**MINISTERUL EDUCAŢIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Programul de studii: Tehnologia informației**

RAPORT

# LUCRARE DE LABORATOR NR. 4

# la Sisteme de Operare

**Tema:Procesarea fluxurilor textuale în SO GNU/Linux***.*

A efectuat:

st. gr. TI-211 Popa Cătălin

A verificat: Victor Colesnic

UTM, Chișinău 2023

**Tema:** Procesarea fluxurilor textuale în SO GNU/Linux.

**Scopul lucării:** Studierea procesului, intrarea și ieșirea standart a procesului, legarea procesului de I/O, utilizarea ieșirii unui proces ca parametru al unui alt proces, expresii regulate și filtrarea fluxurilor textuale.

**Sarcinile pentru lucrarea de laborator:**

Noțiuni

1. **stdin** – Este canalul de intrare standart, care permite unui program să primească date de la utilizator sau din alte surse. De obicei, acest canal este conectat la tastatură sau la un alt program care furnizează date.

2. **stdout** – Este canalul de ieșire standart, utilizat pentru a afișa rezultatele programului. De obicei, acest canal este conectat la ecran sau la un alt program care preia datele.

3. **stderr** – Este canalul de eroare standart, utilizat pentru a afișa mesaje de eroare. Acest canal este, de asemenea, conectat la ecran sau la un alt program care preia mesaje de eroare.

Instrucțiuni

1. **sort** – Utilizată pentru a sorta liniile unui fișier sau a unei intrări standart într-un anumit mod.

2. **uniq** – Elimină liniile dublicate dintr-un fișier care a fost sortat.

3. **cut** – Este folosit pentru a extrage câmpuri specifice dintr-un fișier text. Prin câmp, înțelegem secvență de simboluri până la spațiu.

4. **head** – Afișeză primele linii dintr-un fișier la stdout.

5. **tail** – Afișeză ultimele linii dintr-un fișier la stdout.

6. **tr** – înlocuiește unele simboluri cu altele.

Destinația utilitarelor

1. **grep** - Utilară pentru căutarea și sortarea liniilor introduse și afișarea anumitor tipare(șiruri de caractere) ca rezultat al procesării.

2. **sed** – (Stream Editor) dispozitivul stdin sau dintr-un fișier text, execută unele operații asupra liniilor după care afișează rezultatul pe stdout sau într-un fișier. Sed determină, reieșind din spațiul de adrese dat, liniile asupra cărora vor fi executate operațiile. Spațiul de adresă al liniilor este desemnat de numărul de ordine al acestora sau de șablon. De exemplu, comanda 3d va obliga sed să elimine linia a treia, iar instrucțiunea /windows/d va conduce la eliminarea tuturor liniilor, care conțin cuvântul "**windows**". Cel mai frecvent sunt utilizate comenzile: **p** – imprimare (pe stdout), **d** – eliminare/ștergere și **s** – înlocuire.

3. **awk** – Comanda awk – utilitară pentru căutarea contextuală și transformare a textului, instrument pentru extragerea și/sau procesarea câmpurilor (coloanelor) în fișiere text structurate. Awk împarte fiecare linie în câmpuri individuale. În mod implicit, câmpurile sunt secvențe de caractere separate prin spații, dar există posibilitatea de a desemna alt caracter ca separator de câmp. Awk analizează și procesează fiecare câmp aparte.

**Clase de caractere *POSIX****.*

Clasele de caractere POSIX sunt utilizate în expresiile regulate pentru a defini grupuri de caractere cu anumite proprietăți comune. Acestea sunt folosite cu instrumente precum ***grep*** și ***sed***. Exemple de clase de caractere POSIX includ:

− [:class:] metodă alternativă de a specifica un diapazon de caractere.

− [:alnum:] corespunde caracterelor alfanumerice. Este echivalentă expresiei [A-Za-z0-9].

− [:alpha:] corespunde caracterelor alfabetului. Este echivalentă expresiei [A-Za-z].

− [:blank:] corespunde caracterului spațiu sau tabulare.

− [:cntrl:] corespunde caracterelor de control.

− [:digit:] corespunde setului de cifre zecimale. Este echivalentă expresiei [0-9].

− [:lower:] corespunde caracterelor alfabetului din registrul inferior. Echivalentă expresiei [a-z].

− [:space:] corespunde caracterelor de spațiere (spațiu și tabulare orizontală).

− [:graf:] corespunde unui set de caractere din gama ASCII 33 - 126.

− [:print:] corespunde unui set de caractere din gama ASCII 32 – 126.

