

Nume: _____ Prenume: _____

Județ: _____



Olimpiada de Informatică Aplicată AcadNet 2026

Calculatoare & Sisteme de Operare

Secțiunea 9–10

Citiți cu atenție toate subiectele înainte de a începe rezolvarea. Aveți 10 minute să adresați întrebări pentru clarificări înainte de a contoriza timpul de rezolvare.

Probele sunt independente, astfel că, după fiecare task (exercițiu sau subpunct) va trebui să atenționați unul dintre supraveghetori să vă puncteze. Pentru a obține punctajul pe un subpunct trebuie să arătați rezolvarea acestuia și să demonstrați că task-ul a fost realizat cu succes. Nu se vor acorda punctaje parțiale per subpuncte.

Pentru autentificarea pe mașina virtuală veți folosi <https://guacamole.grid.pub.ro>. Folosiți credențialele primite din partea supraveghetorilor. Folosiți ghidul guacamole primit din partea supraveghetorilor.

Nu aveți voie să modificați parolele pentru contul de administrator. În cazul modificării unei parole de administrator, sunteți descalificați.

Probele sunt punctate în funcție de dificultatea acestora.

Timp de lucru: 2 ore

Question	Points	Score
1	10	
2	5	
3	10	
4	10	
5	10	
6	10	
7	10	
8	10	
9	5	
10	10	
11	10	
Total:	100	

1. Interfața în linia de comandă (10p)

- (2 puncte) Afișați valoarea variabilei de mediu PATH astfel încât fiecare director din aceasta să apară pe o linie separată.
- (2 puncte) Determinați locația executabilului asociat comenzii `python3` și afișați directorul în care se află acesta.
- (3 puncte) Editați prompt-ul shell-ului astfel încât utilizatorul să fie **hack3r** și hostname-ul să fie **acadnet2026**.
- (3 puncte) Scrieți un oneliner care afișează un string aleatoriu de 5 caractere alfabetice lowercase.

2. Utilizatori (5p)

- (1 punct) Scrieți doar parolele criptate din fișierul `/etc/shadow` în fișierul **parole_cript** din directorul user-ului curent `~/`.

- (b) (1 punct) Schimbați permisiunile fișierului astfel încât utilizatorul curent să nu aibă permisiuni, grupurile să aibă toate permisiunile, iar other doar să citească.
- (c) (2 puncte) Creați un user nou numit **parole** și un grup nou numit **criptate**, adăugați user-ul **parole** în grupul **criptate**.
- (d) (1 punct) Schimbați owner-ul fișierului în **parole**, iar grupul fișierului în **criptate** folosind o singură comandă.

3. Utilizarea sistemului de fișiere (10p)

- (a) (2 puncte) Creați următoarea structură de fișiere:

```
Music/
├── Pop/
│   ├── Michael Jackson/
│   │   ├── Madonna.txt
│   │   ├── Queen.pdf
│   │   ├── Taylor Swift.txt
│   │   └── Rihanna.pdf
│   └── Rock and Roll/
│       ├── KISS.txt
│       └── Led Zeppelin.txt
└── Classic/
```

- (b) (2 puncte) Lucrând în directorul **Music/**, mutați fișierele cu extensia **.pdf** din folderul **Pop/** în folderul **Rock and Roll/**.
- (c) (3 puncte) Lucrând în directorul **Rock and Roll/**, folosiți o singură comandă pentru a schimba extensia tuturor fișierelor din folderul curent în **.mp3** (astfel numele unui fișier se va modifica din **<filename>.<extensie>** în **<filename>.mp3**).
- (d) (3 puncte) Lucrând în directorul **~/**, creați 50 de fișiere în folderul **Michael Jackson/** cu numele de forma **<song_NUM>**, unde NUM parcurge toate numerele de la 1 la 50.

4. distros (10p)

În folderul **~/distros** veți găsi mai multe foldere root aparținând unor distribuții variate de Linux.

- (a) (1 punct) Instalați utilitarul **arch-chroot** la nivel de sistem.
- (b) (1 punct) Afișați doar numărul de pachete software ale root-ului **fedora** ce pot fi actualizate.
- (c) (3 puncte) Extrageți din cache-ul de pachete al root-ului **fedora** fișierul **caddy*-x86_64.rpm** și instalați-l pe sistemul principal.
- (d) (2 puncte) Afișați numărul de pachete software orfane din root-ul **arch** și apoi eliminați-le din sistem.
- (e) (3 puncte) Extrageți din cache-ul de pachete al root-ului **arch** fișierul **yazi*-x86_64.pkg.tar.zst** și instalați-l pe sistemul principal.

5. **Procese** (10p)

- (a) (2 puncte) Afișați pe terminal primele 5 procese ale utilizatorului curent, ordonați-le în ordine crescătoare după procentul de CPU utilizat.
- (b) (1 punct) Afișați toate PID-urile al căror PPID este egal cu 1.
- (c) (3 puncte) Programați ca mesajul “**Acadnet 2026**” să apară în terminal pe data de 15 mai 2026.
- (d) (4 puncte) Vrem să folosim semnalele de user, **SIGUSR1** și **SIGUSR2** pentru a efectua comunicarea între procese, Inter-Process Communication, folosind doar shell-ul.
Scrieți un one-liner în care un terminal învață să prindă semnalul **SIGUSR1**, iar de fiecare dată când primește un ping de pe un alt terminal, să răspundă scriind în consolă mesajul “**PING PRIMIT**”.

