Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică şi Microelectronică

Departamentul Ingineria Software și Automatică

**RAPORT**

Lucrarea de laborator nr. 5

# La disciplina „ Sistemul de operare”

Tema: "Monitorizarea proceselor"

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

A verificat: V. Beșliu

Chișinău – 2023

Lucrarea de laborator nr. 5

#### Scopul lucrarii**:** modurile de obținere de informații despre rularea proceselor, despre resursele utilizate de procese, prezentarea rezultatelor în diverse forme.

**Sarcină la laborator 5**

Executați următorii pași:

a) Toate script-urile și fișierele pentru extragerea rezultatelor le veți crea în subdirectorul lab5.

b) Scrieți script-uri, care vor rezolva următoarele probleme:

1) Aflaţi numărul de procese inițiaţializate de utilizatorul user și introduceți într-un fișier perechea „PID: comanda” a proceselor inițializate. Afișați conținutul acestui fișier şi numărul de procese.

2) Afișați PID-ul procesului, ultimul lansat (cu timpul de lansare).

3) Introduceți într-un fișier o listă cu PID-urile proceselor, pornite cu comenzile localizate în /sbin/. Afișați conținutul acest fișier.

4) Pentru fiecare proces, calculați diferența dintre memoria totală (statm: size) şi rezidentă (statm: resident) ale memoriei de proces (în pagini). Introduceți într-un fișier linii de tip „PID:diferența”, sortate în ordinea descrescătoare a acestor diferențe. Afișați conținutul acestui fișier.

5) Pentru toate procesele înregistrate, la moment, în sistem, introduceți într-un fişier linii de tip ProcessID=PID:Parent\_ProcessID=PPID:

Average\_Time=avg\_atom

Preluați valorile PPID și PID din fișierele status,

valoarea avg\_atom

(avg\_atom=se.sum\_exec\_runtime/nr\_switches)

din fișierele sched care se află în subdirectoarele, cu nume ce corespund PID proceselor în directorul /proc. Sortați aceste linii conform identificatorilor proceselor părinte. Rezultatul sortării introduceți într-un fișier și afișați-l.

6) În fișierul obținut în 5, după fiecare grup de înregistrări cu același identificator al procesului părinte, introduceți o linie de tip

Sum\_switches\_of\_ParentID=N is M, unde

N=PPID, iar M este suma calculată

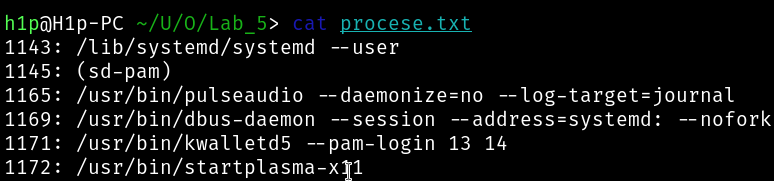
voluntary\_ctxt\_switches+

nonvoluntary\_ctxt\_switches din statuspentru acest proces. Afișați conținutul acestui fișier.

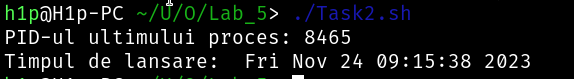
Mersul lucrări:

Aflaţi numărul de procese inițiaţializate de utilizatorul user și introduceți într-un fișier perechea „PID: comanda” a proceselor inițializate. Afișați conținutul acestui fișier şi numărul de procese.

#!/bin/bash   
output\_ps=$(ps -u $USER -o pid,cmd --no-headers)   
echo "$output\_ps" | awk '{print $1 ": " substr($0, index($0,$2))}' > procese.txt   
cat -n  procese.txt

Afișați PID-ul procesului, ultimul lansat (cu timpul de lansare).

#!/bin/bash   
ultimul\_proces=$(ps -eo pid,lstart --sort=start\_time | tail -n 1)   
echo "PID-ul ultimului proces: $(echo $ultimul\_proces | awk '{print $1}')"   
echo "Timpul de lansare: $(echo $ultimul\_proces | awk '{$1=""; print $0}')"



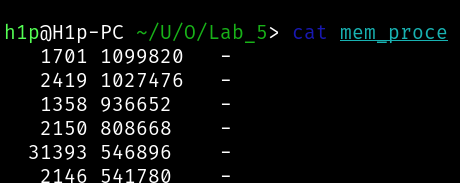
Introduceți într-un fișier o listă cu PID-urile proceselor, pornite cu comenzile localizate în /sbin/. Afișați conținutul acest fișier.

#!/bin/bash   
ps\_pid=$(ps -eo pid,cmd | grep "^ \*[0-9] /sbin/")   
echo $ps\_pid   
echo $ps\_pid | awk '{print $1}' > pid\_proces\_sbin   
cat pid\_proces\_sbin



Pentru fiecare proces, calculați diferența dintre memoria totală (statm: size) şi rezidentă (statm: resident) ale memoriei de proces (în pagini). Introduceți într-un fișier linii de tip „PID:diferența”, sortate în ordinea descrescătoare a acestor diferențe. Afișați conținutul acestui fișier.

