

Raport asupra proiectului "UserFS"

Descrierea Problemei:

Proiectul ales de noi este UserFS, acesta constă în scrierea unui program shell care creează un director rădăcină, pe care noi l-am numit "root", în care se află câte un subdirector pentru fiecare utilizator denumit după numele acestuia. În fiecare subdirector, dacă utilizatorul este activ trebuie să se afle un fișier "procs" care conține procesele sale, iar dacă acesta a devenit inactiv să se afle un fișier "procs" care este gol și un fișier "lastlogin" care conține data și ora ultimului său login.

Soluția adoptată:

Programul scris de noi verifică dacă directorul "root" există pe Desktop, iar, dacă nu există, îl creează.

```
test -d ~/Desktop/root || mkdir ~/Desktop/root
```

Apoi parcurge lista de utilizatori activi returnată de comanda "users".

```
for user in `users`; do  
    [...]  
done
```

Scriptul verifică dacă fiecare utilizator din listă are un director, iar dacă nu are îi creează unul.

```
test -d ~/Desktop/root/$user || mkdir ~/Desktop/root/$user
```

După aceea, în interiorul directorului utilizatorului creează fișierul "procs" în care scrie rezultatul comenzii "ps" (process status) pentru acel utilizator.

```
ps -u "$user" > ~/Desktop/root/$user/procs
```

Și, înainte de a părăsi structura repetitivă, verifică dacă există fișierul "lastlogin" (acesta este creat ulterior, după găsirea utilizatorilor care au devenit inactivi).

```
rm -f ~/Desktop/root/$user/lastlogin
```

După ce termină parcurgerea listei de utilizatori activi, programul parcurge lista de directoare create în numele utilizatorilor.

```
for user in $(ls ~/Desktop/root); do  
    [...]  
done
```

În interiorul structurii repetitive, scriptul verifică, pentru fiecare subdirector al rădăcinii, dacă utilizatorul pe care îl reprezintă este încă activ (adică dacă apare activ în rezultatul comenzii "who").

```
if ! who | awk '{print $1}' | grep -q "^$user$"; then  
    [...]  
fi
```

Dacă utilizatorul este inactiv, atunci data și ora ultimului login sunt obținute dintr-o secvență a rezultatului comenzii "lastlog" și sunt stocate în fișierul "lastlogin".

După aceea conținutul fișierului "procs" este suprascris de ":".

```
lastlog -u "$user" | awk 'NR==2 {print $4" "$5" "$6" "$7" "$8" "$9}' >
~/Desktop/root/$user/lastlogin
: > ~/Desktop/root/$user/procs
```

După ieșirea din blocul condițional scriptul stagnează 30 de secunde, după care continuă de la început, întrucât tot codul precedent se află într-o structură repetitivă care rulează până când este întreruptă manual.

```
while true; do
[... ]
sleep 30
done
```

Dificultăți întâlnite:

Inițial, când încercam să testăm programul, am încercat să folosim comanda "su" (switch user) pentru a comuta către alt utilizator, însă, nici comanda "who", nici comanda "users", nu returnau numele utilizatorului nou, ci numele utilizatorului care a deschis terminalul.

De asemenea, când foloseam "su", iar apoi "lastlog" pentru a afla data și ora exactă a ultimului login al utilizatorului respectiv, comanda returna " **Never logged in** ". Astfel, am aflat că dacă folosim comanda "su", utilizatorul la care am comutat nu este considerat activ de către sistem.

Am surmontat această problemă folosind comanda "ssh <user>@localhost", unde <user> este numele utilizatorului către care voiam să comutăm. Astfel, utilizatorul este considerat ca fiind activ de către sistem, iar toate cele 3 comenzi menționate mai sus funcționează corect.

O altă dificultate întâlnită de noi a fost faptul că nu știam cum să facem ca programul să se actualizeze periodic odată la 30 de secunde.

Inițial ne-am gândit ca programul să se autoapeleze la sfârșit, folosind ". /UserFS.sh", însă nu știam cum să adăugăm pauza de 30 de secunde.

Apoi, ne-am dat seama că în loc să facem programul să se autoapeleze, este mai simplu să îl punem într-o buclă infinită (while true), iar la sfârșitul buclei respective să adăugăm comanda "sleep 30", care stagnează programul timp de 30 de secunde, înainte de a intra în următoarea iterație a buclei infinite.

