

V3b – Linux parancssor: a következő szint

Név: _____ Osztály: _____ Gép: _____

Mit fogsz csinálni? A V3-ban telepített Debian VM-en folytatod a parancssort, de most már olyan dolgokkal, amiket egy rendszergazda nap mint nap használ: súgó, keresés fájlokban, csővezetékek, jogosultságok, felhasználók, folyamatok, naplók, és a végén egy saját szkript. **Előfeltétel:** kész a V3 (fut a 12I-KissP-Debian VM, be tudsz lépni diak felhasználóval, és van internet). **Idő:** kb. 2 óra. Nem kell mindent megcsinálni: haladj sorban, ameddig eljutsz. A **G) rész** a legnehezebb. **Jegyzőkönyv:** Osztaly_V3b_Nev (pl. 12I_V3b_KissP). Ahol **[Kép]** áll, készíts képernyőképet.

Hasznos billentyűk: **Tab** = kiegészíti a parancsot vagy a fájlnevet · **↑** = az előző parancs · **Ctrl + C** = leállítja a futó parancsot · **Ctrl + L** = törli a képernyőt (vagy `clear`).

0. Előkészítés

- ☐ **1.** Indítsd el a VM-et, lépj be (diak / Passw0rd!).
- ☐ **2.** Telepítsd a súgót és néhány hasznos programot (lehet, hogy egy részük már fent van):

```
sudo apt update
sudo apt install -y man-db less file
```

A) Súgó: ha nem tudod, kérdezd meg a gépet

- ☐ **3.** `ls --help` | `less` → görgetés: **↑ ↓ Szóköz**, kilépés: **q**. Melyik kapcsolóval mutatja meg az `ls` a **rejtett** fájlokat is? – _____
- ☐ **4.** `man ls` → a teljes kézikönyv. Keresés benne: írd be `/human`, és nyomj **Entert**. Mit csinál a `-h` kapcsoló? _____
- ☐ **5.** Próbáld ki a saját mappádban (`cd ~`):

```
ls
ls -a
ls -lah
```

Milyen fájlok jelentek meg a `-a` hatására? (Ponttal kezdődnek.) _____ A Linuxban a ponttal kezdődő fájl **rejtett**. Ilyen pl. a `.bashrc`, a parancssor beállításai.

- ☐ **6.** `history` | `tail -5` → az utolsó 5 parancsod. Mit csinál a `!!`? Írd be: `whoami`, aztán `!!` → _____

B) Fájlok tartalma és keresés

- ☐ **7.** Az `/etc/passwd` fájl a gép felhasználóinak listája. Nézd meg többféleképpen:

```
head -3 /etc/passwd
tail -3 /etc/passwd
wc -l /etc/passwd
```

Hány sorból áll, azaz hány felhasználó van a gépen? _____

- ☐ **8.** Egy sor felépítése: diak:x:1000:1000:Diak,,,:/home/diak:/bin/bash. A mezőket : választja el.

Mező	Jelentése
diak	felhasználónév
x	a jelszó nem itt van (hanem az /etc/shadow-ban, amit csak a root olvashat)
1000	felhasználó-azonosító (UID); a 0 a root, 1000-től jönnek az „igazi” felhasználók
/home/diak	a saját mappa
/bin/bash	a parancsértelmező (shell)

Írasd ki csak a saját soradat: `grep diak /etc/passwd` → mi a UID-d? _____

- ☐ **9.** `cat /etc/shadow` → mit kapsz? _____ És `sudo cat /etc/shadow | grep diak`? A jelszó olvasható, vagy csak egy hosszú, értelmetlen szöveg? _____ Ez a jelszó **hash-e** (lenyomata), nem maga a jelszó.
- ☐ **10.** Keresés fájlnev szerint: hol van a gépen a hostname nevű fájl?

```
find /etc -name "hostname"
```

Eredmény: _____

- ☐ **11.** Keresés fájlok **tartalmában**: melyik fájlokban szerepel az `enp0s3` szó az `/etc` alatt?

```
sudo grep -r "enp0s3" /etc 2>/dev/null
```

Melyik fájl(ok)? _____ A `2>/dev/null` a hibaüzeneteket a „szemétbe” küldi, hogy ne zavarjanak.

C) Átírányítás és csővezeték (pipe)

Jel	Jelentése	Példa
>	a kimenetet fájlba írja (felülírja)	<code>ls / > lista.txt</code>
>>	a fájl végéhez fűzi	<code>date >> lista.txt</code>
	az egyik parancs kimenete a következő bemenete lesz	<code>ls / wc -l</code>

- ☐ **12.** Próbáld ki:

```
cd ~/labor
echo "első sor" > proba.txt
echo "második sor" >> proba.txt
cat proba.txt
echo "harmadik sor" > proba.txt
cat proba.txt
```

Mi lett a fájlban a végén, és miért? _____

- **13.** Csak a felhasználónevek, ábécérendben:

```
cut -d: -f1 /etc/passwd | sort
```

Mit csinál a `-d:` és az `-f1`? _____

- **14.** Hány felhasználó használja a `/usr/sbin/nologin` shell-t (ezek a „rendszer-felhasználók”, akikkel nem lehet belépni)?

```
grep nologin /etc/passwd | wc -l
```

→ _____

- **15.** Melyik a 3 legnagyobb mappa a `/usr` alatt?

```
sudo du -sh /usr/* 2>/dev/null | sort -h | tail -3
```

→ _____ **[Kép]** A 13–15. lépés kimenete.

