**Limbajul Shell scripting Unix/Linux**

Un **script shell** este un program scris într-un fișier text format din combinații de comenzi și instrucțiuni specifice unui interpretor de comenzi (shell).

În domeniul informaticii, **shell** este principala interfață de comunicare între utilizator și sistemul de operare. Shell-ul este necesar pentru a invoca sau a executa diferite programe disponibile pe calculator. Cel mai folosit shell este bash. Împreună cu kernelul, shell este una dintre componentele principale ale unui sistem de operare.

În funcție de tipul interfeței, shell poate fi cu interfață text (CLI) sau interfață grafică (GUI).

Shell-uri cu **interfață grafică** sunt Windows Explorer sau medii desktop construite peste X Window System. Shell-urile grafice permit operațiuni precum: deschiderea, închiderea, mutarea, redimensionarea ferestrelor, făcând un sistem de operare mai ușor de accesat.

Shell-uri cu **interfață text** (interpretoare de comenzi) sunt cmd.exe (Windows NT, XP), Windows PowerShell (Windows Vista, Server 2008, 7), Bash (Bourne Again Shell), csh (C Shell). Acesta e un program cu interfață în linie de comandă și rulează în interiorul unui terminal text. La introducerea unei comenzi, programul executat interacționează cu utilizatorul sau afișează date pe terminal.

Noțiunea de shell scripting se aplică shell-urilor cu interfață text. GNU Bash este shell-ul implicit pe majoritatea distribuțiilor Linux. Acesta se folosește și pe Mac OS X.

Utilizarea principal a unui shell script derive din faptul că acesta reprezintă o combinație de comenzi deja existente, deci posibilitatea de automatizare. Un script shell poate descrie sarcini repetitive, deci ori de câte ori este necesară executarea unei saricini, se poate rula un script shell.

**Expresii regulate**

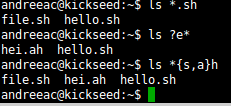
Pentru căutarea și selectarea de linii din fișiere text folosim comanda grep care folosește, la rândul ei, expresii regulate.

Este recomandat ca argumentul de tip expresie regulată transmis comenzii grep să fie plasat între apostrofuri (') pentru a fi escapat. În felul acesta caracterele din expresia regulată vor fi transmise întocmai comenzii grep și nu vor fi interpretate de shell.

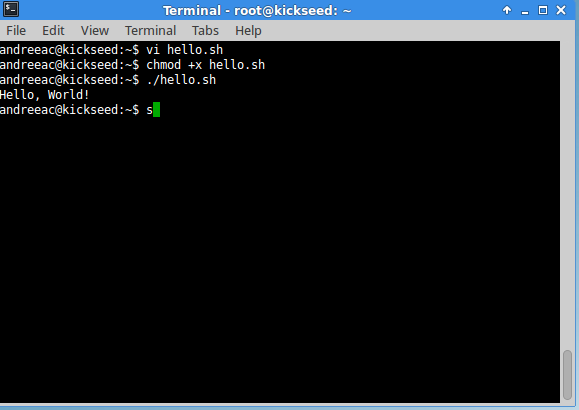
În expresiile regulate obișnuite (basic regular expressions), anumite caractere își pierd rolul lor special. Este cazul caracterului +, dar și al caracterelor ?, {, (, ) și |. Pentru a-și păstra rolul special, acestea trebuie precedate de backslash în construcții de forma \+, \?, \{, \(, \) și \|.

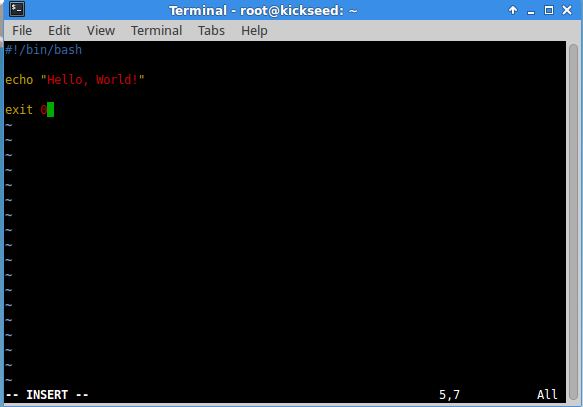
* se potriveste cu orice sir de caracter, inclusiv sirul vid
* ? se potriveste cu un singur caracter
* [...] se potriveste cu unul din caracterele din set; poate fi de genul [abc] sau [a-gA-G] sau [0-5h-zH-Z]
* {sir1,sir2,sir3,..} - se potrivește cu unul dintre șirurile din acolade
* ?(lista\_sabloane) - se potrivește cu o apariție sau cu nici una a unuia dintre șabloane
* \*(lista\_sabloane) - se potrivește cu nici o apariție sau mai multe a unuia dintre șabloane
* +(lista\_sabloane) - se potriveste cu o apariție sau mai multe a unuia dintre șabloane
* @(lista\_sabloane) - se potrivește cu exact un șablon din listă
* !(lista\_sabloane) - se potrivește cu toate șabloanele din listă, mai puțin unul

Pentru pattern-urile de mai sus trebuie activată opțiunea shopt –s extglob.