− [:upper:] corespunde caracterelor alfabetului din registrul superior. Echivalentă expresiei [A-Z].

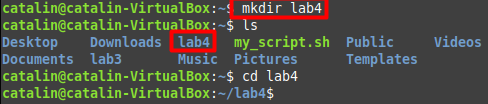
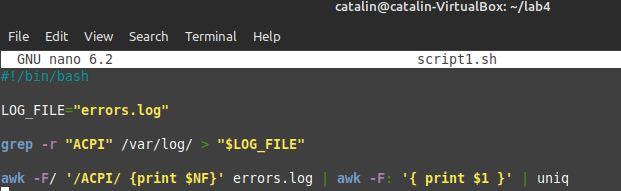
− [:xdigit:] corespunde setului de cifre hexazecimale. Este echivalentă expresiei [0-9A-Fa-f].

Aceste noțiuni și instrumente sunt fundamentale în lucrul cu sistemul de operare GNU/Linux și în prelucrarea textului în mediul Unix.

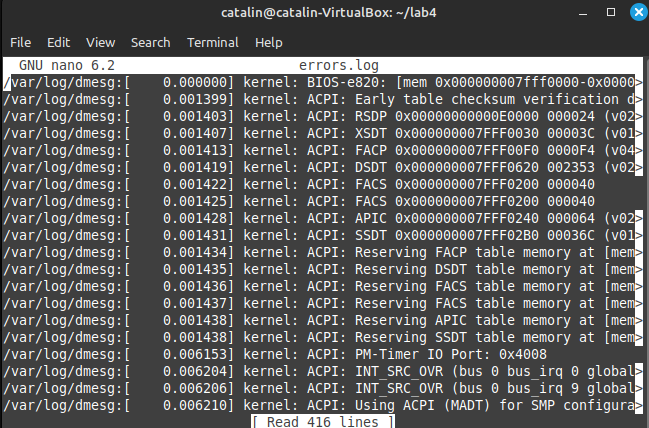
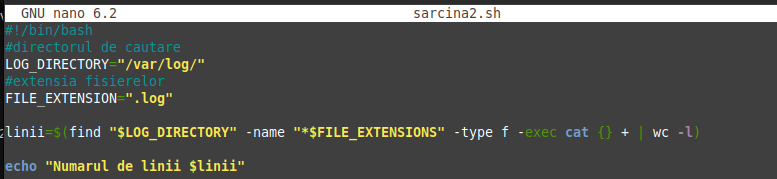
Comanda "uniq -c" în limbajul Unix/Linux este folosită pentru a număra aparițiile consecutive ale liniilor într-un fișier sau flux de date și pentru a afișa numărul de apariții al fiecărei linii alături de linia respectivă.

Mai precis:"uniq" este comanda principală, care elimină liniile duplicate consecutive dintr-un fișier sau flux de intrare.

"-c" este opțiunea care cere comandei să afișeze, înaintea fiecărei linii un număr care reprezintă numărul de apariții consecutive ale liniei respective.

1. a) Creați un nou subdirector unde veți plasa toate script-urile și fișierele pentru extragerea rezultatelor.
2. 
3. **Figura 1** – Creare directoriu nou.
4. b) Scrieți script-uri, care vor rezolva următoarele probleme:
5. **1)** Creați fișierul **errors.log**, în care veți plasa toate liniile din toate fișierele directorului **/var/log/** accesibile pentru citire, care conțin secvența de caractere **ACPI**. Afișați pe ecran liniile din **errors.log**, care conțin numele fișierelor, excluzând calea acestor fișiere.
6. **grep** – caută șiruri de caractere **-r** – căutarea trebuie să fie recursivă. **/var/log/** – fișiere de jurnal
7. **>** - operatorul de redirecționare. Ia ieșirea standart a comenzii **(grep -r "ACPI" /var/log/)** și o redirecționează către fișierul errors.log. Dacă fișierul există, conținutul va fi suprascris.
8. 
   1. **Figura 2** – Creare fisier errors.log.

Aceasta va căuta recursiv în toate fișierele din directoriul /var/log/ și va adăuga liniile care conțin ACPI în fișierul errors.log.

