Benagoub Omar , SWD

Esame Finale - Linux UF

1. Filosofia Unix

La filosofia Unix è un insieme di norme culturali e progettuali che guidano lo sviluppo dei sistemi operativi Unix-like. Alla base promuove la creazione di programmi piccoli, modulari, che eseguono bene una sola funzione e che possono essere combinati per risolvere compiti complessi. Questa filosofia incoraggia semplicità, chiarezza e riutilizzabilità.

- Il primo principio, "Fai una cosa e falla bene", è evidente in strumenti come `grep` (ricerca di testo), `cut` (estrazione di campi) o `sort` (ordina righe). Ogni strumento gestisce un compito specifico con alta efficienza.
- Il secondo principio, "Tutto è un file", semplifica l'interazione con il sistema. Dispositivi, directory, flussi di input/output e socket sono trattati uniformemente come file. Ad esempio, `/dev/null` è un file speciale che scarta tutti i dati scritti, mentre `/proc/cpuinfo` è un file virtuale che contiene informazioni sul processore.
- Il terzo principio, "Costruisci software che funzioni insieme", promuove l'uso delle pipe (`|`). Comandi come `cat /var/log/syslog | grep error | less` combinano strumenti per ottenere obiettivi più complessi.

Questi principi rendono i sistemi Unix potenti e flessibili, favorendo la scriptabilità e l'automazione.

2. Shell di Login vs Non-login

Una shell di login si avvia quando un utente accede al sistema tramite console, SSH o TTY. Esegue i file di avvio come `/etc/profile`, `~/.profile` e `~/.bash_profile` (a seconda della shell). Questi file impostano variabili d'ambiente ed eseguono inizializzazioni.

Una shell non-login si avvia quando si apre un emulatore di terminale in una sessione grafica. Legge la configurazione da `~/.bashrc`, che contiene solitamente alias e impostazioni specifiche della shell.

Ad esempio, se si esporta una variabile in `~/.profile` (es. `export JAVA_HOME=/usr/lib/jvm/java-11-openjdk`), questa non sarà disponibile in un terminale grafico a meno che non venga richiamata anche in `~/.bashrc`. L'accesso via SSH (`ssh user@host`) avvia una shell di login, mentre avviare GNOME Terminal no.

3. **Analisi del Comando: **`awk -F: '{print $1}' /etc/passwd | sort -u`

Questo comando elabora il file `/etc/passwd` per estrarre una lista di nomi utente univoci presenti nel sistema. Analisi:

- `awk -F:` imposta il carattere `:` come delimitatore di campo.
- `'{print $1}'` stampa il primo campo di ogni riga, cioè il nome utente.
- `/etc/passwd` contiene le informazioni sugli account utente in formato separato da due punti.
- `| sort -u` ordina l'output alfabeticamente ed elimina i duplicati.

Utile per controlli di auditing e verifica degli account utente in sistemi complessi o automatizzati.

4. **Alternative a **`ls`

- `tree`: Mostra il contenuto di una directory in formato ad albero. Utile per visualizzare gerarchie. Esempio: `tree /etc`.
- `du`: Stima l'uso dello spazio. `du -sh /var/log` mostra quanto spazio usa la cartella.
- `find`: Cerca ricorsivamente file in base a criteri. Esempio: `find /home -name "*.sh"`.
- `stat`: Mostra metadati dettagliati (dimensione, permessi, data modifica). Esempio: `stat /etc/passwd`.

Ogni comando offre funzionalità aggiuntive rispetto a `ls`, utili per diagnostica e scripting.

5. **Stringa di Permessi: **`-rw-r--r--`

Questa stringa è mostrata con `ls -l`. Il primo carattere `-` indica un file regolare. I tre caratteri successivi `rw-` indicano che il proprietario ha permessi di lettura e scrittura. Il gruppo ha solo lettura (`r--`), così come gli altri (`r--`).

Questo consente solo al proprietario di modificare il file, mentre tutti possono leggerlo. Un `.` o `+` alla fine (non mostrati qui) indica ACL o attributi estesi.

6. Analisi Accesso

Con permessi `r--r-----`, la proprietaria (francesca) ha solo lettura. Il gruppo (studenti) può anch'esso solo leggere. Se `mario` fa parte del gruppo `studenti`, eredita il permesso di lettura, ma non può modificare il file. Nessuno può scrivere a meno di cambiare i permessi con `chmod` o `chown`.

