server e infrastrutture complesse.
3. **Controllo Preciso:** I comandi offrono un controllo molto granulare su quasi ogni aspetto del sistema. Le opzioni dei comandi permettono di specificare esattamente come un'operazione deve essere eseguita, cosa che spesso non è possibile con le GUI che offrono un numero limitato di opzioni.
4. **Accesso Remoto:** La CLI è ideale per l'accesso remoto a server tramite SSH (Secure Shell). Non richiede una GUI, rendendola leggera, veloce e sicura per la gestione di sistemi che non hanno un'interfaccia grafica installata o che sono fisicamente distanti.
5. **Leggerezza delle Risorse:** L'interfaccia a riga di comando consuma molte meno risorse di sistema (CPU, RAM) rispetto a un ambiente desktop grafico completo. Questo la rende perfetta per server o sistemi con risorse limitate.
6. **Potenza e Flessibilità:** È possibile concatenare comandi (piping) e utilizzare operatori di ridirezione per creare operazioni complesse e personalizzate al volo. Ad esempio, l'output di un comando può diventare l'input di un altro, permettendo un'enorme flessibilità.
7. **Coerenza:** I comandi Linux tendono a rimanere coerenti tra le diverse distribuzioni e nel tempo, rendendo più facile il trasferimento di conoscenze e script tra ambienti diversi.
8. **Debugging e Diagnostica:** Molti strumenti di diagnostica e debugging per il sistema e le applicazioni sono basati su riga di comando, fornendo informazioni dettagliate essenziali per la risoluzione dei problemi.

In sintesi, la riga di comando è uno strumento potente e indispensabile per chiunque voglia avere un controllo profondo e efficiente sul proprio sistema Linux.