* + 1. 
    2. **Figura 3** – Afisare fisier errors.log.
    3. 
    4. **Figura 4** – Afisare liniile din errors.log fara calea acestora.
    5. **/** - separator **$NF** – numele fișierului fără calea completă.
    6. **awk** – este comandă AWK, care este un limbaj de programare și un utilitar de procesare a textului în linia de comandă în sistemele Unix și Linux.
    7. **-F/** - Este opțiunea AWK care specifică un separator de câmpuri. În acest caz, separatorul este "/", ceea ce înseamnă că AWK va trata câmpurile din linia de intrare separate de caracterele "**/**".
    8. **{print $NF}** - Aceasta este expresia AWK care specifică acțiunile pe care AWK trebuie să le efectueze pe fiecare linie de intrare. În acest caz, print $NF înseamnă să imprime (afișeze) ultimul câmp din linie ($NF este variabila AWK care reprezintă ultimul câmp).
    9. **2)** Calculați numărul total de linii în fișierele directorului **/var/log/** cu extensia **log**. Afișați acest număr pe ecran.
    10. 
    11. **Figura 5** – Afisare numar de linii.
    12. **total\_lines=$(find "$LOG\_DIRECTORY" -name "\*$FILE\_EXTENSION" -type f -exec cat {} + | wc -l):** Aceasta este comanda principală pentru calculul numărului total de linii în fișierele cu extensia .log.
    13. **-** **find "$LOG\_DIRECTORY" -name "\*$FILE\_EXTENSION" -type f**: Folosește comanda find pentru a găsi toate fișierele cu extensia .log din directorul specificat.
    14. - **exec cat {} +** : Aplică comanda cat pentru a concatena conținutul acestor fișiere.
    15. - **| wc -l**: Folosește comanda wc -l pentru a număra liniile rezultate din comanda cat. Rezultatul este salvat în variabila total\_lines.

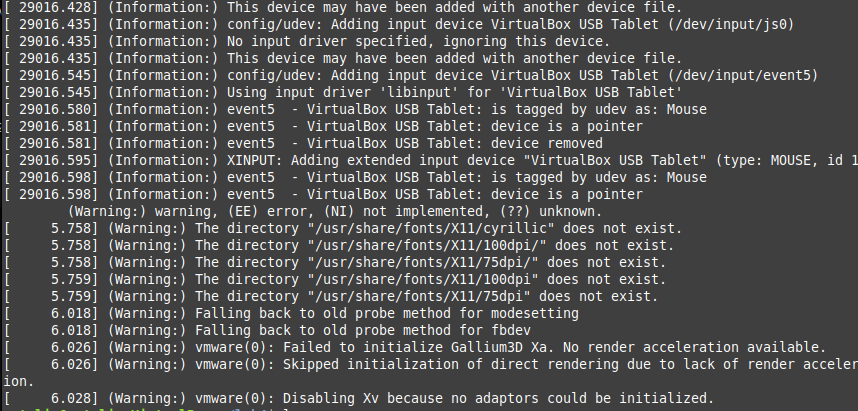
1. **3)** Creați fișierul **full.log**, în care veți plasa liniile din fișierul **/var/log/Xorg.0.log**, care conțin avertizări și mesaje informaționale, înlocuind marcherii avertizărilor (**WW**) și mesajelor informaționale (**II**) cu cuvintele **Warning**: și **Information:**, astfel încât în fișierul rezultat mai întâi să apară toate mesajele informaționale, iar apoi toate avertizările. Afișați conținutul fișierului **full.log** pe ecran.
2. 
3. **Figura 6** – Creare fisier full.log cu cuvintele Warning.

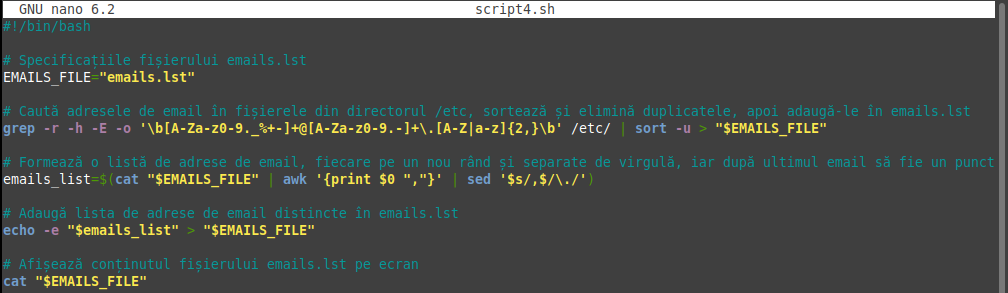
**grep -E 'II' /var/log/Xorg.0.log | sed -e 's/II/Information:/' > "$FULL\_LOG**":

* **grep -E 'II' /var/log/Xorg.0.log** : comanda grep pentru a căuta liniile care conțin marcatorul II (mesaje informative) în fișierul /var/log/Xorg.0.log.
* **sed -e 's/II/Information:/'** : sed pentru a înlocui marcatorul II cu cuvântul "Information:".
* **> "$FULL\_LOG"** : Redirectează rezultatele în fișierul full.log.