7. Comandi di Gestione Utenti e Gruppi

sudo groupadd sysmaint

sudo usermod -aG sysmaint giulia

sudo mkdir -p /opt/maintenance/logs

sudo chown :sysmaint /opt/maintenance/logs

sudo chmod 770 /opt/maintenance/logs

- `groupadd` crea il gruppo `sysmaint`.
- `usermod -aG` aggiunge `giulia` al gruppo.
- `mkdir -p` crea la directory e i parent mancanti.
- `chown` cambia la proprietà del gruppo.
- `chmod 770` dà permessi completi a proprietario e gruppo, nessuno agli altri.

8. **Script: **`errorscan.sh`

#!/bin/bash

Controlla se è stato fornito un file

if [[-z "$1" || ! -f "$1"]]; then

  echo "Uso: $0 <file>"

  exit 1

fi

Conta righe che terminano con 'fail' (case-insensitive)

count=$(grep -iE 'fail$' "$1" | wc -l)

echo "Linee corrispondenti totali: $count"

Questo script valida l'input, cerca corrispondenze case-insensitive con `grep -iE`, e conta le righe che terminano con "fail". Utile per analizzare log di errori.

9. **File di Log: `/var/log/kern.log` e **`/var/log/dmesg`

- `/var/log/kern.log` raccoglie i messaggi del kernel durante il normale funzionamento. Registra eventi hardware, errori driver, moduli del kernel.
- `/var/log/dmesg` mostra i messaggi del buffer circolare del kernel all'avvio. È temporaneo ma utile per debug di avvio, rilevamento dispositivi e inizializzazione del kernel.

Entrambi sono essenziali per diagnosticare problemi di basso livello e hardware.

10. `dmesg | grep -i usb`

Questo comando filtra l'output di `dmesg` per mostrare messaggi relativi all'USB. `-i` abilita la ricerca case-insensitive. Serve per rilevare connessioni USB, errori o driver non caricati. Utile per diagnosticare dispositivi USB non funzionanti.

11. TCP vs UDP

TCP è un protocollo orientato alla connessione che garantisce consegna, ordine e integrità dei dati. Usa handshakes (SYN/ACK), controllo errori e ritrasmissione. Applicazioni come HTTPS e SSH usano TCP per affidabilità e sicurezza.

UDP è senza connessione, più veloce e tollera perdita di dati. Non garantisce ordine o consegna. DNS, VoIP e giochi online usano UDP per bassa latenza.

12. Comando `ss` sulla Porta 80

Comando: `ss -tuln | grep ':80'`

- `-tuln`: Mostra socket TCP/UDP in ascolto, senza risoluzione DNS.
- Se la porta 80 appare in stato `LISTEN`, indica che un server web (es. Apache o Nginx) è attivo e in ascolto per richieste HTTP.

Utile per verificare se un servizio è attivo sulla porta prevista.

13. **Connessione SSH: **`ssh marco@10.0.0.25`

Questo comando avvia una sessione shell sicura. Il client risolve `10.0.0.25`, stabilisce una connessione TCP, e inizia l'handshake SSH. Verifica la chiave del server (`~/.ssh/known_hosts`), negozia algoritmi di cifratura, autentica l'utente (password o chiave). Dopo l'autenticazione, stabilisce una sessione criptata per eseguire comandi remoti.

SSH garantisce riservatezza, integrità e autenticazione opzionale via chiave pubblica.

14. File Crontab

0 3 * * * sudo apt update && sudo apt upgrade -y

0 22 * * 5 find /home/utente/logs -name "*.log" -mtime +7 -delete

Questo file pianifica due attività:

- Alle 03:00 ogni giorno: aggiorna il sistema.
- Alle 22:00 il venerdì: elimina i file `.log` più vecchi di 7 giorni da `/home/utente/logs`. `find` usa `-mtime +7` e `-delete`.

15. SFTP vs FTP

SFTP opera su SSH (porta 22), fornendo autenticazione sicura e cifratura. FTP invece trasmette dati e credenziali in chiaro, rendendosi vulnerabile. SFTP cifra tutto il traffico, supporta autenticazione con chiavi e usa una sola connessione. È quindi preferibile per trasferimenti sicuri.

16. Comandi Apache2

sudo systemctl start apache2

sudo systemctl enable apache2

sudo systemctl status apache2

Questi comandi avviano Apache, lo impostano per l'avvio automatico e ne verificano lo stato. L'output di `status` mostra log, stato di esecuzione e ID processo. Essenziale per amministrazione e diagnostica di server web.