Ministerul Educației, Culturii și Cercetării al Republicii Moldova  
Universitatea Tehnică a Moldovei  
Facultatea Calculatoare, Informatică și Microelectronică  
Departamentul Ingineria Software și Automatica

Disciplina: SOMIPP

**RAPORT**

Lucrarea de laborator nr. 5

**Tema**: “Monitorizarea proceselor”

A efectuat: st.gr:TI-202 Bodrug Mihai

A verificat: asist. univ. Colesnic Victor

Chișinău – 2022

**Scopul lucrării:** Sa se studieze modurile de obținere de informații despre rularea proceselor, despre resursele utilizate de procese, prezentarea rezultatelor în diverse forme. Un proces reprezintă un program în execuție și are ataşate o serie de informaţii specifice precum instrucțiunile programului, resurse folosite (precum fişiere deschise), unul sau mai multe fire de execuție şi alte informații necesare procesului de execuţie în paralel.

Fiecărui proces în Linux îi este asociat un identificator PID (Process Identifier) format dintr-un număr care ia valori între 0 și 65535.

Procesele în Linux sunt ierarhizate sub forma unui arbore, având ca rădăcină procesul init. Părintele - A, al unui proces - B, este procesul ce a creat procesul B. ID-ul procesului părinte este referit din perspectiva procesului copil ca PPID (Parent Process ID). PPID al procesului init este 0.

Orice proces Linux va avea un set de caracteristici comune, ce oferă informații despre acesta:

PID – sau Process ID, este un identificator de proces sub forma unui număr întreg unic.

PPID – similar cu PID, cu excepția că reprezintă identificatorul procesului care a dat naștere procesului curent (cunoscut şi ca proces părinte).

Terminalul atașat – prescurtat TTY, reprezintă terminalul la care procesul curent este ataşat.

RUID – Real User ID, reprezintă identificatorul utilizatorului care a lansat aplicația. Similar există şi EUID (sau Effective User ID) pentru identificarea drepturilor reale la resursele sistemului.

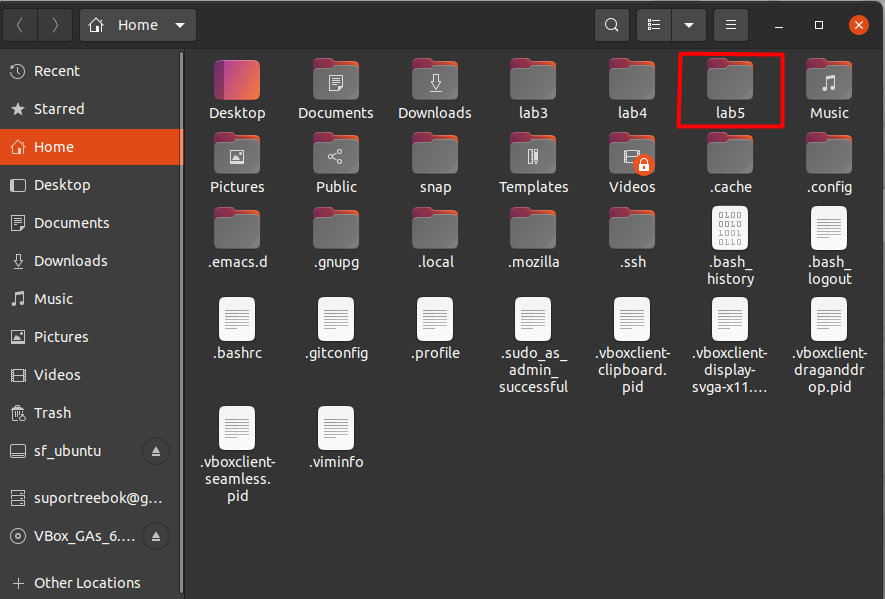
RGID şi EGID – similar cu RUID şi EUID, doar că se referă la identificatorul grupului de utilizatori.

factorul nice – folosit pentru a determina, aşa cum sugerează şi numele, „factorul de prietenie” al procesului cu scopul stabilirii priorităţii de execuţie.