**grep -E 'WW' /var/log/Xorg.0.log | sed -e 's/WW/Warning:/' >> "$FULL\_LOG"** :

* **grep -E 'WW' /var/log/Xorg.0.log**: grep pentru a căuta liniile care conțin marcatorul WW (avertizări) în același fișier.
* **sed -e 's/WW/Warning:/'** : sed pentru a înlocui marcatorul WW cu cuvântul "Warning:".
* **>> "$FULL\_LOG"** : Redirectează rezultatele, de data aceasta adăugându-le la conținutul existent al fișierului full.log (utilizat pentru a adăuga la sfârșitul fișierului).

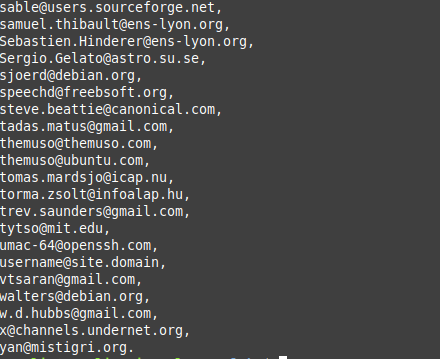
1. 
2. **Figura 7** – Creare fisier full.log cu cuvintele Information.
3. **4)** Creați fișierul **emails.lst**, în care veți plasa toate adresele de poștă electronică, depistate în fișierele directorului **/etc**. Adresele vor urma consecutiv, separate prin virgulă. Afișați conținutul fișierului **emails.lst** pe ecran.



**Figura 8** – Adresele de poștă electronică. @at-sain

**grep -r -h -E -o '\b[A-Za-z0-9.\_%+-]**[**+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b' /etc/**](mailto:+@[A-Za-z0-9.-]+\.%5bA-Z|a-z%5d%7b2,%7d\b'%20/etc/)**:**

* **grep -r -h -E -o** : Caută recursiv în fișiere (-r), fără a afișa numele fișierului (-h), utilizează expresii regulate (-E) și returnează doar textul potrivit (-o).
* Expresia regulată ('\b[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b') este utilizată pentru a găsi adresele de email valide în fișierele din directorul /etc.



**Figura 9** – Afișare emailuri.

* **'\b’** : Aceasta indică începutul unui cuvânt. \b este un delimitator de cuvinte (word boundary), asigurând că adresa de email începe la începutul unui cuvânt.
* **[A-Za-z0-9.\_%+-]+** :Această parte a expresiei corespunde caracterelor care pot apărea în partea locală a adresei de email. Detaliile sunt următoarele:
  + [A-Za-z]: Literele majuscule și minuscule.
  + 0-9: Cifre.
  + .\_%+-: Caractere specifice care sunt permise în partea locală a adresei de email.
* **@ :** Această parte corespunde caracterului "@" care delimitează partea locală de domeniul adresei de email.
* **[A-Za-z0-9.-]+ :** Aceasta corespunde caracterelor care pot apărea în domeniul adresei de email. Detaliile sunt următoarele:
  + [A-Za-z]: Literele majuscule și minuscule.
  + 0-9: Cifre.
  + .- : Caractere specifice care sunt permise în domeniu, inclusiv puncte.
* **\.** : Aceasta corespunde literal caracterului punct care separă numele de domeniu.
* **[A-Z|a-z]{2,} :** Aceasta corespunde cu două sau mai multe litere, reprezentând domeniul de nivel superior (TLD). Detaliile sunt următoarele:
  + [A-Z|a-z]: Literele majuscule și minuscule din TLD.
  + {2,}: Semnifică că trebuie să existe cel puțin două astfel de litere pentru a fi o adresă de email validă.
* **awk '{print $0 ","}'** - Rezultatul afișat de comanda cat este trecut printr-un filtru awk. Aceasta adaugă virgulă la sfârșitul fiecărei linii. $0 reprezintă întreaga linie.
* **sed '$s/,$/\./'** - Rezultatul filtrului awk este apoi trecut printr-un alt filtru sed. Aceasta înlocuiește ultima virgulă de pe ultima linie cu un punct (dot).

**Cat** – concatineaza **grep** – cauta **awk** – utilara de cautare

1. **5)** Găsiți în directorul **/bin** toate fișierele, ce reprezintă script-uri și afișați pe ecran numele interpretoarelor cu numărul ce indică frecvența lor de utilizare (în script-uri). Script-urile pot începe cu una din următoarele linii:

#!/bin/sh

#!/bin/bash

#!/usr/bin/perl

#!/usr/bin/tcl.



