



Olimpiada de Informatică Aplicată AcadNet 2024

Etapă Națională

Secțiunea calculatoare, clasele 11 - 12

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 (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.).

[10p] Proba 2

[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 3 [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 4 (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 5

[2p] (a) Determinați numărul de cache-uri L1, L2 și L3 disponibile.

[2p] (b) Determinați dacă procesorul are sau nu instrucțiuni speciale pentru accelerarea criptării AES.

[1p] (c) Determinați numărul de plăci video prezente pe sistem.

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

[2p] (e) Determinați ce tip de hypervisor este folosit pentru mașina virtuală pe care vă aflați.

[10p] Proba 6

[4p] (a) Realizați un script care să genereze la infinit, la interval de 3 secunde un text random în terminal cu o culoare random din primele 8 culori ANSI.

[2p] (b) Faceți modificările necesare în așa fel încât la introducerea unui semnal de tip SIGINT, script-ul să nu își oprească execuția, ci să afișeze mesajul *“You are not allowed to stop the script by a SIGINT signal”*.

[3p] (c) Realizați o variantă îmbunătățită a script-ului astfel încât la introducerea de la tastatură a oricărui input, programul trebuie să intre într-o stare de așteptare. Altfel spus, script-ul trebuie să ruleze în continuare, dar să nu mai afișeze texte random cu culori random. Deblocarea trebuie făcută prin trimiterea de EOF de la tastatură, moment în care script-ul va începe din nou să afișeze texte random cu culori random în terminal.

[1p] (d) Faceți modificările necesare în așa fel încât script-ul să poată fi rulat din orice director în linia de comandă.



[10p] Proba 7 (veți lucra în repository în directorul emails)

[6p] (a) În fișierul *company_emails.txt* 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 8

[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*" în *root*.

[10p] Proba 9

[2p] (a) Afișați adresele IP ale tuturor nodurilor prin care treceți pentru a ajunge la *dns.google*.

[2p] (b) Afișați adresa IPv6 asociată site-ului *wikipedia.com*.

[2p] (c) Afișați toate porturile TCP deschise ale sistemului.

[2p] (d) Afișați **doar** ruta implicită de rețea.

[2p] (e) Aflați adresa IP a *stackoverflow.com* folosind un query DNS la 8.8.8.8

[10p] Proba 10

[6p] (a) Realizați un script în *bash* care să creeze *X* fișiere în directorul *Y* (dacă nu există trebuie să fie creat), fiecare având dimensiunea de *Z* în kilobytes. *X*, *Y* și *Z* vor fi primite ca argumente ale script-ului.

[4p] (b) Creați un script în *bash* care să verifice conectivitatea la un IP primit ca argument și să scrie într-un fișier numit *ye\$\$* stringul "*DA*" în cazul în care se efectuează cu succes și stringul "*NU*" în caz contrar.