**Un script shell simplu**

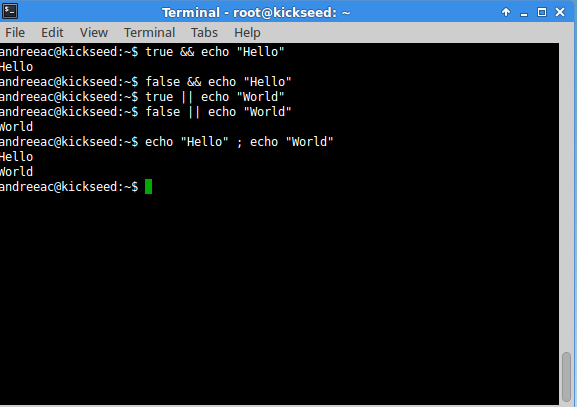


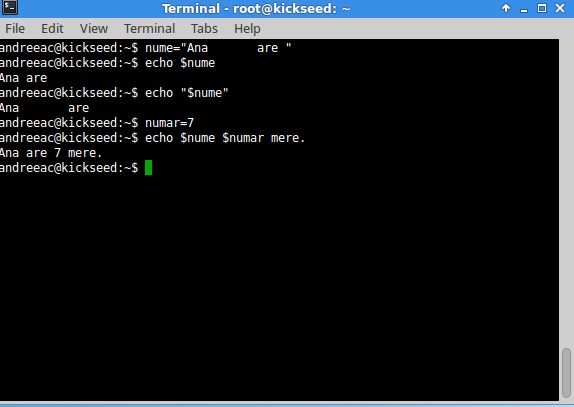


* Şirul #! de la începutul fişierului poartă denumirea de shebang. Acesta indică sistemului ce program va fi invocat pentru a interpreta scriptul.
* Este necesar ca fișierul să fie executabil pentru a putea fi interpretat.
* Caracterul # semnifică începutul unui comentariu care durează pană la sfarşitul liniei.
* Comanda exit este folosită pentru a indica valoarea de return a scriptului, implicit 0.

**Concatenarea comenzilor**

* comanda1 **;** comanda2 - comenzile sunt executate una după alta
* comanda1 **&&** comanda2 – a doua comandă este executată numai dacă prima are valoare de return 0
* comanda1 **||** comanda2 - a doua comandă este executată numai dacă prima are valoare de retur diferită de 0



**Variabile Bash**

Se foloseste sintaxa **nume\_variabilă=valoare**. Nu se lasă niciun spațiu în jurul semnului =, fiind restricție sintactică bash.

În cazul unei variabile nedefinte, aceasta nu e considerată eroare și va fi înlocuită cu șirul vid.

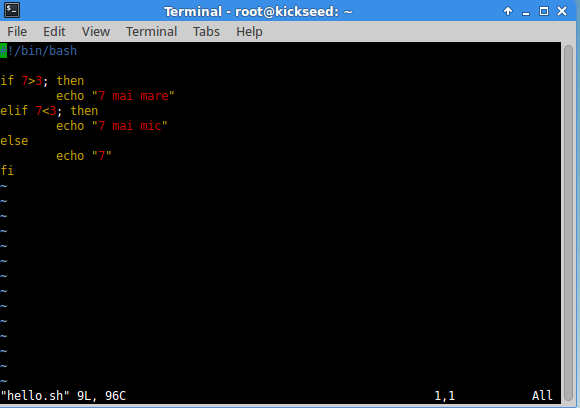
De asemenea de poate observa că spațiile în plus nu au niciun efect. Pentru a repara acest lucru trebuie să se încadreze numele variabilelor între ghilimele. Inconveniența este cauzată de faptul că echo afișează fiecare parametru al său, separat printr-un singur spațiu.

**Variabile speciale**

|  |  |
| --- | --- |
| $# | Numărul de parametri transmiși scriptului |
| $@ | Parametrii efectivi transmiși scriptului |
| $0 | Numele scriptului |
| $1 | Primul argument |
| $2 | Al doilea argument |
| $$ | PID-ul procesului curent |
| $! | PID-ul ultimului process lansat în background |
| $? | Valoarea de exit a ultimei comenzi |

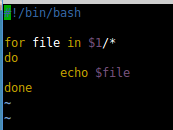
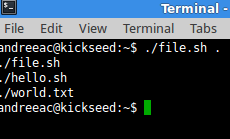
**Structuri de control**

* **IF**

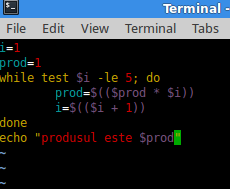


Elif și else sunt opționale.

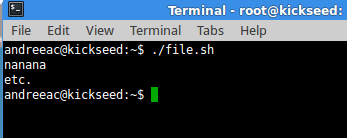
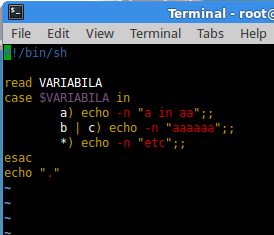
* **FOR**

În bash există și construcții de tipul for.  

