

Laborator 8 – Variabile de Environment

Ce este o variabila de environment ?

O variabila de mediu (environment) este o variabila care afecteaza modul cum un proces ruleaza. Variabilele de mediu nu reprezinta variabile normale din program, ci niste variabile mentinute de sistem in zona de adresare virtuala a procesului. O variabila de mediu este un string de forma:

NUME_VARIABILA=VALOARE

In functie de aceste variabile, un proces are o anumita manifestare. Aceste variabile se mostenesc de la procesul tata la procesul fiu. Fiul va avea o copie a mediului si va putea sa il modifice la randul sau adaugand sau stergand diverse variabile. Bineinteles ca schimbarile pe care le va face un proces mediului sau nu se vor vedea si asupra tatalui, deoarece el are doar o copie.

Cum si **shell-ul** este un proces, si el are diverse variabile de mediu ce ii influenteaza executia. Pentru a afisa toate variabilele de mediu din shell se foloseste comanda :

\$ printenv

O sa obtinem o lista ce va avea pe fiecare linie un tuplu cu formatul mentionat mai sus. Pentru a afisa valoarea unei anumite variabile se foloseste comanda:

\$ echo \$NUME_VARIABILA

Nu este obligatoriu ca numele unei variabile sa fie cu litere mari, dar asa este good-practice. Printre variabilele de mediu ale shell-ului se numara

PATH – ce contine caile catre binarele de sistem, fiecare cale este separata prin ':'

HOME – ce contine calea catre directorul home al utilizatorului logat

PWD – ce contine directorul curent de lucru al shell-ului

Pentru a seta o variabila de mediu in shell se foloseste comanda:

\$ export NUME_VARIABILA=valoare

Problema este ca aceasta variabila va fi setata doar pentru instanta curenta a bash-ului. Daca dorim sa adaugam o variabila de mediu persistenta pentru orice instanta de bash si pentru orice instanta a sistemului trebuie modificat fisierul de configurare ~/.**bashrc** si adaugat o noua linie de forma:

NUME_VARIABILA=valoare

Studiu de caz

Dupa cum am vazut pana acum, atunci cand dorim sa lansam programul rezultat in urma compilarii unui program C trebuie sa introducem ceva de genul:

\$./a.out

'./' trebuie pus deoarece fara el bash-ul ar cauta in caile din variabila de mediu **PATH**, si nu il va gasi. Avem doua optiuni daca dorim sa nu mai mentionam './' :

- punem executabilul a.out in unul din directoarele specificate in **PATH**, lucru care de obicei necesita drepturi de administrator (**root**)
- modificam **PATH-ul** astfel incat sa contina si directorul unde se afla curent a.out. O posibila comanda ar fi:

\$ export PATH=\$PATH:\$PWD

Intr-un program C, variabilele de mediu pot sa fie obtinute in doua moduri:

1. prin variabila globala **environ**
2. prin al treilea parametru la functia **main**.

Variabila globala environ este precum **argv**. Un vector de siruri de caractere. Fiecare sir de caractere este de forma **VARIABILA=valoare**. Ultimul element al lui environ este pointerul **NULL**.

In continuare vom prezenta doua programe C ce afiseaza variabilele de mediu ale unui proces in cele doua moduri precizate.

Prin variabila globala environ:

```
#include<stdio.h>

extern char** environ;
int main()
{
    char** p;

    for ( p = environ ; *p ; p++ )
        puts(*p);

    return 0;
}
```

Prin al treilea argument la functia main:

```
#include<stdio.h>

int main(int argc, char* argv[], char* environ[])
{
    char** p;

    for ( p = environ ; *p ; p++ )
        puts(*p);

    return 0;
}
```

Observam ca nu avem cum sa stim dinainte numarul de elemente din vectorul de variabile de mediu, cum stim la vectorul de argumente din linia de comanda prin **argc**. Deci trebuie sa ne

raportam la pointerul **NULL** ce marcheaza finalul acestui vector.

Pentru a manipula variabilele de mediu avem la dispozitie cateva functii.

Astfel, pentru a afla valoarea unei anumite variabile de mediu putem sa folosim functia:

```
#include <stdlib.h>
```

```
char *getenv(const char * name );
```

Aceasta functie intoarce **NULL** in caz de esec iar in caz de succes returneaza pointerul catre valoarea variabilei de enviroment transmisa ca parametru.

Pentru a adauga o noua variabila de mediu putem folosi functia:

```
#include <stdlib.h>
```

```
int setenv(const char * name , const char * value , int overwrite );
```

Aceasta functie returneaza 0 in caz de succes si -1 in caz de esec. Primul parametru reprezinta numele variabilei pe care dorim sa o adaugam. Al doilea parametru reprezinta valoarea variabilei pe care dorim sa o adaugam. Nu trebuie sa adaugam si semnul '=' acesta va fi adaugat de catre functie. Al treilea parametru este luat in considerare doar daca variabila cu numele **name** exista deja. Daca exista si **overwrite** este 0 atunci nu se va produce nici o modificare. Daca exista si **overwrite** este mai mare ca 0 atunci variabila va fi inlocuita cu noua valoare a variabilei.

Pentru a sterge o variabila cu un anumit nume vom folosi functia:

```
#include <stdlib.h>
```

```
int unsetenv(const char * name );
```

Functia va intoarce 0 in caz de succes si -1 in caz de esec. Argumentul reprezinta numele variabilei pe care vrem sa o inlaturam din lista de variabile.

Pentru a sterge intreaga lista de variabile de mediu putem sa folosim functia:

```
#include <stdlib.h>
```

```
int clearenv();
```

Acesta functie va intoarce 0 daca a sters cu succes intreaga lista de variabile si o valoare mai mare ca 0 in cazul in care a esuat.

Exercitii

1. Scrieti un program care afiseaza valoarea variabilei de environment **TERM**, apoi o asigneaza la valoarea "vt52", apoi genereaza un proces fiu (cu fork) care afisaza valoarea lui **TERM** mostenita.
2. Scrieti un program care sa verifice daca schimbandu-si valoarea variabilei de environment **PWD**, functia **getcwd** returneaza acelasi director curent.

3. Scrieti un program care va sterge continutul variabilei de mediu **PATH** dupa care va face un **execv** la **/bin/bash**. Dupa lansarea programul introduceti comanda **ls**. Ce observati ?
4. Scrieti un program care inlocuieste toate variabilele de mediu cu variabilele de mediu primate ca argument in linia de comanda. Argumentele din linia de comanda sunt de tip **VARIABILA=valoare**.
5. Scrieti un program care isi concateneaza valoarea fiecarei variabile de environment de pe pozitie para la sfarsitul valorii variabilei pe pozitia impara anterioara. Programul isi va afisa environmentul inainte si dupa aceasta operatie.
6. Implementati singuri functile **setenv()**, **unsetenv()** si **getenv()**.