Rezultate experimentale:

Rezultatul pentru scrierea outputului comenzii lastlog neparsat în "lastlogin":

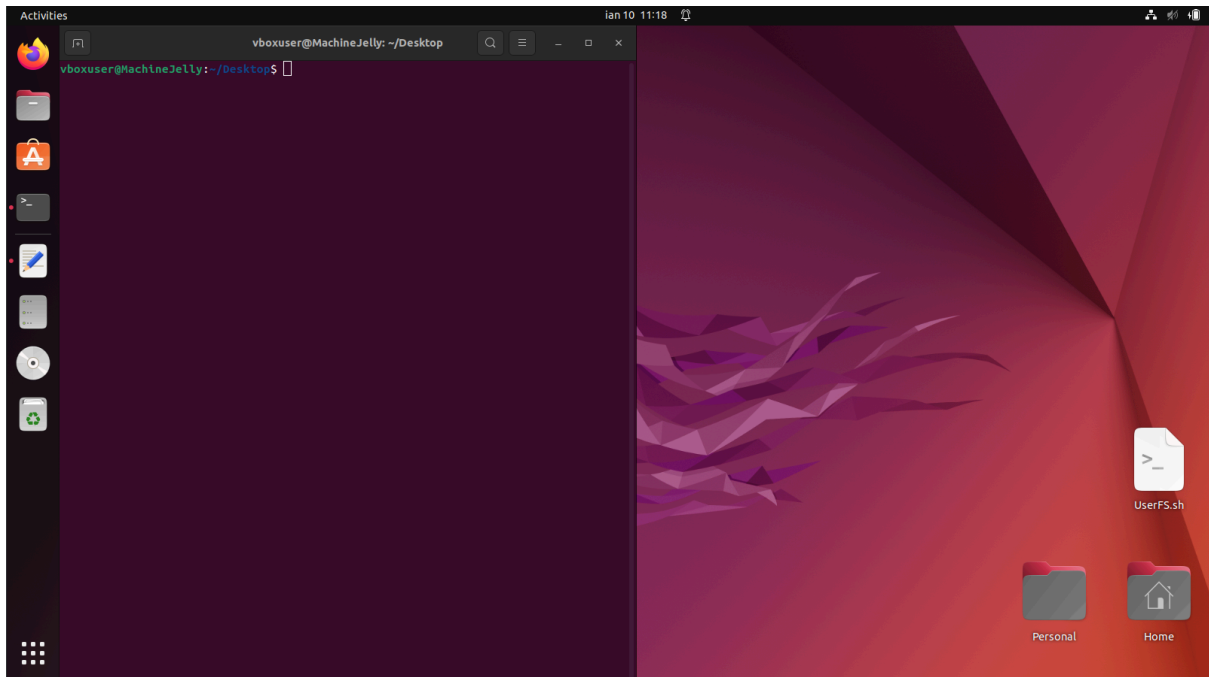
Username	Port	From	Latest
denis	pts/2	127.0.0.1	Vi ian 10 11:07:04 +0200 2025

Outputul de mai sus ne-a ajutat sa realizam că trebuie parsat.

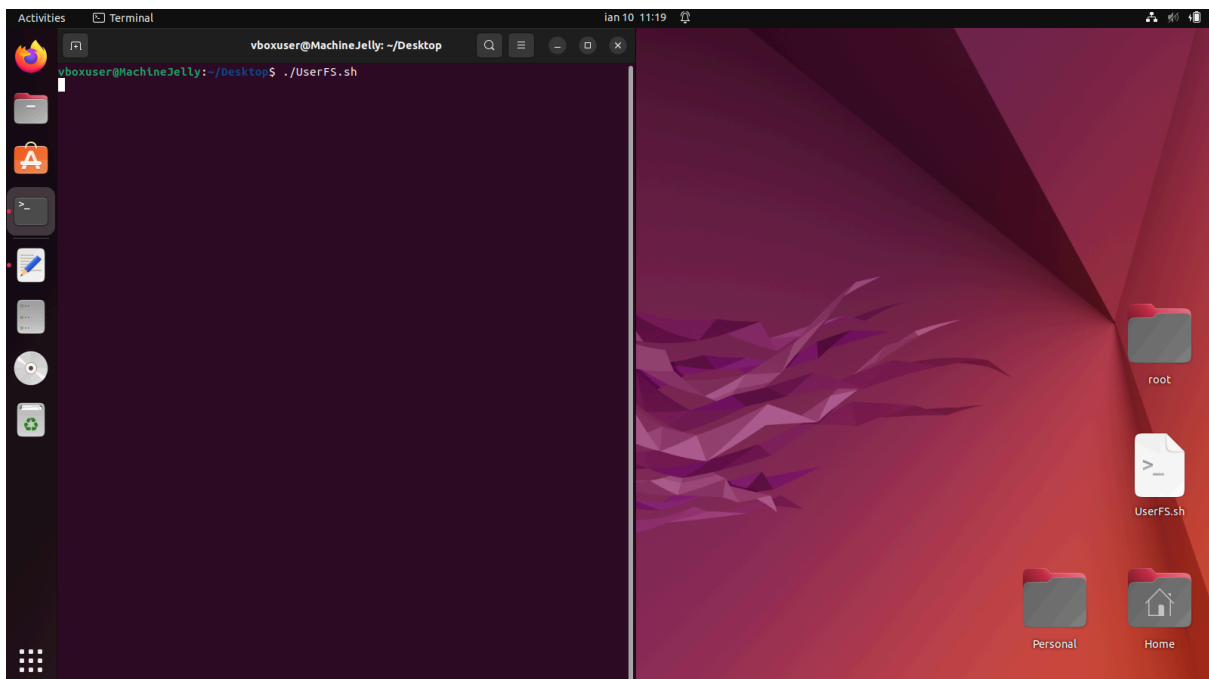
Rezultatul după parsare, folosind "awk":