* **WHILE**



* **CASE**



**Comenzi utile**

**echo** – Comanda echo exemplificată mai sus este folosită pentru a afișa un șir de caractere, o variabilă la ieșire standard.

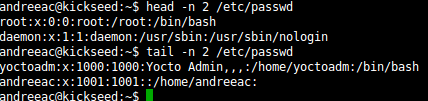
**cat -** afişează conţinutul unui fişier sau al intrării standard

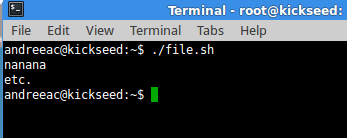
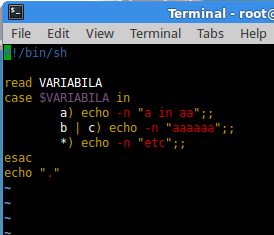
**tac** - afişează conţinutul unui fişier inversat

**head** - afişează începutul unui fişier sau al intrării standard *head [-n lines] files*

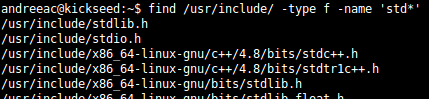
**tail** - afişează sfârşitul unui fișier sau al intrării standard *tail [-n lines] files*

Primul argument, daca există, afișează primele, respectiv ultimele n linii din text. Lipsa acestuia impune n = 10.



**read** - este folosită pentru citirea de informaţii de la intrarea standard 

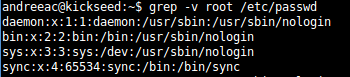
**find** - comandă fundamentală pentru parcurgerea intrărilor dintr-o ierarhie a sistemului de fişiere



**grep** - permite localizarea liniilor într-un fișier care conține o expresie căutată

Sintaxa: grep word file

O opțiune utilă este -v. Aceasta permite căutarea acelor linii care NU conțin cuvântul transmis ca parametru



Când nu se precizează un fișier se folosește intrarea standard, devenind ideal pentru lucrul cu pipes



De multe ori dorim să realizăm căutarea în mod case-insensitive. Pentru aceasta folosim opțiunea –i

Opțiunea -n permite afișarea numărului liniei care conținea cuvântul căutat.

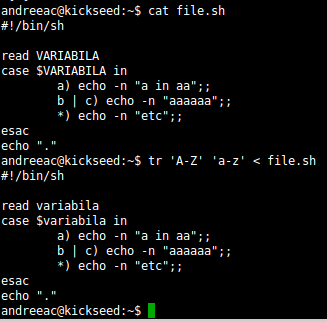
Putem de asemenea să afișăm numai fișierele care conțin acel cuvant (fără afișarea liniilor). Pentru aceasta folosim opțiunea -l (listare). De obicei este folosită în conjuncție cu opțiunea -R pentru căutarea recursivă în cadrul unei structuri de directoare

**tr** - folosită pentru a translata caracterele dintr-un set de caractere într-un alt set de caractere

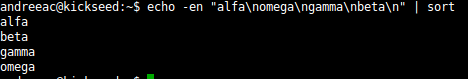
*tr 'set1' 'set2'*

În cazul în care setul set2 are mai puține caractere decât setul set1 acestea vor fi multiplicate pentru a ajunge la aceeași dimensiune.

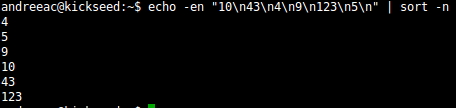
Caracterele [ și ] sunt escapate folosind \. Dacă dorim să translatam literele mari în litere mici:



**sort** - e utilizată pentru sortarea liniilor alfabetic.

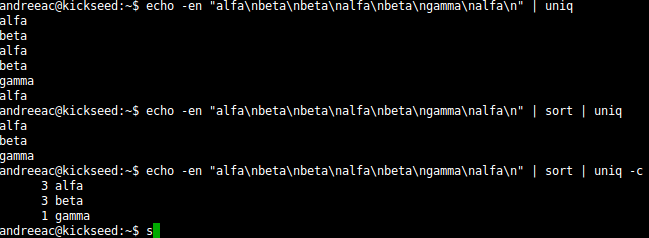


Sau pentru a sorta după valoarea numerică a șirurilor, pentru asta folosindu-se –n



Dacă output-ul apare într-o formă în care elementele de sortat sunt într-o altă coloană, se poate folosi optiunea -k (key) *sort -k start, end file*

**uniq** - Se pot elimina duplicatele folosind opțiunea -u la sort. Pentru a afișa de câte ori apare un cuvânt vom folosi comanda uniq cu optiunea -c. Se face eliminarea duplicatelor numai daca liniile sunt sortate



**wc** - folosită pentru a contoriza numărul de linii, de cuvinte sau de caractere dintr-un text sau de la intrarea standard. Pentru aceasta i se pot specifica opțiunile -c, -w, -l.



**cut** - selectează numai anumite părți (coloane) ale fișierului de intrare sau ale intrării standard

*cut –d delim –f field* Folosindu-se delimitatorul delim se vor selecta numai campurile fields.

