**Documentatie Probleme**

**Setul 1**

**1.** Se cere un fisier de comenzi care are in linia de comanda un nume de fisier f si un numar n. Se cere crearea fisierului f prin concatenarea fisierelor f.1, f.2, ..., f.n. Daca unul din aceste fisere nu exista el va trebui creat si va trebui sa contina textul "Eroare: acest fisier nu a existat inainte de executia comenzii".

Aspecte importante din rezolvare: verificarea parametrului dat daca este fisier si adaugare la finalul fisierului initial folosind operatorul “>> “.

**2.** Pentru fiecare fisier ASCII dintr-un director dat ca parametru si din toate subdirectoarele lui, se vor afisa primele n linii (n dat ca parametru).

Aspecte importante din rezolvare: preluarea din totae subdirectoarele directorului a tuturor fisierelor $(find *cale* –type f), afisarea a N linii utilizand comanda $head -N *file .*

**3**. Sa se creeze doua fisiere, unul care va contine lista tuturor fisierelor dintr-un director si din subdirectoarele acestuia, iar al doilea lista tuturor subdirectoarelor. Fisierele vor fi ordonate dupa dimensiune, iar directoarele alfabetic.

Aspecte importante din rezolvare: folosirea comenzi find cu redirectionarea pipe-ului catre comanda sort apoi catre scrierea in fisier.

**4.** Un fisier de comenzi care ordoneaza dupa numarul de cuvinte continute, fisierele date ca si parametrii.

Aspecte importante din rezolvare: prelucrarea tuturori parametrilor dati utilizand comanda “for i in $\*” si afisarea listei sortate utilizand instructiunea for redirectionata catre comanda sort.

**5.** Un fisier de comenzi care pentru fiecare user dat ca si parametru creeaza cate un un fisier cu numele userului respectiv care va contine (printre altele) numele complet al utilizatorului.

Aspecte importante din rezolvare: preluarea fiecarui parametru folosind comanda “while test $# -ne 0 … shift”.

**6.** Fie cateva fisiere care contin nume de utilizatori. Se cere un fisier de comenzi care pentru fiecare nume de fisier (ce contine useri) primit ca si parametru trimite cate un mail userilor din fisierul respective.

Aspecte importante din rezolvare: instalarea packetului ssmtp, utilizarea acestuia pentru a trimite un mail si utilizarea exprsiilor regulata pentru a gasi intr.-un fisier adresele de mail ("\b[A-Za-z0-9.\_%+-][+@[A-Za-z0-9.-]+\.[A-Za-z]{2,6}](mailto:+@[A-Za-z0-9.-]+\.%5bA-Za-z%5d%7b2,6%7d)").

**7.**Un fisiere de comenzi care afiseaza numarul de fisiere din fiecare director dat ca si parametru si afiseaza spatiul ocupat pe disc de directoarele respective.

Aspecte importante din rezolvare: folosirea comenzó du pentru a obtine spatiul ocupat de director.

**8.**Un fisier de comenzi care pentru un tip de fisier dat ca si parametru (ex \*.c sau a\* sau \*~) sterge fisierele respective dintr-un director dat ca si parametru si afiseaza dimensiunea directorului inainte si dupa stergere

Aspecte importante din rezolvare: folosirea proprietatii –delete a comenzii find pentru a sterge fisierele gasite.

**9.**Un fisier de comenzi care din 10 in 10 secunde ( folosind un ciclu while) afiseaza incarcarea sistemului (comanda uptime) si spatiul ocupat pe disc.

Aspecte importante din rezolvare: repetarea din 10 in 10 secunde folosind o bucla infinita impreuna cu comanda sleep, afisarea incarcarii sistemului folosind comanda uptime si afisarea spatiului pe disc cu ajutorul comenzii df.

**10.**Un fisier de comenzi care din 10 in 10 secunde afiseaza pe ecran utilizatorii din retea (in ordine alfabetica) si statiile de pe care sunt conectati.

Aspecte importante din rezolvare folosirea comenzii id care afiseaza date complete despre utilizator.

**11**.Se da un fisier (sau mai multe) cu utilizatori. Sa se scrie un fisier de comenzi care din 20 in 20 de secunde trimite un mesaj (aflat intr-un fisier primit ca si parametru) utilizatorilor din fisierul (sau fisierele) respective.

Aspecte importante din rezolvare: trimiterea de mesaje catre utilizatori folosind comanda write.

