

Utilizarea Sistemelor de Operare: Tema 5

Termen de predare: Miercuri, 16 Ianuarie 2013, ora 23:55

Descriere

Tema constă în realizarea unui set de task-uri asupra unui sistem de operare Linux, gata instalat într-o mașină virtuală.

Pentru predarea temei sunt necesare:

- conexiunea la internet

Termen de predare:

- Timpul de lucru efectiv este de 10 de zile
- Uploadarea codului: 16 ianuarie 2013, ora 23:55
- Uploadarea se face pe formularul de la adresa următoare:
<https://docs.google.com/spreadsheet/viewform?formkey=dF9zaENLR0J0cEJHaW9jUGtIazZhREE6MA>

Deadline-ul temei este hard. **NU** se acceptă întârzieri după 16 ianuarie, ora 23:55.

Atenție! Important!

Pentru a putea rula `uso submit` cu succes `uso check` **NU** trebuie să se blocheze. **NU** oferim suport pentru eșecul de a trimite tema dacă `uso check` se blochează. Verificați comenzile și scripturile folosite pentru a nu bloca verificarea!

Atenție!

Chiar și după ce ați predat codul, **NU** ștergeți arhiva cu mașina virtuală până nu vă aflați nota! Dacă vor exista probleme în corectarea temei, fără arhivă **NU** se pot face contestații sau rezolva eventualele probleme apărute.

Testare

Testarea temei se va face **DOAR** autentificat ca utilizator **root**, folosind programul instalat **uso**. Programul **uso** implementează testele pentru toate task-urile din această temă. Acesta poate fi executat în orice moment pentru a verifica dacă au fost rezolvate corect task-urile.

Versiunea programului check este afișată pe prima linie a output-ului.

Versiunea curentă este: **v6**. Codename: **So long and thanks for the fish**.

În cazul în care testele vor fi actualizate, modificările ajung în mod automat pe mașinile virtuale. Comanda *uso update* va descărca ultima versiune a programului *uso* folosit pentru testarea temei. Pentru a actualiza manual checker-ul, puteți apela *uso update* ca utilizator **root**.

Testarea finală a temei se realizează prin intermediul utilitarului automat de verificare a temelor. Tot ce trebuie voi să faceți este să rulați **uso submit** și să încărcați codul în formularul descris în pagina anterioară.

Pentru a ușura testarea, recomandăm:

- Setările mașinii virtuale **NU** ar trebui modificate (nici măcar o actualizare a VMWare Tools). Dacă totuși considerați că este necesară o modificare întrebați întâi pe forum-ul temei dacă este acceptată.
- Mașina virtuală să pornească iar sistemul de operare să se încarce complet **fără interacțiune din exterior**
- Parola de root să rămână cea implicită ("student")

ATENȚIE: Mașina virtuală vine implicit cu aceste condiții îndeplinite și nici un task nu va afecta vreuna din ele. Vă rugăm să **NU** alterați vreuna din aceste componente deoarece **riscați să primiți 0 puncte pe temă**.

Notare

Nota pe întreaga temă este dată de punctajul acumulat pe toate task-urile, punctaj afișat de către programul check la execuția fără nici un parametru plus 10 puncte suplimentare oferite pe coding-style.

Se pot lua maxim **150** de puncte pe întreaga temă. Acest punctaj este echivalent cu **0.5** puncte din nota finală.

Nu este obligatorie rezolvarea tuturor task-urilor. Task-urile pot fi rezolvate în orice ordine, mai puțin în situația în care un task depinde de rezolvarea unui alt task.

Task 1 (140p)

Manager de pachete

Ca administrator IT într-o companie trebuie să afii informații despre pachetele instalate pe diverse servere. Pentru tema actuală, va trebui să construiești un wrapper peste `dpkg` și `apt` (cu toate derivatele lor) astfel încât să poți afla:

- detalii despre pachete
- pachete instalate în sistem
- pachete ce pot fi instalate în sistem
- fișiere aparținând unor pachete
- pachete ce dețin anumite fișiere

Atenție Scriptul nu va instala sau șterge pachete!

Mai exact, trebuie să construiești un script `pkgman.sh` aflat în `/home/student` care să accepte următoarele moduri de apelare:

- `pkgman.sh help` - afișează un mesaj de help cu argumentele acceptate
- `pkgman.sh packages` - afișează detalii despre toate pachetele din sistem: număr pachete, pachete virtuale, pachete lipsă
- `pkgman.sh names` - afișează toate numele pachetelor ce pot fi instalate (nu sunt sortate!)
- `pkgman.sh names PREFIX` - reduce lista anterioară doar la pachetele ce încep cu `PREFIX`
- `pkgman.sh names -c` - sumarizează lista, afișând doar numărul intrărilor
- `pkgman.sh names PREFIX -c` sau `pkgman.sh names -c PREFIX` - combină cele două efecte anterioare
- `pkgman.sh installed` - afișează toate numele pachetelor ce pot fi instalate, sortate alfabetic
- `pkgman.sh installed PATTERN` - reduce lista anterioară doar la pachetele ce se potrivesc cu șablonul `PATTERN`
- `pkgman.sh installed -c` - sumarizează lista, afișând doar numărul intrărilor
- `pkgman.sh installed PATTERN -c` sau `pkgman.sh installed -c PATTERN` - combină cele două efecte anterioare
- `pkgman.sh files PACKAGE` afișează toate fișierele deținute de un pachet precum și directoarele ce le conțin
- `pkgman.sh pkgof PATTERN` - afișează în ordine alfabetică toate pachetele ce dețin fișiere sau directoare ce se potrivesc cu șablonul `PATTERN` și se află în ierarhia de fișiere folosită de utilitarele de pachete
- `pkgman.sh info PACKAGE` - afișează informații despre un pachet. Lista acestor informații poate fi restrânsă prin selectorii. În cazul în care selectorii lipsesc se consideră că au fost pasați toți. Selectorii pe care trebuie să-i suportați sunt:

- `--maintainer` - selectează maintanerul pachetului
- `--priority` - selectează prioritatea pachetului
- `--section` - selectează secțiunea pachetului
- `--homepage` - selectează homepage-ul pachetului
- `--deb` - afișează numele fișierului `.deb` asociat pachetului.
- `--md5` - afișează MD5-ul acestui fișier
- `--version` - afișează versiunea pachetului
- `--tags` - afișează tag-urile pachetului
- `--provides`, `--suggests`, `--recommends`, `--recommends`, `--depends`, `--conflicts`, `--breaks` - afișează (dacă există) secțiunile corespunzătoare din descrierea pachetului
- `--vcs` - afișează cele 2 URL-uri pentru versionarea codului pachetului (dacă există): repository și interfața web
- `--srcs` - afișează sursele din care se compileaza pachetul

Pot exista mai multe apariții ale aceluiași selector. Fiecare selector trebuie "executat" în ordinea în care apare.

În caz de eroare de apel trebuie afișat mesaj corespunzător și întors cod de eroare 1. Dacă un packet nu există, se va întoarce cod de eroare 2 și mesajul corespunzător.

Sintaxa mesajelor este `script: mesaj argumente`. Pentru fiecare task în care este necesar, puteți vedea mesajul ce ar fi trebuit afișat rulând `uso check`.

Deoarece mesajul de help este mai complex, la rularea `uso check`, dacă nu există scriptul, se va crea un schelet ce va conține o porțiune din liniile ce trebuiesc afișate la help. Rămâne să faceți voi scriptul executabil și să-l modificați pentru celelalte taskuri.

Scriptul de check rulează teste pentru fiecare combinație posibilă și pentru o multitudine de pachete, fișiere și șabloane. Deoarece sunt multe teste, eșuarea unuia presupune saltul la primul test din următoarea categorie:

```
Test [001] ..... FAIL (pkgman.sh )
Test [003] ..... FAIL (pkgman.sh help)
Test [008] ..... FAIL (pkgman.sh packages)
Test [013] ..... FAIL (pkgman.sh files)
Test [021] ..... FAIL (pkgman.sh pkgof)
Test [029] ..... FAIL (pkgman.sh badparam)
Test [031] ..... FAIL (pkgman.sh names)
Test [057] ..... FAIL (pkgman.sh installed)
Test [083] ..... FAIL (pkgman.sh info pkgnf)
TOTAL: 0p
```

În cazul unui test picat se afișează în paranteză un detaliu al modului prin care testul apela scriptul. În plus, se afișează lista de diferențe între cele două outputuri: liniile cu `---` reprezintă liniile dorite iar cele cu `+++` liniile corespunzătoare obținute de scriptul vostru. Desigur, în cazul testelor cu output mare, vor fi mai greu de vizualizat diferențele. Dacă testul se termină cu succes se afișeaza punctajul asociat (în majoritatea cazurilor 1p, există doar un test de 5p și 4 de 2p)

```
Test [098] ..... PASS ( 1p)
--- desired
+++ obtained
@@ -2 +2 @@
-MD5: 6d4edb50cb33fdec61a3ae9c7d9ec202
+pkgman.sh: unknown info flag --md5
Test eşuat la pachetul git
Test [099] ..... FAIL (pkgman.sh info --md5 pkg)
TOTAL: 98p
```

Recomandarea este să nu opriți procesul de check pe parcurs. În majoritatea cazurilor nici nu veți putea dar dacă ați reuși riscați să stricați structura internă, ceea ce ar cauza eşuarea tuturor testelor următoare (până la ștergere scriptului și reluarea de la 0).

La pornirea procesului de test, scriptul verifică existența în sistem a tuturor pachetelor necesare pentru rulare. Astfel, erorile de scripting ce cauzează ștergerea unor pachete sunt evitate.

Lista schimbărilor