

# UserFS

## 1. INTRODUCERE

În cadrul acestui proiect, am implementat un script shell care monitorizează utilizatorii activi dintr-un sistem Linux. Scriptul creează un director pentru fiecare utilizator activ, în care se află un fișier procs ce listează procesele curente ale utilizatorului. Informațiile despre utilizatori sunt actualizate periodic, iar pentru utilizatorii care nu mai sunt activi, scriptul creează un fișier procs gol și un fișier suplimentar lastlogin care afișează data ultimei sesiuni a utilizatorului.

## 2. DESCRIEREA PROBLEMEI

Pe un sistem Linux, administrarea utilizatorilor activi și a proceselor acestora poate fi esențială pentru monitorizarea performanței și a securității. În acest context, problema constă în:

- Identificarea utilizatorilor activi din sistem.
- Crearea unui director pentru fiecare utilizator activ și stocarea informațiilor despre procesele curente într-un fișier procs.
- Actualizarea periodică a acestor informații.
- Gestionarea utilizatorilor care nu mai sunt activi, prin crearea unui fișier procs gol și a unui fișier lastlogin care conține data ultimei sesiuni de login.

## 3. DESCRIEREA SOLUȚIEI ADOPTATE

Soluția adoptată este un script shell care:

- **Detectează utilizatorii activi:** Folosește comanda who pentru a obține lista utilizatorilor activi și comanda ps pentru a colecta procesele acestora.
- **Creează un director pentru fiecare utilizator activ:** Fiecare utilizator activ va avea un director dedicat în cadrul unui director rădăcină (/tmp/active\_users).
- **Actualizează informațiile periodic:** La fiecare 30 de secunde, scriptul actualizează informațiile despre utilizatori și procesele acestora.

- **Gestionează utilizatorii inactivi:** Dacă un utilizator nu mai este activ, scriptul golește fișierul procs și creează un fișier lastlogin cu data ultimei sesiuni.

Scriptul folosește comenzi precum ps, who, last, și awk pentru a colecta informațiile necesare și a le salva în fișiere.

## **COD**

```
#!/bin/bash

# Directorul rădăcină unde vor fi stocate informațiile despre utilizatori
ROOT_DIR="/tmp/active_users"
# Intervalul de actualizare
INTERVAL=30

#Verificam dacă directorul rădăcină există
mkdir -p "$ROOT_DIR"

while true; do
    echo "- Actualizarea informațiilor despre utilizatori -"

    # Obținem lista utilizatorilor activi din sistem
    active_users=$(who | awk '{print $1}' | sort )
    # who: Listează utilizatorii conectați în sistem
    # awk '{print $1}': Extrage numele utilizatorilor
    # sort : Sortează utilizatorii

    echo "Utilizatorii activi detectați: $active_users"

    # Actualizează informațiile pentru utilizatorii activi
    for user in $active_users; do
        echo "Procesăm utilizatorul: $user"

        # Creează directorul pentru utilizatorul curent
        user_dir="$ROOT_DIR/$user"
        mkdir -p "$user_dir"

        # Actualizează fișierul "procs" cu procesele utilizatorului
        ps -u "$user" -o pid,cmd > "$user_dir/procs"
        # ps -u "$user": Listează procesele utilizatorului
        # -o pid,cmd: Afișează doar ID-ul procesului și comanda asociată
        if [ $? -eq 0 ]; then
            # $? : codul ultime comenzi
            echo "Fișierul 'procs' pentru $user a fost actualizat."
```

```

echo "Procesele curente ale utilizatorului $user:" > "$user_dir/procs"
ps -u "$user" -o pid,cmd >> "$user_dir/procs"
cat "$user_dir/procs"
echo "-----"
else
echo "A apărut o eroare la actualizarea fișierului 'procs' pentru $user."
fi
done

# Scriptul parcurge toate directoarele din ROOT_DIR și verifică dacă utilizatorul respectiv nu mai este
activ
for user_dir in "$ROOT_DIR"/*; do
user=$(basename "$user_dir")
if ! echo "$active_users" | grep -q "^$user$"; then
echo "Utilizatorul $user nu mai este activ. Golesc fișierul 'procs' și actualizez 'lastlogin'."

# Golim fișierul "procs"
> "$user_dir/procs"

# Cream/actualizeazăm fișierul "lastlogin" cu data ultimei sesiuni
last_login=$(last -n 1 "$user" | grep "$user" | head -n 1 | awk '{print $4, $5, $6, $7}')
# last -n 1 "$user": Afișează ultima sesiune a utilizatorului
if [ -n "$last_login" ]; then
echo "$last_login" > "$user_dir/lastlogin"
echo "Ultimul login pentru utilizatorul $user: $last_login"
else
echo "Data ultimei sesiuni nu este disponibilă pentru $user." > "$user_dir/lastlogin"
fi
fi
done

echo "- Finalizat ciclul de actualizare. Așteptăm $INTERVAL secunde. -"
sleep "$INTERVAL"
done

```

## 4. DIFICULTĂȚI ÎNTÂMPINATE ȘI REZOLVĂRI

În implementarea acestui proiect, au apărut mai multe dificultăți care au fost rezolvate astfel:

- **Identificarea utilizatorilor inactivi:** Utilizarea comenzii `last` pentru a obține data ultimei sesiuni a utilizatorilor a fost uneori problematică, deoarece utilizatorii inactivi nu aveau informații complete de login. Am gestionat acest lucru prin verificarea erorilor și crearea unui fișier `lastlogin` care indică fie data ultimei sesiuni, fie un mesaj de eroare.

