Linux Core

[Comenzi Utile](https://docs.google.com/spreadsheets/d/1Cpd_2X7-OLUJWiid6lAKhBXV492QJzfuzpY5iTGSj44/edit?usp=drive_link)

Clonati acest document. În box-urile goale completati cu rezolvarile voastre (comenzi folosite). Dati apoi share la document si trimiteti-l pe discord in mesaj direct (nu pe grup).

### Exercitiul 1: Useri si permisiuni

**Cerinta 1:** Aveți o echipa formată din 3 membrii: **alice**, **bob** si **charlie**.

Creati acesti 3 useri. Dați switch la fiecare user si verificati ca a fost creat cu success.

|  |
| --- |
| sudo adduser alice  sudo adduser bob  sudo adduser charlie  su alice  su bob  su charlie  exit |

**Cerinta 2:** În echipa avem 2 roluri: developer si tester.

Creati 2 grupuri numite “**developers**” si “**testers**”.

În grupul de developers ii avem pe alice si pe bob iar in grupul testers pe bob si charlie (bob este in 2 grupuri).

Verificati pentru fiecare user că este în grupurile corecte folosind comanda **id** sau **groups**.

|  |
| --- |
| sudo addgroup developers  sudo addgroup testers  sudo usermod -aG developers alice  sudo usermod -aG developers bob  sudo usermod -aG testers bob  sudo usermod -aG testers charlie  groups alice  id bob  id charlie |

**Cerinta 3:** Fiecare role poate accesa doar anumite directoare ale proiectului.

Creați 2 directoare numite /developers si /testers in radacina sistemului de fisiere. În fiecare director pot scrie doar userii grupului respectiv, dar poate citi și executa oricine. Verificati permisiunile directoarelor folosind comanda ls.

|  |
| --- |
| mkdir /developers  mkdir /testers  ls -l /developers  ls -l /testers  sudo chown :developers /developers  sudo chown :testers /testers  chmod 775 /developers  chmod 775 /testers |

**Cerinta 4:** Dati switch la userul alice si creati un fisier cu numele [script.sh](http://script.sh/) in directorul /developers. In script puneti:

|  |
| --- |
| echo "Salut developer" |

Dati permisiuni de execuție pentru toata lumea pe acest fișier, dar de modificare doar pentru grupul developers.

Verificati ca toți userii pot executa acest fișier, dar doar useri din grupul developers îl pot modifica.

|  |
| --- |
| su alice  cd /developers  echo ‘echo “Salut developer”’ > script.sh  ls -l script.sh  chmod 575 [script.sh](http://script.sh/)  exit  sudo chown :developers script.sh  ls -l script.sh |

**Cerinta 5:** În timp membrii celor 2 echipe au fost comansati într-o singura echipa numita “**tech**”.

Creati un nou group numit “tech”.

Mutați group owner-ul pentru toate fișierele din directorul /developers si /testers la grupul “**tech**”.

Stergeti grupurile developers si testers.

Verificati ca încă mai puteți edita/rula scripturile cu userul charlie.

|  |
| --- |
| sudo addgroup tech  sudo chgrp -R tech /developers /testers  sudo delgroup developers  sudo delgroup testers  su charlie  ./script.sh  vim script.sh |

### Exercitiul 2: Pipelines si Fluxuri de date

Rezolvați următoarele cerințe folosind operatorul pipe |.

**Cerinta 1:** Găsește toate liniile de log de sistem care contin cuvantul error, indiferent de majuscule.

Hint: Vezi /var/log/syslog

|  |
| --- |
| cat /var/log/syslog | grep -i error |

**Cerinta 2:** Găsește utilizatorii care au setat ca default shell /bin/bash. Câți utilizatori nu folosesc acest shell?

Hint: Lista de utilizatori o gasiti in /etc/passwd

|  |
| --- |
| cat /etc/passwd | grep /bin/bash  cat /etc/passwd | grep -v /bin/bash | wc -l |

**Cerinta 3:** Afișați primele și ultimele 5 comenzi utilizate de voi.

Hint: history, head, tail

|  |
| --- |
| history | head -n 5  history | tail -n 5 |

**Cerinta 4**.Încercați să creați un user fara drepturi de sudo. Cum putem sa ignoram erorile de permisiuni (a.i. acestea sa nu mai apara in consola)?

