

# Olimpiada de Informatică

## Aplicată AcadNet 2026

Calculatoare 9-10

Soluții

### § 1 Interfața în linia de comandă

a. `echo "$PATH" | tr ':' '\n'`

Altă variantă:

```
sed 's:/\n/'
```

b. `dirname $(which python3)`

c. `PS1 = \[ \e]0;hack3r@acadnet2026: \w\a\]\u@h:\w\ $`

Înlocuiți `\u`, respectiv `\h`.

d. `echo $RANDOM | tr '[0-9]' '[a-j]'`

Altă variantă:

```
sudo apt install pwgen ; pwgen 5 1 --no-numerals --no-capitalize
```

### § 2 Utilizatori

a. `sudo cat /etc/shadow | cut -d ":" -f 2 | grep -w -v '\*' > ~/parole_cript`

b. `chmod 074 parole_cript`

c. `sudo useradd parole`  
`sudo groupadd criptate`  
`sudo usermod -aG criptate parole`

d. `chown parole:criptate parole_cript`

## § 3 Utilizarea sistemului de fișiere

- a. 

```
mkdir Music
cd Music
mkdir Pop "Rock and Roll" Classic
cd Pop
mkdir "Michael Jackson"
touch Madonna.txt Queen.pdf "Taylor Swift.txt" Rihanna.pdf
cd ../"Rock and Roll"
touch KISS.txt "Led Zeppelin.txt"
```
- b. 

```
mv Pop/*.pdf "Rock and Roll"/
```
- c. 

```
for f in *; do mv "$f" "${f%.*}.mp3"; done
```
- d. 

```
touch Music/Pop/"Michael Jackson"/song_{1..50}
```

## § 4 distros

- a. 

```
sudo apt update
sudo apt install arch-install-scripts
```
- b. 

```
sudo arch-chroot ~/distros/fedora
dnf check-update | wc -l ( >=368 )
```
- c. 

```
mv ~/distros/fedora/var/cache/dnf/caddy* ~/
sudo apt install alien
alien --to-deb ~/.rpm
sudo apt install ~/.deb
```

Altă variantă, prin instalare manuală:

```
sudo apt install p7zip-full sau rpm2cpio
7z x ~/yazi-26.1.22-4-x86_64.pkg.tar.zst
```

sau

```
rpm2cpio ~/.rpm > ~/yazi.cpio
```

```
cat ~/.cpio | cpio -idmv
sudo mv ~/usr/* /usr/
```

- d. 

```
sudo arch-chroot ~/distros/arch
pacman -Qtdq | sudo pacman -Rns -
```

e. În chroot:

```
pactree yazi # pentru a identifica dependențele
sudo mv ~/distros/arch/var/cache/pacman/pkg/jemalloc* ~/
sau sudo apt install jemalloc
```

```

sudo mv ~/distros/arch/var/cache/pacman/pkg/liblua54* ~/
sau sudo apt install liblua5.4
sudo mv ~/distros/arch/var/cache/pacman/pkg/oniguruma* ~/
sau sudo apt install libonig5
sudo mv ~/distros/arch/var/cache/pacman/pkg/yazi* ~/
tar xf ~/.pkg.tar.zst
sudo mv ~/usr/* /usr/

```

## § 5 Procese

- `ps -u $USER --sort=%cpu | head -n 6`
- `pgrep -P 1`
- `echo "wall 'Acadnet 2026'" | at 12:00 May 15 2026`
- Terminal 1 - receiver:

```

trap 'echo "PING primit"' SIGUSR1; echo "PID_Terminal_1: $$"; while true;
do sleep 1; done

```

Terminal 2 - sender:

```
kill -SIGUSR1 <PID_Terminal_1>
```

## § 6 Master of Makefile

- ```

build:
    g++ main.cpp decoder.o -o decoder

```
- ```

build:
    g++ main.cpp decoder.o -o decoder
    g++ main.cpp fake_decoder.cpp -o other_decoder

run: build
    ./decoder

clean:
    rm -f decoder other_decoder

```
- ```

.PHONY: pack
pack:
    tar -cvzf master_of_makefile.tar.gz *.cpp Makefile

```

## § 7 Personalizare Shell

- ```

sudo apt install zsh
sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"

```

- b. În fișierul `/.zshrc`:

```
plugins=(git npm docker)
```

- c. `ZSH_THEME="awesomepanda"`

- d. `alias boop="echo beep boop"`  
`alias brb="cd ~"`

```
source ~/.zshrc # pentru a persista schimbările
```

Dacă ați setat zsh ca shell default, scrieți “bash” în terminal pentru reveni la shell-ul de bază.

## § 8 Componente hardware

- a. `lscpu | grep "Core(s) per socket" | cut -d':' -f2 | xargs`

- b. `free | awk '/^Mem:/ {print $2 - $4}'`

Altă variantă:

```
free | awk '/^Mem:/ {print $3 + $6}'
```

- c. 

```
#!/bin/bash
A=$(lspci | wc -l)
B=$(lsusb | wc -l)
if [ $((A - B)) -gt 0 ]; then
    echo "PCIeBest"
else
    echo "USBBest"
fi
```

## § 9 no-more-french-package

- a. `sudo apt-get install wfrench`

```
apt-cache depends wfrench
```

Pachetul de dependență este debconf.

- b. `find / -type f -name french 2>/dev/null`
- c. `grep '^f[ae]r.*o$' /usr/share/dict/french`
- d. `sudo apt-get purge wfrench`

## § 10 Acadcore

- a. `cp tc.iso new.iso`

```
fdisk new.iso
```

Apoi

```
e,+80M; a; n
```

b. `doas kpartx -a new.iso; mkfs.ext4 /dev/mapper/loop0p2`

c. 

```
mkdir root
mount /dev/mapper/loop0p2 root
touch hydra.txt
sudo kpartx -d new.iso
```

În loc de kpartx se poate folosi și folosirea losetup.

## § 11 Workflow git: ramuri și conflict

a. 

```
mkdir ~/lab_git
cd ~/lab_git
git init
git config --local user.name "Nume Prenume"
git config --local user.email "email@example.com"
git config --local --list
```

b. 

```
echo "alfa" > note.txt
git add note.txt
git commit -m "Initial alfa"
printf "alfa\nbeta\ngama\n" > note.txt
git add note.txt
git commit -m "Add beta gama"
```

c. 

```
git log --oneline
git show HEAD~1:note.txt
```

d. 

```
git checkout -b feature
sed -i 's/beta/BETA-MODIFICAT/' note.txt
git add note.txt
git commit -m "Modificare beta"
```