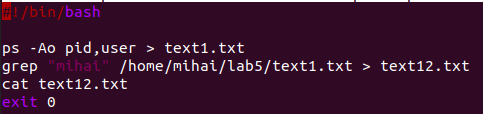
**Sarcină pentru lucrarea de laborator :**

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



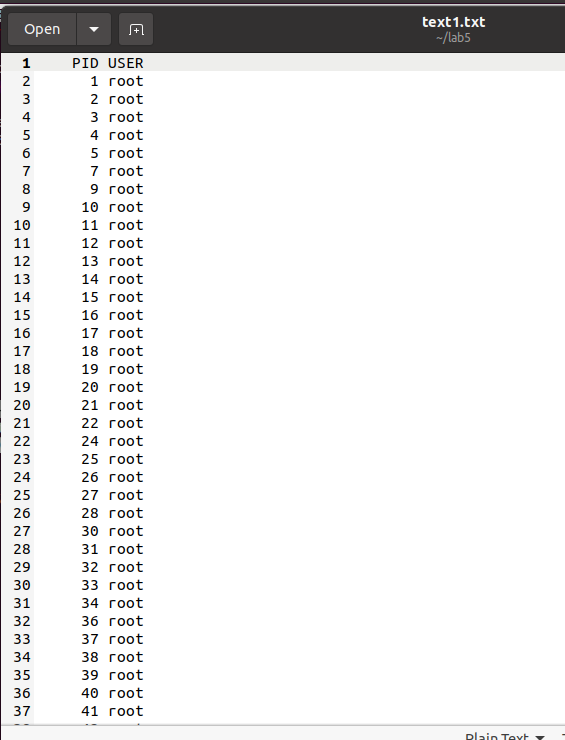
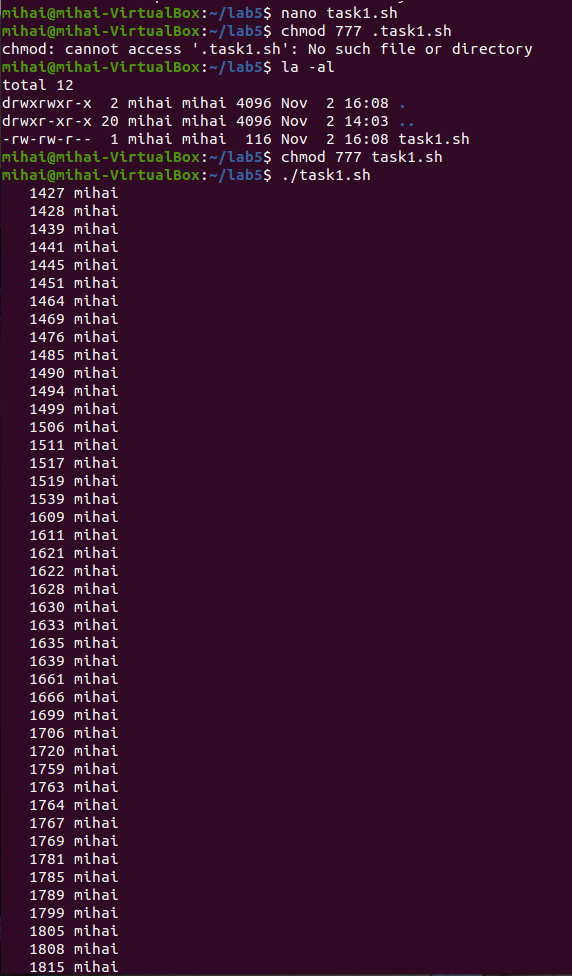


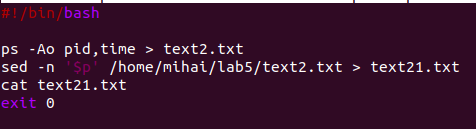
1. **Scrieți script-uri, care vor rezolva următoarele probleme:**
2. Aflaţi numărul de procese inițiaţializate de utilizatorul user și introduceți într-un fișier perechea „PID: comanda” a proceselor inițializate. Afișați conținutul acestui fișier şi numărul de procese.