Hint: trap

6. **Master of Makefile** (10p)

- (a) (4 puncte) Intrați în directorul **Subiect_Makefile/**, creați un fișier Makefile care să compileze atât **main.cpp** și **fake_decoder.cpp** într-un executabil cu numele **bad_decoder**.
Fișierul trebuie să conțină regula **build**, unde vor fi compilate sursele menționate. Afișați mesajul rulând executabilul.
- (b) (4 puncte) Adăugați o regulă de compilare a fișierului obiect **decoder.o** astfel încât să fie linkat cu fișierul **main.cpp**. Numele executabilului va fi **good_decoder**. Adăugați regulile **clean** și **run**: **clean** va șterge fișierele executabile și obiect, cu excepția fișierului **decoder.o**, iar **run** va executa fișierul obținut după compilare. Afișați mesajul decodat în terminal.
- (c) (2 puncte) Nu ștergeți sau modificați fișierul **pack** în orice fel. Creați o nouă regulă cu numele **pack** în Makefile astfel încât toate fișierele cu extensia **.cpp** și **Makefile** să fie puse într-o arhivă de tip **.tar.gz** cu numele **master_of_makefile.tar.gz**.

7. **Personalizare Shell** (10p)

- (a) (3 puncte) Instalați **zsh** și framework-ul **oh-my-zsh** pe stația voastră.
Important: Nu setați zsh drept shell default pe masina voastră!
- (b) (3 puncte) Configurați plugin-urile de **zsh** pentru **npm**, **docker** și **git**.
- (c) (1 punct) Setați tema **awesomepanda** în **zsh**.
- (d) (3 puncte) Efectuați setările necesare pentru **zsh** astfel încât la tastarea comenzilor:
 - **boop** să afișeze “**beep boop**”
 - **brb** să înlocuiască “**cd ~**”

Setările trebuie să fie persistente.

8. **Componente hardware** (10p)

- (a) (2 puncte) Afișați doar numărul de core-uri al procesorului sistemului.
- (b) (4 puncte) Afișați câtă memorie ram este utilizată în total, incluzând memoria cache.
- (c) (4 puncte) Scrieți un script **bash** care afișează mesajul “**PCIeBest**” dacă numărul de dispozitive PCI este mai mare decât numărul de dispozitive USB conectate sau “**USBBest**” în caz contrar.

9. **no-more-french-package** (5p)

La această problemă va trebui să lucrați cu pachetul de limbă franceză. Și nu cel la care vă gândiți. Este vorba de **wfrench**, o listă cu toate cuvintele în limba franceză.

- (a) (1 punct) Instalați pachetul **wfrench** pe sistem și listați pachetul de care acesta depinde.
- (b) (1 punct) Localizați pe sistem calea absolută a fișierului **french** descărcat la subpunctul anterior.
- (c) (2 puncte) Identificați cuvântul din fișierul **french** care îndeplinește simultan următoarele condiții:
 - Începe cu litera **f** și se termină cu litera **o**;
 - A doua literă este **a** sau **e**;
 - A treia literă este **r**;
 - Conține exact **7** litere.
- (d) (1 punct) Atât pentru astăzi. Timpul să faceți ce vă doriți cu adevărat: ștergeți pachetul de limbă franceză de pe sistem.

10. **Acadcore** (10p)

Tot lovindu-te de problemele care apar când folosești containere pentru a orchestra totul, te decizi că ai avut destul și vrei să te întorci la mașini virtuale.

Pentru a-ți îndeplini scopul, te gândești să te folosești de ultimul **.iso** pe care l-ai folosit înainte de revoluția k8s, dar rapid observi o problemă.

Discul este partiționat complet incorect, iar formatarea partițiilor este greșită!

Dorind să arăți superioritatea mașinilor virtuale prietenului tău Lucian, ramâi neclintit și rezolvi imaginea astfel:

- (a) (3 puncte) Copiază **~/Tc/tc.iso** într-un fișier nou și repartizionează-l astfel: Prima partiție trebuie marită până la 80MB (fără a o șterge) și declarată bootabilă. Spațiul rămas apoi trebuie umplut cu încă o partiție primară.
- (b) (3 puncte) Reformatează cea de-a doua partiție cu sistemul de fișiere ext4.
- (c) (4 puncte) Creează un fișier normal în cea de a doua partiție numit **hydra.txt**

Hint: Pentru a ieși din consola serial QEMU trebuie apasat <Ctrl-a> x

11. **Workflow git: ramuri și conflict** (10p)

- (a) (3 puncte) Creați directorul **~/lab_git** și inițializați-l ca repository git. Configurați doar la nivel de repository numele și email-ul pentru git. Verificați configurațiile locale.
- (b) (3 puncte) Creați fișierul **note.txt** cu trei linii: alfa, beta, gama. Faceți două commit-uri separate: primul doar cu alfa, al doilea cu toate cele trei linii. Mesajele commit-urilor: **“Initial alfa”** și **“Add beta gama”**.
- (c) (2 puncte) Afișați istoricul commit-urilor doar cu hash scurt și mesaj, pe o singură linie per commit. Apoi afișați conținutul fișierului **note.txt** așa cum era la primul commit, fără a modifica starea curentă.
- (d) (2 puncte) Creați o ramură nouă feature și pe ea modificați linia beta în **BETA-MODIFICAT**. Faceți commit cu mesajul **“Modificare beta”**.