|  |
| --- |
| adduser newuser 2> /dev/null |

**Cerinta 5:** Afișați toate comenzile rulate (cu succes) cu sudo de toti userii.

Hint: /var/log/auth.log

|  |
| --- |
| sudo cat /var/log/auth.log | grep -i command | grep -w sudo | wc -l |

**Cerinta 6:** Downloadati fisierul app-1.log folosind comanda **wget** fisierul:

<https://raw.githubusercontent.com/amihai/devops/refs/heads/main/python/log-generator/app-1.log>

Explorați puțin continutul fisierului: afisati primele si ultimele 5 linii si cate linii are fișierul

|  |
| --- |
| wget https://raw.githubusercontent.com/amihai/devops/refs/heads/main/python/log-generator/app-1.log  cat spp-1.log | wc -l  head -n 5 app-1.log  tail -n 5 app-1.log |

**Cerinta 7:** Ce useri au erori deoarece le-a expirat tokenul (NU Permission denied).

|  |
| --- |
| cat app-1.log | grep -i “expired” |

**Cerinta 8 (bonus):** Incercati sa extrageti in fisiere txt separate urmatoarele informatii din logurile de mai sus:

* Lista cu numele unice ale tuturor userilor ce au avut cel putin o eroare
* Lista de request\_id-uri unice.
* Ce request\_id-uri nu au nicio eroare

Hint: incercati sa folositi comenzile: awk, sort, uniq

|  |
| --- |
| cat app-1.log | grep -i error | awk '{print $4}' | awk -F= '{print $2}' | sort | uniq  cat app-1.log | awk -F"request\_id=" '{print $2}' | sort | uniq  cat app-1.log | grep -vi error | awk -F"request\_id=" '{print $2}' | sort | uniq |

### Exercitiul 3: Procese

**Cerinta 1**: Porniti un process ce ruleaza comanda **sleep Infinity** intr-un terminal. Deschideți alt terminal și terminati cu eliberare de resurse procesul (graceful).

|  |
| --- |
| terminal1: sleep infinity  terminal2: ps -aux  terminal2: kill <pid> |

**Cerinta 2:** Downloadati următorul script folosind comanda **wget**:

<https://raw.githubusercontent.com/amihai/devops/refs/heads/main/linux/die-hard.sh>

Da-ti permisiuni de execuție pe el si rulati-l.

Omorati programul. 🙂

|  |
| --- |
| wget https://raw.githubusercontent.com/amihai/devops/refs/heads/main/linux/die-hard.sh  chmod a+x die-hard.sh  ./die-hard.sh  ps aux  kill <pid>  kill -9 <pid> |

**Cerinta 3:** Deschideti un tab de firefox. Căutați procesul si terminati-l.

|  |
| --- |
| ps -aux | grep firefox  killall firefox |

**Cerinta 4:** Care sunt PID-urile primele 5 programe ca utilizare de CPU. Dar ca memorie RAM.

Hint: Puteti utiliza comanda top sau ps cu sort dupa memorie.

|  |
| --- |
| top -o %CPU  top -o %MEM |

**Cerinta 5:** Cate procese are pornite userul curent? Dar root?

|  |
| --- |
| ps aux | awk '$1 == "vvovnen+" {count++} END {print count}'  ps aux | awk '$1 == "root” {count++} END {print count}' |

**Cerinta 6:** Downloadati scriptul următor si dati-i permisiuni de execuție:

<https://raw.githubusercontent.com/amihai/devops/refs/heads/main/linux/run-forever.sh>

* Rulati scriptul in background in asa fel incat daca se inchide terminalul sa nu se omoare si scriptul. Redirectati outputul scriptului catre un fisier de log cu numele run-forever.log
* Omorati terminalul si verificati ca scriptul încă rulează folosind comanda ps si folosind tail -f pe fisierul de loguri.
* Omorati scriptul si verificati ca a fost oprit.

|  |
| --- |
| wget <https://raw.githubusercontent.com/amihai/devops/refs/heads/main/linux/run-forever.sh>  chmod a+x run-forever.sh  nohup ./run-forever.sh 1>run-forever.log &  Inchide termonalul  ps aux | grep run-forever.sh  tail -f run-forever.log |