Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică şi Microelectronică

Departamentul Ingineria Software și Automatică

**RAPORT**

Lucrarea de laborator nr. 6

# La disciplina „ Sistemul de operare”

Tema: "Gestionarea proceselor în SO GNU/Linux"

A efectuat: st. gr. SI-211 S. Chirita

A verificat: V. Beșliu

Chișinău – 2023

Lucrarea de laborator nr. 6

#### Scopul lucrarii**: studierea comenzilor destinate gestionăriiproceselor, programarea timpului de start al proceselor,transferul și gestionarea datelor între procese.**

**Sarcină la laborator 6**

1) Creați și executați o singură dată script-ul (în acest script, nu puteți utiliza operatorul de condiție și operatorii de control a proprietăților și valorilor), care va încerca să creeze directorul test în directorul home. În cazul în care directorul va fi creat, script-ul va scrie în fișierul ~/raport un mesaj "catalog test was created successfully" şi va crea în directorul test un fișier numit Data\_Ora\_Lansarii\_Scriptului. Apoi, indiferent de rezultatele etapei anterioare, script-ul trebuie să interogheze, folosind comanda ping, adresa

www.traiasca\_moldova.md

și în cazul în care host-ul nueste disponibil, adăugați un mesaj de eroare în fișierul~/raport (se recomanda - >,>>, ~, date, ||, &&).

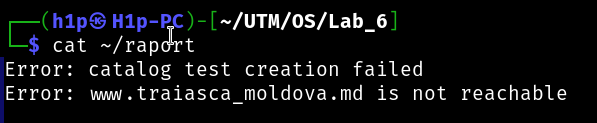
2) Modificați script-ul din 1) pentru încă o singură executare peste 2 minute. Controlați conținutul fișierului ~/raport și afișați liniile noi apărute ( se recomanda – at, tail).

3) Modificați script-ul din 1) ca să ruleze la fiecare 5 minute din oră, în zilele pare ale săptămânii.

4) Creați două procese de fundal care efectuează același ciclu infinit de calcul (de exemplu, înmulțirea a două numere). După lansarea proceselor, prevedeți posibilitatea de a utiliza consolele virtuale din care au fost lansate. Folosind comanda top, analizați procentul de utilizare a microprocesorului de către aceste procese. Rezervați primului proces lansat o rată de utilizare a resurselor microprocesorului nu mai mare de 20% (se recomanda – nice, kill, cpulimit).

5) Procesul "Generator" transmite informații procesului "Handler" utilizând un fișier ascuns. Procesul "Handler" trebuie să efectueze următoarele procesări asupra liniilor noi în acest fișier: dacă linia conține un singur caracter "+", procesul "Handler" comută modul în adunare și așteaptă introducerea datelor numerice. Dacășirul conține un singur caracter "\*", atunci procesul "Handler" comută modul în înmulțire și așteaptă introducerea datelor numerice. În cazul când linia conține un întreg, atunci procesul "Handler" execută operația curentă activă (modul activ curent) asupra valorii variabilei calculate (curente) și valoarea recent introdusă (de exemplu, adună sau înmulțește rezultatul calculat anterior cu numărul recent introdus). Când lansați script-ul, modul se va seta în adunare, iar variabila calculată – în 1. Dacă este primit QUIT, script-ul afișează un mesaj despre finalizare și finalizează activitatea. Dacă sunt primite alte valori, script-ul finalizează activitatea cu un mesaj de eroare - date de intrare eronate.

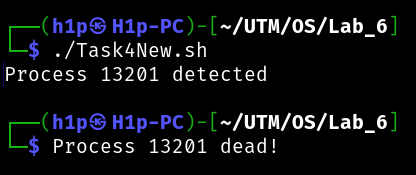
**1,2,3)**mkdir ~/test 2>/dev/null   
if [ $? -eq 0 ]; then   
   echo "catalog test was created successfully" > ~/raport   
   touch ~/test/Data\_Ora\_Lansarii\_Scriptului   
else   
   echo "Error: catalog test creation failed" > ~/raport   
fi   
  
ping -c 1 www.traiasca\_moldova.md >/dev/null 2>&1   
if [ $? -ne 0 ]; then   
   echo "Error: www.traiasca\_moldova.md is not reachable" >> ~/raport   
fi

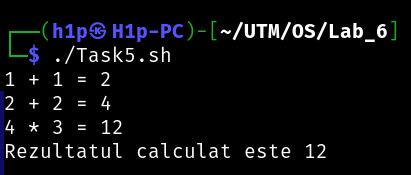
echo "bash ~/calea-catre-script/script.sh" | at now + 2 minutes  
day\_of\_week=$(date +%u)   
if [ $((day\_of\_week % 2)) -eq 0 ]; then   
   echo "\*/5 \* \* \* \* bash ~/calea-catre-script/script.sh" | crontab   
fi****

**4)**

#!/bin/bash   
# Functie pentru calcul ciclic infinit   
calculate\_infinite\_loop() {   
   while true; do   
       id    
   done   
}   
# Lansarea primului proces în fundal cu limitare de utilizare a CPU   
cpulimit -l 20 -- pidof -x "$(basename "$0")"   
#calculate\_infinite\_loop &   
# Lansarea celui de-al doilea proces în fundal fără limitare de utilizare a CPU   
#bash -c  "calculate\_infinite\_loop()" &   
# Așteptarea ca procesele să ruleze   
#wait

Varianta 2

nice -n 10 cpulimit -l 20 bash -c 'while true; do echo "Calculare proces 1"; done' &   
bash -c 'while true; do echo "Calculare proces 2"; done' &

#!/bin/bash   
mode="adunare"   
calculate=1   
if [ ! -e .generator\_file ]; then   
   echo "Fișierul ascuns .generator\_file nu există. Porniți Generatorul."   
   exit 1   
fi   
while read -r line || [[ -n "$line" ]]; do   
   case $line in   
       "+")   
           mode="adunare"   
           ;;   
       "\*")   
           mode="inmultire"   
           ;;   
       [0-9]\*)   
           if [ "$mode" == "adunare" ]; then   
               var=$calculate   
               calculate=$((calculate + line))   
               echo "$var + $line = $calculate"   
           elif [ "$mode" == "inmultire" ]; then   
               var=$calculate   
               calculate=$((calculate \* line))   
               echo "$var \* $line = $calculate"   
           fi   
           ;;   
       "QUIT")   
           echo "Finalizare..."   
           exit 0   
           ;;   
       \*)   
           echo "Eroare: Date de intrare eronate - \"$line\""   
           exit 1   
           ;;   
   esac   
done < .generator\_file   
echo "Rezultatul calculat este $calculate"  
Concluzie: Studiul comenzilor destinate gestionării proceselor, programării timpului de start al acestora, precum și transferul și gestionarea datelor între procese constituie aspecte esențiale în dezvoltarea competențelor de administrare a sistemelor informatice. În contextul sistemelor de operare, aceste cunoștințe oferă posibilitatea de a gestiona eficient resursele sistemului, de a programa execuția proceselor și de a facilita comunicarea și partajarea datelor între diversele componente ale sistemului.