



Olimpiada de Informatică Aplicată AcadNet 2024

Etapă Națională

Secțiunea calculatoare, clasele 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.

Pentru unele dintre probe veți avea nevoie să folosiți un repository de tip git ce conține fișierele necesare. Pentru a avea aceste fișiere în mașina virtuală, executați comanda `git clone https://github.com/acadnet2024/acadnet-2024-etapa-nationala.git`. Aceasta va clona local repository-ul în directorul `acadnet-2024-etapa-nationala`. Pentru probele ce depind de acest repository este menționat acest lucru în dreptul numărului probei.

Timp de lucru: 120 min

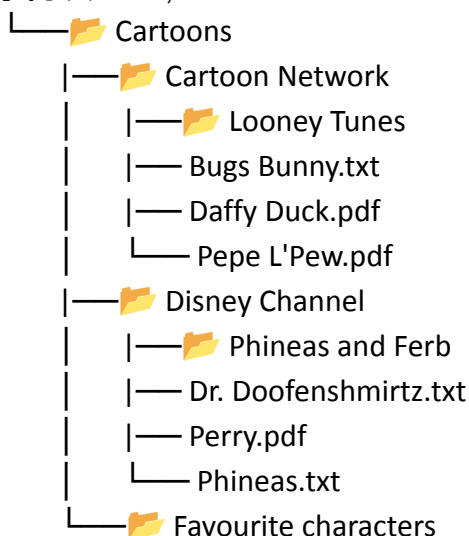
Punctaj maxim: 100 puncte

Mult succes!



[10p] Proba 1

[3p] (a) Creați următoarea structură de directoare și fișiere:



[1p] (b) Folosind o singură comandă, aflându-vă în directorul *"Cartoons"* creați în directorul *"Cartoon Network"* un fișier ascuns *"Adventure Time"*, în care să scrieți textul *"Finn and Jake"*.

[2p] (c) Copiați toate fișierele cu extensia *.txt* din cadrul ierarhiei în directorul *"Favourite characters"*.

[4p] (d) Creați un nou director *"Teen Titans"* în cadrul directorului *"Cartoon Network"*. În directorul *"Teen Titans"* creați 50 de fișiere cu numele de forma *new_character_NUM*, unde *NUM* este un număr de la 1 la 50.

[10p] Proba 2 (veți lucra în repository în directorul *senses*)

[2p] (a) Copiați recursiv tot conținutul directorului *"source_directory"* în directorul *"destination_directory"*. Suprascrierea unui fișier deja existent se va face doar dacă este mai vechi (veți suprascrive fișierele cu același nume **doar** dacă sunt mai noi decât cele pe care le înlocuiți).

[2p] (b) Modificați permisiunile fișierelor din directorul *"destination_directory"* astfel încât conținutul lor să nu mai poată fi modificat de niciun utilizator. Trebuie în continuare să poată fi create fișiere în directoare.

[2p] (c) Adăugați numele tuturor fișierelor care diferă în *"destination_directory"* față de *"source_directory"* în urma comenzilor, în fișierul *reports.txt* din *"destination_directory"*, folosind un one-liner.

[4p] (d) Ascundeți toate fișierele cu extensia *.txt* din întreaga ierarhie *"destination_directory"*, folosind un one-liner.



[10p] Proba 3

[2p] (a) Afișați dimensiunea totală a întregii ierarhii a directorului `/var/log` în format human-readable.

[3p] (b) Creați un fișier numit `acad_out.txt` care să conțină linia "Ac@dnet\$" de 2024 ori.

[3p] (c) Afișați numărul de apariții ale caracterului "c" din fișierul `/etc/shadow`.

[2p] (d) Afișați pe ecran informații despre memoria RAM și memoria swap în KB.

[10p] Proba 4

[1p] (a) Creați utilizatorul `lambda`.

[2p] (b) Schimbați shell-ul utilizatorului `lambda` în `zsh`.

[3p] (c) Ca utilizatorul `lambda`, creați un fișier numit "`hot_potato`" și schimbați permisiunile sale astfel încât utilizatorul care îl deține să aibă drepturi complete asupra sa, grupul care îl deține să poată scrie și restul utilizatorilor să îl poată executa.

