**Subiecte de pregătire către lucrarea de laborator 6**

**Explicaţi comenzile**

**Kill**- transmite un semnal procesului. Semnalul poate fi un număr sau un nume simbolic. În mod implicit (fără a specifica semnalul), se transmite semnalul de terminare a procesului. Identificarea procesului se petrece după PID. Utilizaţi comanda kill –l şi primiţi o listă a semnalelor de sistem disponibile în GNU/Linux.

**Killall**- funcționează în același mod ca și comanda kill, dar pentru identificarea procesului utilizează un nume simbolic în loc de PID;

**Pidof**- afişează ID-ul procesului a unui program în execuție;

Pgrep- determină PID-ul proceselor cu caracteristicile specificate (de exemplu, lansate de un anumit utilizator);

**Pkill**- trimite un semnal unui grup de procese cu caracteristicile specificate;

**Nice-**lansează procesul cu valoarea de prioritate specificată. Reducerea valorii (creșterea priorității execuției) poate fi inițiată numai de utilizatorul root;

**Renice-** modifică prioritate (valoarea) procesului lansat. Reducerea valorii (creșterea priorității execuției) poate fi inițiată numai de utilizatorul root;

**At-** comandă introdusă de la tastatură, care urmează să fie executată ulterior;

**Tail-** vă permite să depistaţi și afişaţi liniile la sfârșitul fișierului;

**Sleep-** introduce o pauză în execuţia script-ului;

**Cron-** un daemon care permite execuţia comenzilor la un moment de timp.

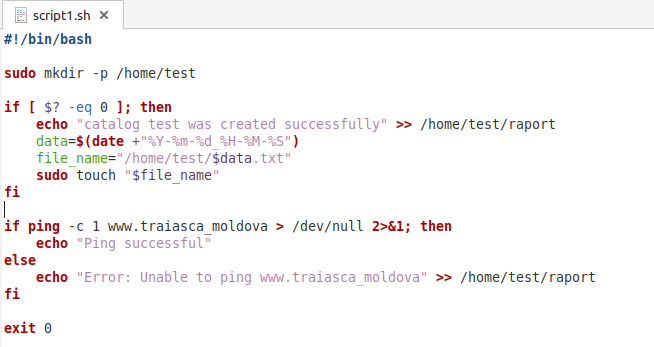
**Trap-** Pentru prelucrarea semnalelor în sh (bash).

Comanda are doi parametri: acțiunea (action) executată la recepționarea semnalului și semnalul pentru care va fi efectuată acțiunea specificată. În mod obișnuit, acțiunea este indicată prin apelarea funcției (cod de script) scrise anterior apelului. Cu comanda trap puteți să schimbați handler-ul pentru unele semnale de sistem (cu excepția celor ale căror interceptare este interzisă). În acest caz, handler-ul executat este indicat de argumentul action în comanda trap.

**Explicati notiunea de handler.**

Tratarea semnalelor se realizează prin asocierea unei funcții (handler) unui semnal. Funcția (rutină, handler) reprezintă o secvență de cod, care va fi apelată în momentul în care procesul recepționează semnalul respectiv. Semnalele de sistem au handler-ele sale, de obicei.

**1.**



**-p** asigură că toate directoarele intermediare necesare sunt create în cazul în care nu există.

**if [ $? -eq 0 ]; then ... fi:** Aceasta este o condiție care verifică rezultatul ultimei comenzi. **$?** reprezintă codul de ieșire al ultimei comenzi executate.

**data=$(date +"%Y-%m-%d\_%H-%M-%S"):** Acesta stochează data și ora curentă în variabila **data** într-un format specific (%Y-%m-%d\_%H-%M-%S).

**sudo touch "$file\_name"**: Acesta creează un fișier gol cu numele specificat anterior în directorul /home/test folosind comanda touch.

**-c 1** specifică să se trimită doar un pachet de ping. Rezultatul este redirecționat către /dev/null pentru a suprima output-ul, iar mesajul de eroare este redirecționat la fel către /dev/null.

**/dev/null** este un dispozitiv special în sistemele Unix/Linux care "arde" (discardă) orice date trimise către el. Prin redirecționarea ieșirii către /dev/null, efectiv se elimină orice afișare a rezultatelor în terminal.

**2>&1**: Aceasta redirecționează erorile standard (file descriptor 2, notat cu 2) către aceeași destinație ca și output-ul standard (file descriptor 1, notat cu 1). În acest caz, ambele sunt redirecționate către /dev/null. Acest lucru înseamnă că atât output-ul cât și erorile generate de comandă sunt eliminate și nu sunt afișate în terminal.

**2.**



**-n :** permite specificarea numarului de linii care sa fie afisate

**+$ :** va lua numarul liniei, care este numarul total de linii.

**3.**





**(\*/5):** Reprezintă minutele.

\* - orele

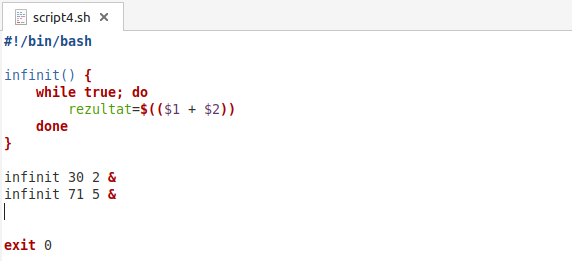
\* - ziua lunii

\* - oricare luna

2,4,6 – ziua saptamanii, 0 – duminica, 6 - sambata



**Ex4**



**&** - procesul se va rula in fundal.



**Cpulimit -p pid -l 20**