Vi ian 10 11:09:54 +0200 2025

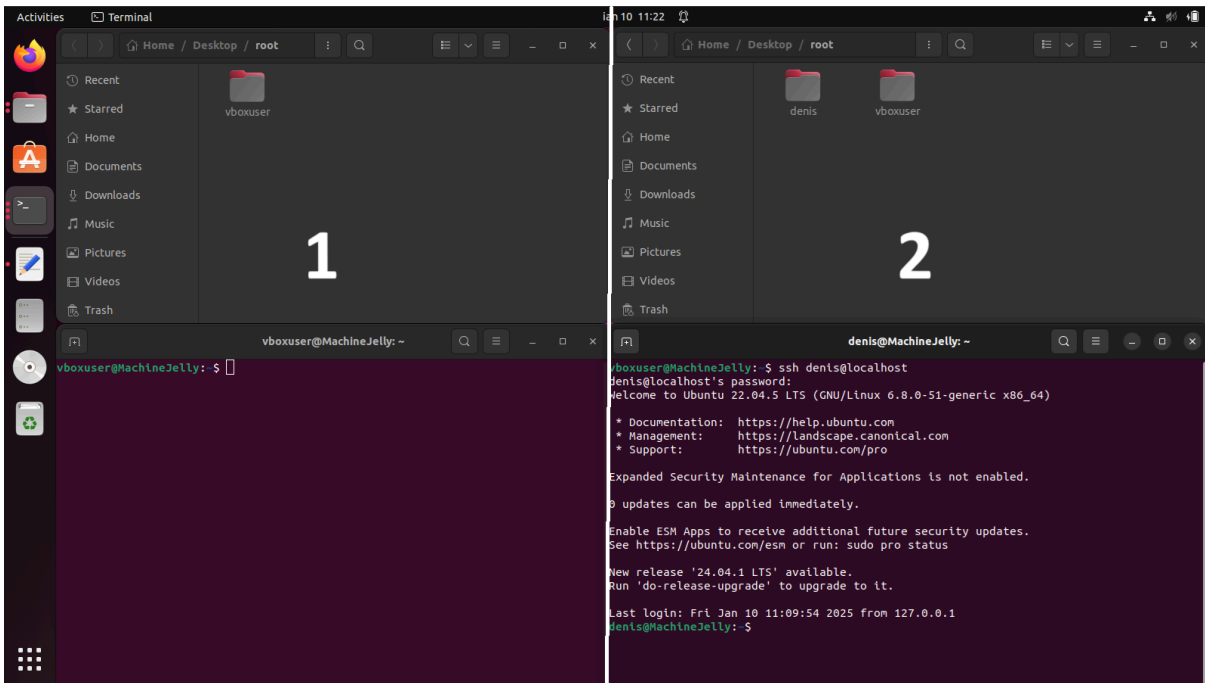
Desktopul înainte de a rula programul:



Desktopul după ce este rulat programul:



Conținutul directorului "root" înainte(1) și după(2) ce dă log in un nou utilizator (denis) folosind "ssh".



Directorul "denis" și conținutul fișierelor din interiorul său când utilizatorul este(1) activ și când nu(2) este activ.