[4p] (d) Ca utilizatorul `lambda`, schimbați utilizatorul care deține fișierului "`hot_potato`" cu `root`.

[10p] Proba 5

[1p] (a) Afișați toate procesele de pe sistem împreună cu PID-ul lor, sub formă arborescentă.

[3p] (b) Afișați pentru fiecare proces al utilizatorului `root`, următoarele atribute: PID-ul, PPID-ul, utilizatorul ce le-a creat, și comanda cu care a fost pornit.

[2p] (c) Afișați toate PID-urile al cărui PPID-ul este egal cu 1.

[1p] (d) Afișați PID-ul shell-ului curent.

[3p] (e) Sortați în ordinea consumului de memorie procesele utilizatorului `root`. Afișați primele 5 procese.

[10p] Proba 6 (veți lucra în repository în directorul emails)

[6p] (a) În fișierul `company_emails.txt` din repo-ul clonat există mai multe șiruri de caractere pe câte un rând fiecare. Filtrați folosind o expresie regulată adresele de email valide din fișier. Afișați de asemenea numărul de adrese valide. Pentru simplitate, considerăm că o adresă validă este de forma "`username@domain.top-level-domain`", unde: username-ul poate conține litere mici, mari, cifre, `'` și `_` și începe cu o literă; domeniul poate conține litere mici și mari; domeniul top-level este o grupare de 2-4 litere mici. Exemplu de adresă email validă: "`John_Doe.32@hotmail.org`".

[1p] (b) Filtrați adresele de email valide care se termină în `.com`, și redirectați output-ul acestei comenzi într-un fișier `valid_emails.txt`.

[3p] (c) Pentru adresele de email salvate în fișierul `vaild_emails.txt`, înlocuiți domeniul adreselor cu "`christmasparty`" (nu și top-level-domain) și redirectați output-ul acestei comenzi în fișierul `christmas_emails.txt`.



[10p] Proba 7

[3p] (a) Scrieți într-un fișier toate numele fișierelor (doar numele, nu și calea) cu extensia `.log` din întreaga ierarhie a directorului `/var/log`.

[3p] (b) Creați o arhivă de tip `tar`, care să conțină toate fișiere modificate în ultimele 10 minute din întreaga ierarhie `/var/log`.

[2p] (c) Actualizați arhiva creată la subpunctul precedent cu fișierele modificate în ultimele 5 minute din întreaga ierarhie `/var/log`.

[2p] (d) Afișați fără a dezarhiva conținutul arhivei create.

[10p] Proba 8

[2p] (a) Configurați numele și mail-ul vostru pentru a folosi git în terminal.

[2p] (b) Creați directorul "Elevi". Inițializați directorul ca un repository de git.

[1p] (c) În interiorul directorului creați un fișier numit *Haralambie*, în care să scrieți "Romana: 8,5 | Matematica: 9,5 | Informatica: 10".

[5p] (d) Faceți un commit cu fișierul creat, cu mesajul "Elev adaugat" și afișați istoricul commit-urilor.

[10p] Proba 9

[3p] (a) Determinați numărul de cache-uri L1, L2 și L3 disponibile.

[3p] (b) Determinați dacă procesorul are sau nu instrucțiuni speciale pentru accelerarea criptării AES.

[2p] (c) Determinați numărul de plăci video prezente pe sistem.

[2p] (d) Determinați driver-ul folosit de placa de rețea.

[10p] Proba 10 (veți lucra în repository în directorul make)

[5p] (a) Folosind fișierele *secret.o*, *second_secret.o* și *main.c*, creați un Makefile care să conțină o regulă de build pentru compilarea unui executabil *main* ce poate fi executat și să afișeze următorul output:

Sum: 12

Dif: -2

Product: -24

[2p] (b) Adăugați o regula `run` la Makefile pentru executarea binarului obținut la subpunctul precedent. Dacă binarul nu există acesta va fi creat în același mod ca la subpunctul precedent (binarul va fi creat **doar** dacă el nu există deja).

[3p] (c) Creați o regulă de `clean` care să șteargă executabilul *main*. Pentru a rezolva acest exercițiu, **nu** aveți voie să faceți nicio modificare fișierului **clean** existent deja în arhivă (el nu poate fi șters, redenumit, etc.).