

# **UserFS**

## **(Monitorizarea utilizatorilor activi în sistem utilizând un script shell)**

*Sleep Token*

*Drăgunoi Miruna, Panaet Maria-Alexandra*

### **Rezumat al temei de proiect**

Proiectul constă în dezvoltarea unui script shell care automatizează monitorizarea utilizatorilor activi într-un sistem Linux/UNIX. Soluția presupune crearea unui director pentru fiecare utilizator activ, ce conține informații despre procesele acestuia, actualizarea periodică a datelor și gestionarea informațiilor despre utilizatorii inactivi. Raportul include descrierea problemei, specificația și designul soluției, implementarea și evaluarea performanței.

### **Sumarul capitolelor**

1. Descrierea problemei
2. Specificația soluției
3. Design
4. Implementare
5. Evaluarea soluției
6. Concluzii

### **Descrierea problemei**

*Context general:* Gestionarea resurselor utilizatorilor activi este critică pentru administrarea sistemelor Linux/UNIX. Automatizarea acestui proces reduce timpul necesar monitorizării manuale.

### **Obiective**

- Automatizarea procesului de monitorizare a utilizatorilor activi.
- Crearea unui sistem fiabil pentru actualizarea periodică a datelor despre utilizatori.
- Gestionarea corectă a utilizatorilor inactivi prin logarea sesiunilor.

### **Cerințe generale**

- Funcționale:
  - Detectarea utilizatorilor activi.
  - Crearea și actualizarea fișierelor procs și lastlogin.
- Tehnice:
  - ❖ Sistem de operare: Linux/UNIX.
  - ❖ Utilizarea comenzilor shell (ps, who, awk, etc.).
  - ❖ Compatibilitate cu toate distribuțiile Linux.

### **De ce este nevoie de o soluție?**

Monitorizarea manuală a proceselor utilizatorilor este inefficientă și predispusă la erori. Automatizarea asigură consistență și eficiență.

### **Posibile soluții**

1. Script Shell: Simplitate, dar limitări în scalabilitate.
2. Script Python: Flexibilitate mai mare, dar necesar de resurse suplimentare.

### **Evaluarea soluției**

- Corectitudine: verificarea fișierelor generate pentru fiecare utilizator activ.
- Performanță: măsurarea timpului de actualizare a informațiilor.

### **Specificația soluției**

Soluția noastră are și presupuneri, dar și constrângeri, cum ar fi faptul că scriptul funcționează exclusiv pe Linux/UNIX, iar actualizarea datelor este periodică și configurabilă la 30 de secunde.

### **Caracteristici ale prototipului**

*Descriere:* Script-ul nostru shell este un mecanism ce rezolvă problema prin intermediul creării unui set de directoare și fișiere pentru utilizatorii activi.

*Drept cerințe urmărite în timpul realizării script-ului shell avem:* sistemul de operare să fie unul de Linux, iar comenzi shell folosite să fie unele standard.

*Use case-uri urmărite:* detectarea utilizatorilor activi din sistemul de operare, crearea/actualizarea fișierelor procs (ce memorează istoricul activității unui utilizator activ) și crearea fișierului lastlogin pentru utilizatorii inactivi (fișier ce memorează data ultimei activități ale utilizatorului inactiv din sistem).

### **Design**

Arhitectura software este alcătuită din următoarele puncte atinse:

- Detectarea utilizatorilor activi utilizând who.
- Crearea directoarelor pentru utilizatori activi.
- Scrierea proceselor curente într-un fișier procs utilizând ps -u.
- Crearea fișierului lastlogin pentru utilizatorii inactivi.

În alegerea mecanismelor am urmărit următoarele:

- Avantaje: simplitatea scriptului shell și o performanță ridicată pentru operații simple.
- Dezavantaje: e mai greu de extins pentru scenarii complexe.

### **Diagramă procese proiect**

Detectare utilizatori	Creare directoare	Scriere procese în fișierul procs	Creare fișier lastlogin
1	2	3	4

### **Implementare**

#### ***Componente software surprinse în proiect:***

1. Script shell principal.

## 2. Comenzi utilizate:

- **who**: detectare utilizatori.
- **ps**: listare procese active.
- **mkdir, touch**: gestionare fișiere/directoare.

### ***Probleme tehnice întâlnite pe parcursul proiectului:***

- Gestionarea numelor de utilizatori cu caractere speciale.
- Verificarea drepturilor de acces.

### ***Soluții implementate:***

- Escaparea caracterelor speciale.
- Verificarea existenței fișierelor/directoarelor.

### **Evaluarea soluției**

#### **Plan de evaluare urmărit:**

- Am încercat verificarea corectitudinii fișierelor generate constant.
- S-a realizat măsurarea timpului de actualizare pentru fiecare proces în parte.

#### **Experimente:**

- **Arhitectură:**
  - Sistem: 4 core-uri, 4 GB RAM, Linux.
  - Software: Shell Bash 5.0.
- **Testare:**
  - 10 utilizatori activi simultan.
  - Actualizare periodică la 30s.

---

## **Concluzii**

- Scriptul shell asigură o soluție eficientă și robustă pentru monitorizarea utilizatorilor activi.
- Limitările actuale pot fi abordate prin utilizarea unui limbaj de programare mai avansat precum Python.