

## Test Practic Final USO

Varianta 133, 30 ianuarie 2023

Nume: \_\_\_\_\_


Grupa: \_\_\_\_\_

### Indicații

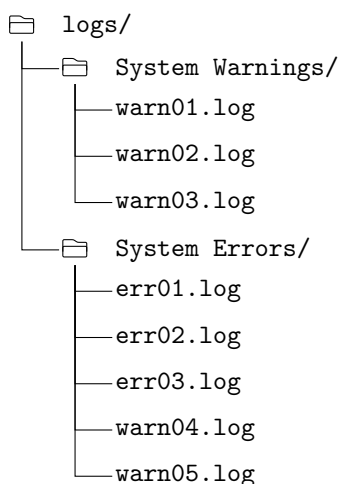
- Testul durează 100 minute și conține 6 subiecte. Fiecare subiect se punctează cu maxim 20 de puncte pentru o rezolvare corectă și completă. Rezultatul se trunchiază la 100 de puncte.
- Se pot obține punctaje parțiale pentru un subiect numai dacă acest lucru este specificat.
- Ordinea de rezolvare a subiectelor este la alegerea voastră.
- Rezolvările subiectelor pot fi realizate utilizând **doar** aplicații disponibile în linia de comandă.
- Subiectele se rezolvă pe mașinile virtuale puse la dispoziție fiecărui student. Orice rezolvare pe mașina locală este nevalidă.
- Dacă nu este menționată calea completă a unui fișier sau director, acesta trebuie creat în `/home/student`.
- Platforma pe care o să fie dat examenul este <https://guacamole.grid.pub.ro>

Punctaj total

### Subiecte

1. (a) În directorul „`home/logs`” aveți următoarea structură de fișiere: **Notă:** Directoarele au  la începutul numelui.

3



Faceți următoarele modificări: creați fișierele „`err04.log`” în directorul „`System Errors`” și „`warn06.log`” în directorul „`System Warnings`” folosind o singură comandă.

- (b) Mutați fișierele „`warn04.log`” și „`warn05.log`” în directorul „`System Warning`”.
- (c) Creați o legătură simbolică către fișierul „`warn01.log`” numită „`warn-link`” în directorul „`logs`”.
- (d) Redenumiți toate fișierele „`warn*`” din directorul „`System Warnings`” în „`sys_warn*`” folosind o singură comandă.  
De exemplu, fișierul „`warn03.log`” va deveni „`sys_warn03.log`”.

5

5

7

2. (a) Creați un script ce deschide 30 de procese „`sleep 100`”, apoi omorâți-le pe toate din linia de comandă.

5

- (b) Afișați **doar** pid-ul, ppid-ul și procentul de memorie utilizată pentru primele 3 procese din sistem ordonate descrescător după procentul de procesor utilizat.

5

- (c) Conectați-vă folosind „**TCP**” la adresa „**tcpbin.com**” pe portul 4242 și trimiteți stringul „**I’ll be back**”. 5
- (d) Sortați toate obiectele JSON din fișierul „**ip-names.json**” aflat la adresa „**http://elf.cs.pub.ro/uso/res/final/133/ip-names.json**” după câmpul „**ip\_address**”. 5
3. (a) Afișați **doar** numele utilizatorilor cu shellul „**/bin/bash**”. 5
- (b) Faceți modificările necesare pe utilizatorul student pentru a putea folosi „**sudo**” fără a vi se cere parola. 5
- (c) Creați un fișier numit „**hi\_ana**” în directorul „**/home/student**” și faceți modificările necesare astfel încât **doar** utilizatorii „**student**” și „**ana**” să poată modifica fișierul 5
- (d) Faceți modificările necesare astfel încât parola utilizatorului „**ana**” să fie „**are-mere**”. 5
4. (a) Creați un director nou numit „**my-app**” în directorul „**/home/student**” și inițializați un repository local nou în el. 5
- (b) Aplicați patch-ul aflat la adresa „**http://elf.cs.pub.ro/uso/res/final/133/0001-Add-README.md-file.patch**” peste repository-ul nou creat. 5
- „**Hint:**” „**git am**”.
- (c) Creați un fișier nou numit „**main.c**” și faceți un commit cu el. 5
- (d) Adăugați un fișier „**.gitignore**” ce ignoră fișierele cu extensia „**.old**” sau „**.tmp**”. 5
5. (a) Descărcați fișierul „**http://elf.cs.pub.ro/uso/res/final/133/USO\_encoded**”. În fișierul „**USO\_encoded**” se află un fișier encodat folosind „**base64**”. Decodificați fișierul și aflați tipul lui. 5
- (b) Faceți setările necesare pentru a preveni pachetul „**ltrace**” din a se mai actualiza în viitor. 5
- (c) Creați un script cu numele „**get\_ip.sh**” care primește ca argument numele unui website și printează adresa IP asociată. 5
- „**Exemplu:**” „**/get\_ip upb.ro**” va afișa „**141.85.220.33**”
- (d) Folosind o singură comandă, afișați fișierele cu extensia „**.h**” din directorul „**/usr/include**” care au dimensiunea cuprinsă între „**50K**” și „**100K**”, împreună cu dimensiunea lor. 5
- „**Exemplu output:**”  
52K /usr/include/linux/netfilter/nf\_tables.h  
92K /usr/include/linux/video\_dev2.h  
...
6. (a) Scrieți un script numit „**many\_files**” care crează 100 de fișiere text, de la „**00.txt**” la „**99.txt**” și le împarte în directoare diferite, fiecare fișier fiind la final aflat în directorul cu numele „**index fișier / 10**”. 5
- (i.e. directorul „**0/**” va conține fișierele „**00.txt - 09.txt**”, directorul „**1/**” fișierele „**10.txt - 19.txt**” etc.)
- (b) Realizați modificările necesare astfel încât scriptul de la punctul anterior să poată fi rulat de oriunde din sistem ca „**many\_files**”. 5
- (c) Scrieți un script care generează un număr aleator mai mic ca 100 și îl printează la stdout. 5
- (d) Modificați scriptul de la subpunctul c astfel încât acesta să citească de la tastatură numere și să afișeze dacă numărul introdus este mai mare sau mai mic ca cel generat, până când utilizatorul găsește numărul generat inițial. 5
- „**Exemplu rulare:**”  
„**./numbers.sh**”  
25  
Go higher  
60  
Go lower  
30

Go lower  
27  
Good job!