D) Jogosultságok

- **16.** `ls -l ~/labor` → egy sor eleje így néz ki: `-rw-r--r--` 1 diak diak ...

Rész	Kinek szól?	Jelentése
-		fájl (d = mappa)
rw-	a tulajdonos (user)	olvashat, írhat, nem futtathat
r--	a csoport (group)	csak olvashat
r--	mindenki más (others)	csak olvashat

r = read (olvasás), w = write (írás), x = execute (futtatás).

- **17.** Írj egy szkriptet: `nano hello.sh`, a tartalma:

```
#!/bin/bash
echo "Szia, $USER! A gép neve: $(hostname)"
date
```

Futtasd: `./hello.sh` → milyen hibát kapsz? _____

- **18.** Adj futtatási jogot, és próbáld újra:

```
chmod +x hello.sh
ls -l hello.sh
./hello.sh
```

Mi változott az `ls -l` kimenetében? _____

- ☐ **19.** Vedd el mindenki olvasási jogát a `proba.txt`-ről, a sajátodat is: `chmod 000 proba.txt → cat proba.txt → _____` Add vissza: `chmod 644 proba.txt → ls -l proba.txt → -rw-r--r--?` Számokkal: $r = 4$, $w = 2$, $x = 1$. A `644` = tulajdonos 6 ($4+2$: rw), csoport 4 (r), mások 4 (r). Mi lenne a `755`? _____

E) Felhasználók és csoportok

- ☐ **20.** Hozz létre egy új felhasználót (a kérdésekre elég Entert nyomni, jelszónak adj meg egy könnyűt):

```
sudo adduser teszt
```

`tail -1 /etc/passwd →` mi a teszt UID-ja? _____ Létrejött a `/home/teszt` mappa? _____

- ☐ **21.** Lép át a teszt felhasználóra, és próbálj meg rendszergazda lenni:

```
su - teszt
whoami
sudo apt update
```

Mit ír a `sudo`? _____ Miért? _____ Vissza a saját felhasználódra: `exit`

- ☐ **22.** Hozz létre egy csoportot, és tedd bele a teszt-et:

```
sudo groupadd labor
sudo usermod -aG labor teszt
groups teszt
```

Milyen csoportokban van a teszt? _____ Vigyázz: a `-aG`-ből az `a` (*append, hozzáadás*) nagyon fontos. Nélküle a felhasználó **minden más** csoportjából kikerül!

- ☐ **23.** Takarítás: `sudo deluser --remove-home teszt` és `sudo groupdel labor`.

F) Folyamatok, szolgáltatások, naplók

- ☐ **24.** Indíts el egy „lassú” programot a **háttérben** (a `&` jel miatt), és keresd meg:

```
sleep 600 &
jobs
ps aux | grep sleep
```

Mi a `sleep` folyamat azonosítója (**PID**, a 2. oszlop)? _____

- ☐ **25.** Állítsd le: `kill <PID>` (a saját számodat írd!) → `jobs` → mit ír? _____
- ☐ **26.** A szolgáltatások (a háttérben folyamatosan futó programok) listája:

```
systemctl list-units --type=service --state=running
```

Írj fel hármat: _____

- ☐ **27.** A rendszernapló utolsó 20 sora: `sudo journalctl -n 20` · Csak a hibák a mostani indítás óta: `sudo journalctl -b -p err` Van hiba? Ha igen, egy példa: _____
- ☐ **28.** Erőforrások egy pillantásra:

```
free -h  
df -h /  
uptime
```

Szabad memória: _____ Szabad lemezhely: _____ Mióta fut a gép? _____ **[Kép]** A 26–28. lépés kimenete.

G) Kihívás: rendszerjelentés szkripttel

- ☐ **29.** Írj egy `jelentés.sh` szkriptet, ami **egy fájlba** (`~/jelentés.txt`) írja:
 - a mai dátumot és a gép nevét,
 - a VM IP-címét,
 - a szabad lemezhelyet és a szabad memóriát,
 - a felhasználók számát az `/etc/passwd`-ben,
 - a futó szolgáltatások számát.

Segítség: minden sor egy `echo` vagy egy parancs, `>>`-vel a fájl végére fűzve. Egy szám kiírására példa:

```
echo "Felhasznalok szama: $(wc -l < /etc/passwd)" >> ~/jelentés.txt
```

A `$ ( ... )` helyére a zárójelben lévő parancs eredménye kerül.

- ☐ **30.** Futtasd: `chmod +x jelentés.sh` → `./jelentés.sh` → `cat ~/jelentés.txt`. Ha kétszer futtatod, mi történik a fájljal? Hogyan javítanád, hogy mindig csak a friss jelentés legyen benne? _____ **[Kép]** A szkript és a `jelentés.txt` tartalma.
- ☐ **31.** Leállítás: `sudo poweroff`.

Kérdések (a jegyzőkönyvbe)

1. Mi a különbség a `>` és a `>>` között?
2. Mit jelent a `|` (pipe)? Írj egy saját példát, ami két parancsot köt össze.
3. Mit jelent a `-rwxr-x---` jogosultság? Ki mit tehet a fájljal?
4. Miért nem tudott a teszt felhasználó `sudo`-t használni, a diák pedig igen?
5. Mi a különbség a folyamat (process) és a szolgáltatás (service) között?

Leadás

- ☐ A jegyzőkönyvben a 3 képernyőkép (15., 28., 30. lépés) és a kérdések válaszai.
- ☐ A VM leállítva, a teszt felhasználó törölve.