**12.** Se da un fisier cu utilizatori. Sa se scrie un fisier de comenzi care, luand pe rand utilizatorii din fisier, la interval de 30 secunde le trimite cate un mail.(continutul mailului se afla intr-un fisier transmis ca si parametru).

Aspecte importante din rezolvare: rescrierea pentru fiecare utilizator citit a fisierului care continea mesajul, deoarece acesta trebuie sa contina si destinatarul.

**13.** Sa se scrie un fisier de comenzi care are ca parametri perechi formate dintr-un nume de fisier si un numar k. Pentru fiecare pereche, se vor afisa numele fisierului numarul k si apoi primele k linii din fisier.

Aspecte importante din rezolvare: preluarea parametrilor pereche folosind instructiunea while si shift.

**Setul2**

**1.** Se da un program C. Sa se determine, pentru fiecare functie definita in program, numarul locurilor unde este apelata, afisandu-se si numerele de linii din program unde apar aceste apeluri.

Aspecte importante din rezolvare: utilizarea expresiilor regulate impreuna cu comanda grep pentru a gasi patternul de declarare de functie ("\s\*[a-Z]\*\s\*(\s\*.\*\s\*)\s\*{").

**2.** Se da un fisier text. Sa se creeze un director care sa contina 26 de fisiere dictionar, astfel: primul fisier contine, ordonate alfabetic, cate unul pe linie, toate cuvintele din fisierul text care incep cu litera 'A', al doilea cu litera 'B' s.a.m.d.

Aspecte importante din rezolvare: preluarea fiecarui cuvant fdi fisierul dat cu ajutorul instructiunii “for w in `cat "$1"`”, si scrierea acestuia in fisierul corespunzator “echo "$w" >> "$name/${w:0:1}"”.

**3.** Se da un fisier cu studentii grupei ordonati in functie de media notelor primite la sesiunea din iarna. Dandu-se pentru fiecare student notele la cele 5 examene din vara (in fisierul 'notevara') sa se actualizeze fisierul initial astfel incat sa contina clasamentul real pe anul universitar respectiv.

Aspecte importante din rezolvare: innainte de a prelua notele din fisier am le-am sortat pentru a fi sigur ca acestea apartin aceluaias student.

**4.** Pentru o lista de nume de fisiere data si pentru un director dat, sa se afiseze pentru fiecare nume de fisier toate subdirectoarele in care acesta apare. Sa se afiseze subdirectoarele in ordinea descrescatoare a datei crearii fisierului.

Aspecte importante din rezolvare: folosirea comenzii find cu ajutorul proprietatii –printf.

**5.** Sa se scrie un program Shell care pentru o extensie data, muta intr-un director dat toate fisierele cu acea extensie din toate subdirectoarele directorului curent.

Aspecte importante din rezolvare: utilizarea proprietatii –exec (similara functiei fork() din C) pentru a muta fisierele rezultate in urma comenzii find.

**6.** Sa se scrie un program Shell care primeste ca parametri doua nume de directoare si muta ramura desemnata de cel de-al doilea director ca subramura a primului, adaugand extensia .unu la fiecare fisier.

Aspecte importante din rezolvare: folosire instructiunii for impreuna cu comanda find pentru a lua fiecare rezulat al cautarii.

**7.** Sa se creeze doua fisiere care sa contina fiecare cate o lista a tuturor fisierelor si subdirectoarelor directorului dat ca parametru (daca nu se da parametru atunci a directorului curent), lista ordonata crescator in functie de:

data crearii, in primul fisier

alfabetic, in cel de-al doilea.

In dreptul numelui se va indica daca este este fisier sau director.

Aspecte importante din rezolvare: redirectionarea pipe-ului comenzii ls catra awk apoi mai departe carte sort iar in final catre scrierea in fisier.

**15.** Sa se scrie un program Shell care, pentru fiecare fisier din ramura desemnata de un director dat ca parametru, realizeaza urmatoarele actiuni:

daca este fisier cu extensia c, il compileaza

daca este fisier executabil, il lanseaza in executie

in caz contrar, daca este fiser text ii afiseaza continutul.

Aspecte importante din rezolvare: preluarea fiecarui fisier din directorul dat folosind un array si instructiunea for si preluarea extensiei unui fisier folosind comanda "${ech##\*.}".

**16.** Sa se scrie un program Shell care sa lanseze comanda kill pentru anumite procese. Programul va afisa mai intii lista tutror proceselor, va cere utilizatorului sa indice PID-ul unui proces care sa fie oprit, dupa care va relua operatia (afisarea tuturor proceselor si cererea facuta utilizatorului).