- **Condiții de sincronizare:** Dacă scriptul detectează un utilizator activ într-un moment dat, dar acesta devine inactiv într-un ciclu următor, a existat riscul ca informațiile să fie inconsistent actualizate sau afișate greșit. Pentru a rezolva această problemă, am implementat un mecanism de actualizare periodică folosind comanda sleep, care a asigurat o actualizare corectă la intervale fixe, fără a afecta fluxul general de execuție. De asemenea, am introdus un mecanism de verificare a statusului fiecărui utilizator activ într-o manieră care să permită adaptarea la modificările din sistem.
- **Gestionarea fișierelor de ieșire:** Fiecare utilizator are un director dedicat în care sunt stocate informațiile sale. În cazul utilizatorilor inactivi, am golit fișierul procs și am creat fișierul lastlogin. Acest lucru a fost realizat printr-o verificare a utilizatorilor activi și actualizarea fișierelor corespunzătoare.

## **5. REZULTATE EXPERIMENTALE**

Am testat scriptul pe un sistem Linux și am obținut următoarele rezultate:

- **Test 1: Monitorizarea utilizatorilor activi:**

Obiectiv: Testarea capacității scriptului de a detecta utilizatorii activi și de a crea directoarele corespunzătoare acestora.

Descriere: Scriptul a fost executat într-un mediu cu mai mulți utilizatori autentificați simultan. Am verificat dacă scriptul poate identifica corect utilizatorii activi și să creeze directoarele în /tmp/active\_users.

Pentru fiecare utilizator activ, scriptul a creat un director dedicat în /tmp/active\_users/, iar fișierul procs a fost generat corect cu informațiile despre procesele curente ale utilizatorului, folosind comanda ps.

În toate cazurile, fișierul procs conținea informațiile corecte și complete despre procesele fiecărui utilizator activ.

Exemplu: Dacă utilizatorul „user1” era activ, scriptul crea directorul /tmp/active\_users/user1, iar fișierul procs conținea o listă a proceselor curente ale acestuia.

Concluzie: Testul a fost de succes. Scriptul a detectat corect utilizatorii activi și a creat directoarele și fișierele necesare.

```
andra@andra-VirtualBox:~/Desktop/itbi$ ./tema.sh
- Actualizarea informațiilor despre utilizatori -
Utilizatorii activi detectați: andra
Procesăm utilizatorul: andra
Fișierul 'procs' pentru andra a fost actualizat.
Procesele curente ale utilizatorului andra:
  PID CMD
  2901 /usr/lib/systemd/systemd --user
  2905 (sd-pam)
  2918 /usr/bin/pipewire
  2919 /usr/bin/pipewire -c filter-chain.conf
  2931 /usr/bin/wireplumber
  2932 /usr/bin/pipewire-pulse
  2936 /usr/bin/dbus-daemon --session --address=systemd: --nofork --nopidfile -
-systemd-activation --syslog-only
  2944 /usr/bin/gnome-keyring-daemon --foreground --components=pkcs11,secrets -
-control-directory=/run/user/1000/keyring
  2978 /usr/libexec/xdg-document-portal
  2988 /usr/libexec/xdg-permission-store
  3008 /usr/libexec/gdm-wayland-session env GNOME_SHELL_SESSION_MODE=ubuntu /us
r/bin/gnome-session --session=ubuntu
  3016 /usr/libexec/gnome-session-binary --session=ubuntu
  3092 /usr/libexec/gcr-ssh-agent --base-dir /run/user/1000/gcr
  3094 /usr/libexec/gnome-session-ctl --monitor
  3110 /usr/libexec/gvfsd
  3124 /usr/libexec/gvfsd-fuse /run/user/1000/gvfs -f
  3128 /usr/libexec/gnome-session-binary --systemd-service --session=ubuntu
  3165 /usr/bin/gnome-shell
  3166 /usr/libexec/at-spi-bus-launcher --launch-immediately
  3179 /usr/bin/dbus-daemon --config-file=/usr/share/defaults/at-spi2/accessibi
lity.conf --nofork --print-address 11 --address=unix:path=/run/user/1000/at-spi/
bus
  3241 /usr/libexec/at-spi2-registryd --use-gnome-session
  3262 /usr/libexec/gnome-shell-calendar-server
  3271 /usr/libexec/evolution-source-registry
  3282 /usr/bin/gjs -m /usr/share/gnome-shell/org.gnome.Shell.Notifications
  3284 /usr/bin/ibus-daemon --panel disable
  3285 /usr/libexec/gsd-a11y-settings
```

```

Procesăm utilizatorul: maria
Fișierul 'procs' pentru maria a fost actualizat.
Procesele curente ale utilizatorului maria:
  PID CMD
  4677 /usr/lib/systemd/systemd --user
  4681 (sd-pam)
  4704 /usr/bin/pipewire
  4706 /usr/bin/pipewire -c filter-chain.conf
  4712 /usr/bin/wireplumber
  4716 /usr/bin/pipewire-pulse
  4719 /usr/bin/dbus-daemon --session --address=systemd: --nofork --nopidfile -
-systemd-activation --syslog-only
  4724 /usr/bin/gnome-keyring-daemon --foreground --components=pkcs11,secrets -
-control-directory=/run/user/1001/keyring
  4775 /usr/libexec/xdg-document-portal
  4776 /usr/libexec/gdm-wayland-session env GNOME_SHELL_SESSION_MODE=ubuntu /us
r/bin/gnome-session --session=ubuntu
  4788 /usr/libexec/xdg-permission-store
  4789 /usr/libexec/gnome-session-binary --session=ubuntu
  4913 /usr/libexec/gcr-ssh-agent --base-dir /run/user/1001/gcr
  4916 /usr/libexec/gnome-session-ctl --monitor
  4932 /usr/libexec/gvfsd
  4941 /usr/libexec/gvfsd-fuse /run/user/1001/gvfs -f
  4944 /usr/libexec/gnome-session-binary --systemd-service --session=ubuntu
  4979 /usr/libexec/at-spi-bus-launcher --launch-immediately
  4991 /usr/bin/dbus-daemon --config-file=/usr/share/defaults/at-spi2/accessibi
lity.conf --nofork --print-address 11 --address=unix:path=/run/user/1001/at-spi/
bus
  4992 /usr/bin/gnome-shell
  5061 /usr/libexec/at-spi2-registryd --use-gnome-session
  5066 /usr/bin/gjs -m /usr/share/gnome-shell/org.gnome.Shell.Screencast
  5071 /usr/libexec/gnome-shell-calendar-server
  5083 /usr/libexec/evolution-source-registry
  5093 /usr/bin/gjs -m /usr/share/gnome-shell/org.gnome.Shell.Notifications
  5110 /usr/bin/ibus-daemon --panel disable
  5113 /usr/libexec/gsd-a11y-settings
  5117 /usr/libexec/goa-daemon
  5118 /usr/libexec/gsd-color