**Figura 10** - Găsire scripturi.

**grep -rlE '^#!/bin/bash' /bin**: Caută recursiv în directorul /bin fișiere care conțin linia #!/bin/bash.

**grep**: Comandă utilizată pentru căutarea de șabloane în fișiere.

**-r**: Opțiunea care specifică căutarea recursivă în directoare și subdirectoare.

**-l**: Opțiunea care afișează doar numele fișierelor care conțin șablonul, nu și liniile care se potrivesc.

**-E**: Opțiunea care indică faptul că șabloanele date sunt expresii regulate extinse (ERE). Expresiile regulate extinse sunt un tip de limbaj pentru definirea șabloanelor de căutare mai complexe și puternice.

**local files=$1** : Definește o variabilă locală files care primește primul argument al funcției.

**local shebang=$2** : Definește o variabilă locală shebang care primește al doilea argument al funcției.

**if [ -n "$files" ]; then ... else ... fi** : Verifică dacă variabila files nu este goală (adică există fișiere).

**grep -h "^$shebang" $files | awk -F'/' '{print $4}' | sort | uniq -c | awk '{print $2 " : " $1 " utilizari"}':** Afișează utilizările pentru un anumit shebang.echo "0 utilizari pentru $shebang": Afișează "0 utilizari pentru [shebang]" dacă nu există fișiere.

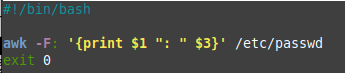
**grep -h "^$shebang" $files :**

**-h:** Opțiunea care oprește afișarea numelui fișierelor în rezultat.

**"^$shebang"** : Șablonul căutat. **^** indică începutul liniei, iar $shebang este valoarea din variabila shebang.

**awk -F'/' '{print $4}' -** Această parte utilizează AWK pentru a extrage numele fișierului din calea completă a fișierului. De exemplu, pentru /usr/bin/example, va extrage example.

**6)** Afișați lista utilizatorilor sistemului cu indicarea **UID** al fiecăruia, cu sortarea după **UID**. Informațiile despre utilizatori sunt păstrate în fișierul **/etc/passwd**. În fiecare linie a acestui fișier primul câmp conține numele utilizatorului, iar câmpul al treilea - **UID,** separatorul – două puncte (**:**).



**Figura 11** – Afișare utilizatori.

awk -F: '{print $1 ": " $3}' /etc/passwd:

* **awk** : Este o utilitară puternică pentru procesarea textului în linia de comandă.
* **-F:** : Specifică separatorul de câmpuri, care în acest caz este două puncte (:). Aceasta indică că AWK trebuie să interpreteze câmpurile în linia de intrare ca separate de două puncte.
* **'{print $1 ": " $3}'** : Aceasta este expresia AWK care indică să se printeze primul câmp ($1 - numele utilizatorului), urmat de două puncte și un spațiu, apoi al treilea câmp ($3 - ID-ul utilizatorului).
* **/etc/passwd** : Este calea către fișierul /etc/passwd, care conține informații despre utilizatori într-un sistem Unix/Linux.

**7)** Afișați 5 cel mai frecvent întâlnite cuvinte din **man bash** cu lungimea de cel puțin 3 caractere.

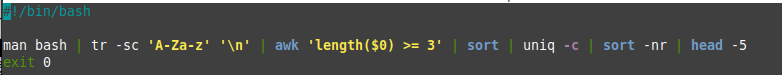


Figura 12 – Afișare 5 cel mai frecvent cuvinte intâlnite.

* **man bash** : Inițiază afișarea paginilor manualului pentru shell-ul Bash. Comanda man este folosită pentru a afișa paginile manuale pentru diverse comenzi și utilitare.
* **tr -sc 'A-Za-z' '\n'**: Traduce caracterele folosind comanda tr. Opțiunea -s stoarce multiple apariții ale setului specificat de caractere într-o singură apariție. Opțiunea -c completează setul de caractere, iar aici menține doar caracterele alfabetice (A-Za-z). În final, înlocuiește toate aparițiile ale acestor caractere cu caracterele noi de linie (\n).
* **awk 'length($0) >= 3**: Awk este folosit pentru a filtra liniile bazat pe lungimea lor. Afișează doar liniile în care lungimea întregii linii ($0) este mai mare sau egală cu 3.
* **sort**: Ordonează liniile textului alfabetic.
* **uniq -c**: Numără aparițiile fiecărei linii unice.
* **sort -nr**: Ordonează rezultatul în ordine numerică inversă (-n pentru numeric, -r pentru invers).head -5: Afișează primele 5 linii ale rezultatului.