Aspecte importante din rezolvare: rescreierea la inceputul programului a blocului de instructiuni din interiorul instructiunii while (deoarece nu am gasit instructiunea do.. while) si folosirea comenzii kill pentru a ori un proces.

**Setul 3**

**1.** Sa se afiseze ora curenta sub forma

ora xx, xx minute, xx secunde

Aspecte importante din rezolvare: preluarea orei in format convenabil , si afisarea acesteai in formatul cerut.

**2.** Sa se afiseze procesele ce au consumat mai mult de 10s de unitate centrala.

Aspecte importante din rezolvare: folosirea condtiilor in comanda awk.

**3.** Sa se listeze cele mai mari 10 fisiere dintr-un director dat ca parametru si din subdirectoarele sale, in ordine descrescatoare a lungimii

Aspecte importante din rezolvare: redirectionarea pipe-ului catre comanda awk unde folosesc parametrul FNR –current record number.

**4.** Se da, in fisierul studenti.lst lista studentilor in formatul:

Nume Prenume nume-utilizator grupa

(coloanele sunt separate prin blancuri; numele compuse sunt separate prin '-'; exemplu: Popescu Ana-Maria pa12345 221 . Dandu-se o lista de grupe, sa se trimita fiecarui student cate un mesaj (mesajul se da intr-un fisier al carui nume este de asemenea dat ca parametru).

Aspecte importante din rezolvare: scrierea mesajului intr-o variabila pentru al trimite studentului folosind comanda echo si write.

**5.** Pornind de la un director, sa se afiseze toate fisierele, din el si din subdirectoarele sale, care au proprietar diferit de proprietarul directorului in care se gasesc.

Aspecte importante din rezolvare: preluarea creatorului folerului folosing comanda ls redirectionata catre grep si apoi awk

**6.** Sa se afiseze liniile din fisiere care contin un acelasi cuvint aflat in pozitii consecutive. Pentru liniile respec- tive sa se afiseze si numarul liniei (in cadrul fisierului din care face parte).

Aspecte importante din rezolvare: folosirea instructiuni repetitive for in interiorul comenzii awk.

**7.** Se da un fisier cu utilizatori in formatul urmator: Nume Prenume nume-utilizator zi luna an unde zi, luna si an reprezinta data nasterii acelei persoane Se cere sa se trimita fiecarui utilizato care isi aniverseaza data nasterii in ziua curenta un mesaj de forma

La multi ani, Prenume Nume !

Aspecte importante din rezolvare: folosirea variabilelor in interiorul comenzii awk pentru a cauta utilizatorul care isi serbeaza ziua.

**8.** Se da un fisier cu datele privind un examen, in formatul urmator

Nume Prenume nume-utilizator nota1 nota2 ... nota\_n

(numarul de note este acelasi pentru toti studentii) Se cere construirea unui fisier in format Nume Prenume nume-utilizator media media fiind media aritmetica a notelor

Aspecte importante din rezolvare: folosirea instructiuni repetitive for in interiorul comenzii awk.

**9.** Pentru fisierul de iesire de la problema 8, sa se trimita fiecarui student un mesaj de forma:

Catre Prenume Nume,

Va informam ca la examenul de sisteme de operare ati obtinut nota

media.

unde bineinteles Prenume, Nume si media se vor inlocui cu valorile adecvate.

Aspecte importante din rezolvare:fiecare coloana din fisier este preluata cu ajutorul comenzii awk si apoi pentru fiecare user estre trimis un mesaj cu ajutorul instructiunii write.

**10.** Sa se afiseze numarul maxim de linii consecutive care coincid dintr-un acelasi fisier si continutul liniei respective precum si numele fisierului care o contine.

Aspecte importante din rezolvare: cu ajutorul comenzii awk initiem variabilele apoi verificam conditia si in functie de ea se executa alt bloc de instructiuni, iar in final este afisat rezultatul.

**11.** Pentru toate fisierele cu sufixul .c, .cc sau .h dintr-un director si din subdirectoarele sale sa se calculeze si sa se afiseze numarul mediu de cuvinte pe linie, apoi sa se calculeze numarul mediu pentru toate fisierele (media generala).

Aspecte importante din rezolvare: cautarea fisierelor corespunzatoare folosint proprietatea –name a comenzii find.

