

Sisteme de operare

Laborator 1

Interpretorul de comenzi

1. Creati o structura de directoare pe care sa o folositi pentru programele de la laborator, dupa cum urmeaza:

```
cd
pwd
mkdir -p SO/laborator/lab1
cd SO/laborator/lab1
pwd
mkdir doc src bin
ls -l
```

Creati apoi un fisier `.myrc` in directorul `SO` care sa creeze alias-uri pt comenzile `rm` si `ls` si sa extinda variabila de mediu `PATH` dupa cum urmeaza:

```
cd ../../
pwd
echo alias rm='rm -i\' >> .myrc
echo alias l='ls -l\' >> .myrc
ls -la
cat .myrc
env | grep PATH
env | grep HOME
echo export PATH=$PATH:$HOME/SO/laborator/lab1/bin >> .myrc
cat .myrc
source .myrc
rm .myrc

cd -
pwd
l
```

Pentru a intelege mai bine ce face fiecare comanda, cititi paginile de manual (eg, *man mkdir*). In subdirectorul *doc* veti salva documentatia de laborator, in subdirectorul *src* veti crea programele sursa C (folosind editorul text preferat), iar in subdirectorul *bin* veti pune programele executabile rezultate ca urmare a compilarii programelor sursa C.

Compilarea programelor se va face cu urmatoarea comanda:

```
cd ~/SO/laborator/lab1/src
pwd
gcc -o ../bin/fisier_executabil fisier_sursa.c
```

unde `fisier_executabil` si `fisier_sursa` sunt nume de fisiere pe care le alegeti dupa dorinta fiecaruia.

Apoi puteti executa programul executabil rezultat, din orice director, cu comanda

```
fisier_executabil
```

Aflati de ce este posibil asa ceva, cata vreme *fișier_executabil* se afla in subdirectorul *bin* iar la momentul executiei nu se specifica calea catre fisierul executabil.

2. Scrieti un program C care primeste ca parametru in linia de comanda un nume de utilizator si afiseaza informatii despre ele cum ar fi: user ID-ul si group ID-ul, home directory, shell.

Indicatie: Folositi functia *getpwnam* (*man getpwnam*).

3. Scrieti un program C care copiaza in bucla informatia citita de la standard input (file descriptor 0) la standard output (file descriptor 1). Cum puteti folosi acest program pentru a simula efectul comenzii *cp*, care copiaza un fisier sursa intr-un fisier destinatie?

4. Scrieti un program C care afiseaza continutul unui director furnizat ca parametru al programului, simuland functionarea simpla, neparametrizata, a comenzii *ls*.

Indicatie: Folositi functiile *opendir/readdir/closedir*.

5. Scrieti un program C care afiseaza PID-ul, UID-ul si GID-ul procesului curent.

Indicatie: Folositi functiile *getpid, getuid, getgid*.

6. Scrieti un program C care emuleaza operatia *getenv* (*man getenv*). Programul primeste ca parametru numele unei variabile de mediu (eg, HOME) si tipareste pe ecran continutul variabilei respective.

Indicatie: Parcurgeti vectorul de string-uri *envp* (al treilea argument de apel al functiei *main*) in cautarea numelui variabilei de mediu. Pentru comparatia de string-uri folositi *strcmp/strncmp*.

7. Exersati comezile shell din prezentarea laboratorului, deopotriva comenzi interne si externe. Pentru a afla calea absoluta a unui executabil folositi comanda

which cmd

Pentru a cauta un fisier oarecare dupa un substring din numele sau folositi comanda

locate string

Daca locate nu functioneaza, instalati comanda folosind package managerul sistemului Linux pe care il folositi si apoi apelati *updatedb* in calitate de root.