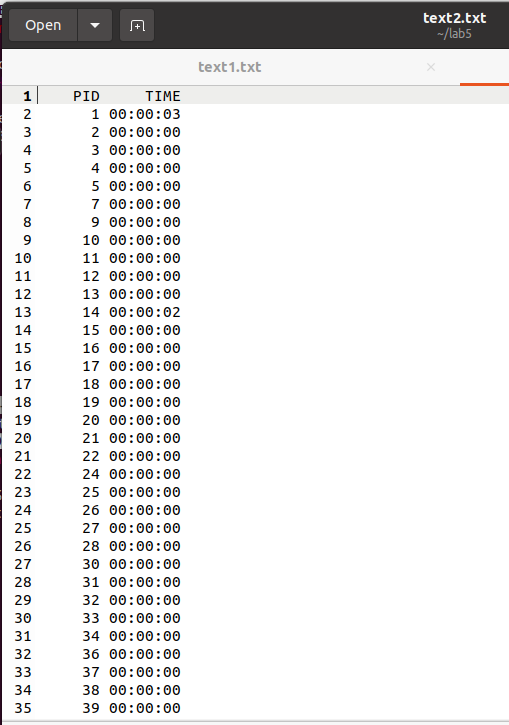


#ps - process status

#A - Select all process



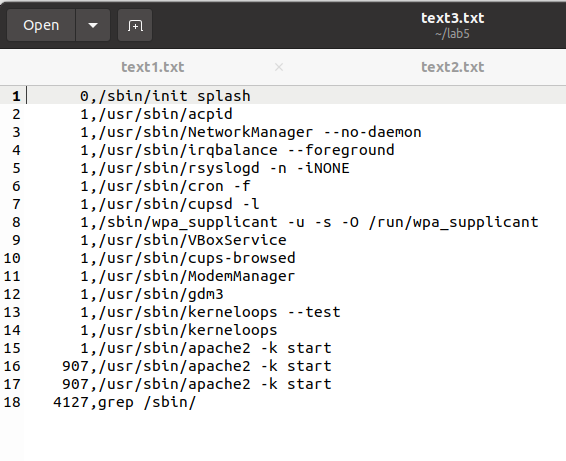
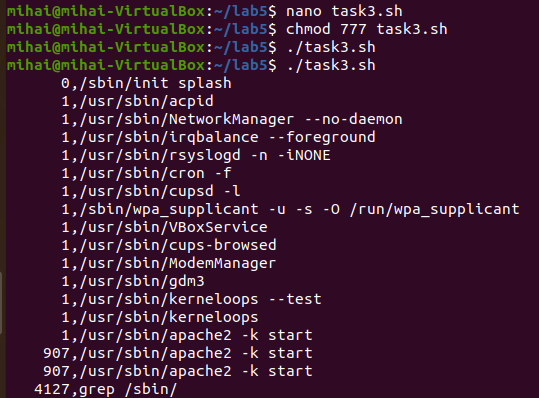
1. Afișați PID-ul procesului, ultimul lansat (cu timpul de lansare). 



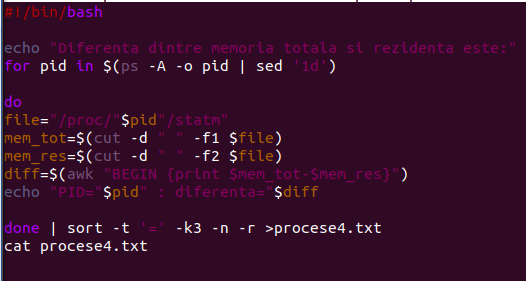
C:\Users\bodru\AppData\Local\Microsoft\Windows\INetCache\Content.Word\somipp3.png



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



1. 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.



#echo - afisarea unei linii de text

# for – operator de ciclu iterativ.

# pid - selecteaza dupa id-ul procesului

#$-toata valoarea din paranteze va fi salvata in $

#ps -A - afiseaza procesele care sunt in executie cu un format definit de utilizator

#-o - Afișează informații conform specificațiilor de format date.

Pipe-iesirea unei comenzi devine intrarea pentru urmatoarea

#Sed - determină, reieșind din spațiul de adrese dat, liniile asupra cărora vor fi executate operațiile.

#'1d' - elimina prima linie a fișierului de intrare

Do-pentru ciclu for

File-variabila in care salvam directoriu respective

$pid – pidul procesului

Memoria totala –variabile

Memoria rezidenta -variabila

# cut - extrage unele câmpuri din fișierele textuale

#-d - cu această opțiune, utilizatorul își setează propriul delimitator în locul TAB-ului standard.

#-f (--fields=LIST) - listă de câmpuri de tăiat.

$file-valoarea variabilei file

Diff-variabila in care se inscribe diferenta dintre mem totala si cea rezidenta

#awk - împarte fiecare linie în câmpuri individuale

#sort -nr - sortarea descrescatoare

# -t - permite să specificam un separator de obiecte de sortare diferit în locul unui spațiu.

#-k3 - dupa campul 3 cu conditia ca este spatiu

# -n –r ca sa fie descrescatoare sortarea

#cat - afiseaza fisierul pe ecran