**12.** Sa se inlocuiasca intr-un numar oarecare de fisiere orice apari- tie a unui anumit cuvint cu un alt cuvint; cuvintele respective vor fi date in linia de comanda awk ca si parametri).

Aspecte importante din rezolvare: folosirea functiei “gsub(a,b)” a comenzii awk pentru a inlocui cuvantul cautat.

**Setul 4**

**1.** Sa se scrie un program C care interclaseaza doua liste de nume, primite ca parametru, aflate in doua fisiere diferite. Rezultatul va fi tot un fisier.

Aspecte importante din rezolvare: algoritmul este acelasi ca cel pentru interclasare a 2 vectori (https://tutoriale-pe.net/interclasarea-a-doi-vectori-in-c/) doar adaptat pentru fisiere.

**2.** Sa se scrie un prgogram C care parcurge recursiv un director si raporteaza intr-un fisier continutul acestuia (spre exemplu: directorul contine 9 subfoldere, 4 fisiere c, 5 programe executabile, etc).

Aspecte importante din rezolvare: folosirea functiei readdir() din biblioteca dirent.h pentru a citi urmatorul fisier din director.

**3.** Sa se creeze un program C care prezinta operatii de adaugare, stergere pe un fisier cu tip.

Aspecte importante din rezolvare: folosirea functiei fseek() pentru repozitionarea cursorului.

**4.** Sa se sorteze crescator un sir de numere citit dintr-un fisier.

Aspecte importante din rezolvare: citim din fisier fiecare numar si incrementam nr total de valori pentru a afla dimensiunea, apoi se sorteaza un vector de intregi iar la final acesta este afisat.

**5.** Sa se implementeze o comanda de tipul wc.

Aspecte importante din rezolvare: se deschide fisierul trimis ca parametru in functia main, pentru fiecare caracter nr\_c creste, pentru fiecare spatiu creste nr\_s si nr\_w, pentru fiecare “\n” creste nr\_w iar la final se afiseza toate.

**Setul 5**

**1.** Se da un numar natural N. Se cere sa se genereze toate numerele prime pana la N, in modul urmator: Se creaza N procese: P[1]...P[N]. Procesul P[1] va genera lista numerelor naturale de la 2 la N. Procesul P[k], 2<=k<N, va citi lista de la P[k-1], va elimina multiplii lui k si va trimite rezultatul lui P[k+1]. In fine, P[N] va scrie lista finala.

Aspecte importante din rezolvare: initial N=10 dar daca programul este apelat cu parametru acesta ia valoarea parametrului cu ajutorul functiei atoi(), avem un pointer \*glob\_var catre o adresa de memorie comuna proceselor pentru asta am folosit comanda mmap(), intr.-un for se creeaza N copii pentru procesul parinte, fiecare copil elimina multiplii lui si modifica glob\_var si la final este afisat glob\_var iar zona de memorie este dealocata.

**5.** Se dau doua siruri de numere reale, stocate in doua fisiere binare. Sa se calculeze suma vectorilor (suma element cu element) si sa se scrie intr-un al treilea fisier. Se va folosi cate un proces pentru fiecare element.

Aspecte importante din rezolvare: Asemanator problemei anterioare cu diferenta ca un se vor mai creea N copii ci doar unul care calculeaza suma primului fisier iar parintele adauja la suma anterioara elementele din al doilea fisier, fisierele folosite sunt scrise si citite in binar folosind la functia fopen() al doilea parametru cu “r/wb”.

**6.** Se da un sir de numere. Sa se calculeze suma lor prin 'divide et impera': se va crea un arbore binar de procese, fiecare proces va calcula suma a doua numere.

Aspecte importante din rezolvare: acelasi pointer catre o zona de memorie comuna, o functie care pe copilul procesului trimite jumatate din lungimea vectorului trimis ca parametru iar pe parinte ceallat jumatate, ea se apeleaza recurziv pana cand lungimea vectorului trimis ca parametru este 1.

**7.** Se da un sir de numere. Sa se calculeze cmmdc al lor prin 'divide et impera': se va crea un arbore binar de procese, fiecare proces va calcula cmmdc a doua numere.

Aspecte importante din rezolvare: Asemeni problemai anterioare daor ca in loc de suma este calculat cmmdc-ul.

**Setul 6**

**1.** Sa se scrie un program C care afiseaza adresele IP ale calculatorului.

Aspecte importante din rezolvare: gethostname() intoarce hostname-ul in variabila buffer, preluam numele hostului folosind functia gethostbyname() iar apoi inet\_ntoa() returneaza IP-ul.