```

- **Test 2: Gestionarea utilizatorilor inactivi:**

Obiectiv: Verificarea funcționării corecte a scriptului atunci când un utilizator devine inactiv, prin golirea fișierului procs și actualizarea fișierului lastlogin cu data ultimei sesiuni de login.

Descriere: Am întrerupt activitatea unui utilizator (prin deconectarea acestuia) și am observat comportamentul scriptului. După deconectare, am verificat dacă fișierul procs era golit și dacă fișierul lastlogin conținea data ultimei sesiuni. Când un utilizator devenea inactiv, scriptul a detectat acest lucru și a golit corect fișierul procs.

Fișierul lastlogin a fost creat și actualizat cu data ultimei sesiuni a utilizatorului, obținută prin comanda last.

În cazul în care utilizatorul nu avea sesiuni recente, fișierul lastlogin conținea un mesaj corespunzător („Data ultimei sesiuni nu este disponibilă pentru [utilizator]”).

Exemplu: Dacă utilizatorul „user2” devine inactiv, fișierul procs din /tmp/active\_users/user2/ a fost golit, iar fișierul lastlogin a fost actualizat cu data ultimei sesiuni, precum „Sat Jan 5 14:23:14 2025”.

Concluzie: Testul a fost un succes. Scriptul a gestionat corect utilizatorii inactivi și a actualizat fișierele în mod corespunzător.

## **6. CONCLUZII ȘI ÎMBUNĂTĂȚIRI**

Proiectul a demonstrat cu succes că este posibilă implementarea unei soluții eficiente pentru monitorizarea utilizatorilor activi și inactivi într-un sistem Linux folosind un script shell simplu. În urma testării și experimentării cu diferite scenarii, am ajuns la următoarele concluzii principale:

-Monitorizarea utilizatorilor activi și inactivi:

Scriptul a funcționat corect în detectarea utilizatorilor activi și a proceselor lor curente, creând directoare dedicate pentru fiecare utilizator activ și actualizând periodic informațiile din fișierele procs.

De asemenea, gestionarea utilizatorilor inactivi a fost eficientă: scriptul a golit fișierele procs pentru aceștia și a creat fișierele lastlogin, conținând data ultimei sesiuni sau un mesaj de eroare dacă informațiile nu erau disponibile.

-Ușurința utilizării și extensibilitatea:

Scriptul a fost ușor de implementat și utilizat, iar logica sa simplă și modulară permite adăugarea de funcționalități suplimentare sau ajustarea comportamentului său pentru a satisface cerințele specifice ale unui sistem sau utilizator.

În ansamblu, soluția propusă a demonstrat eficiență și utilitate în monitorizarea și gestionarea utilizatorilor într-un sistem Linux, îmbunătățind securitatea și performanța sistemului prin furnizarea de informații actualizate despre utilizatori și procese.

Grigore Bianca

Chirițoiu Andra