#!/bin/bash   
#Sarcina nu are rost deoarece avem - la res   
ps\_output=$(ps -eo pid,size,resident --no-header|awk '$2>0'|sort -k2 -n -r)   
printf "%s" "$ps\_output" > mem\_proce   
cat mem\_proce



‘

Pentru toate procesele înregistrate, la moment, în sistem, introduceți într-un fişier linii de tip ProcessID=PID:Parent\_ProcessID=PPID:

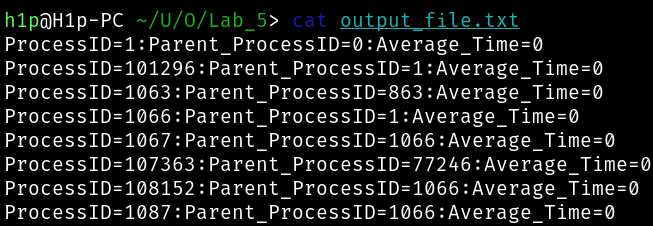
Average\_Time=avg\_atom

Preluați valorile PPID și PID din fișierele status,

valoarea avg\_atom

(avg\_atom=se.sum\_exec\_runtime/nr\_switches)

din fișierele sched care se află în subdirectoarele, cu nume ce corespund PID proceselor în directorul /proc. Sortați aceste linii conform identificatorilor proceselor părinte. Rezultatul sortării introduceți într-un fișier și afișați-l.

proc\_dir="/proc"   
  
for pid\_dir in $proc\_dir/[0-9]\*/; do   
   pid=$(basename "$pid\_dir")   
  
   status\_file="$pid\_dir/status"   
   if [ -f "$status\_file" ]; then   
       ppid=$(awk '/PPid/ {print $2}' "$status\_file")   
       sum\_exec\_runtime=$(awk '/se.sum\_exec\_runtime/ {print $2}' "$status\_file")   
       nr\_switches=$(awk '/nr\_switches/ {print $2}' "$status\_file")   
  
       if [ "$nr\_switches" -ne 0 ]; then   
           avg\_atom=$((sum\_exec\_runtime / nr\_switches))   
       else   
           avg\_atom=0   
       fi   
  
       echo "ProcessID=$pid:Parent\_ProcessID=$ppid:Average\_Time=$avg\_atom" >> output\_file.txt   
   fi   
done

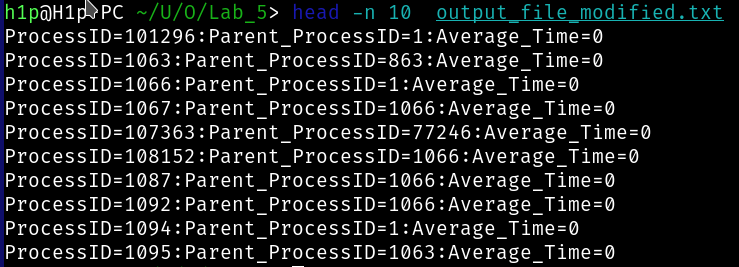
În fișierul obținut în 5, după fiecare grup de înregistrări cu același identificator al procesului părinte, introduceți o linie de tip

Sum\_switches\_of\_ParentID=N is M, unde

N=PPID, iar M este suma calculată

voluntary\_ctxt\_switches+

nonvoluntary\_ctxt\_switches din statuspentru acest proces. Afișați conținutul acestui fișier.

input\_file="output\_file.txt"   
output\_file="output\_file\_modified.txt"   
  
sort -t= -k4,4n -k8,8n "$input\_file" > "$output\_file"   
  
current\_parent\_id=""   
sum\_switches=0   
  
while IFS= read -r line   
do   
   ppid=$(grep -oP 'Parent\_ProcessID=\K\d+' <<< "$line")   
   switches=$(grep -oP 'voluntary\_ctxt\_switches:\s+\K\d+' <<< "$line")   
  
   if [ "$ppid" = "$current\_parent\_id" ]; then   
       sum\_switches=$((sum\_switches + switches))   
   else   
       echo "Sum\_switches\_of\_ParentID=$current\_parent\_id is $sum\_switches" >> "$output\_file"   
       current\_parent\_id="$ppid"   
       sum\_switches="$switches"   
   fi   
done < "$output\_file"   
  
echo "Sum\_switches\_of\_ParentID=$current\_parent\_id is $sum\_switches" >> "$output\_file"   
  
cat "$output\_file"

Concluzie: În concluzie, monitorizarea proceselor într-un sistem informatic este esențială pentru înțelegerea și optimizarea utilizării resurselor. Prin implementarea unor sarcini precum identificarea și înregistrarea proceselor, analiza diferențelor în utilizarea memoriei, și evaluarea performanțelor în funcție de timpul de execuție, putem obține o perspectivă detaliată asupra activității sistemului. Această monitorizare nu doar oferă informații utile pentru identificarea problemelor de performanță, ci și pentru optimizarea generală a resurselor și proceselor, contribuind astfel la asigurarea unui mediu informatic eficient și stabil.