SISTEME DE OPERARE

- Laborator 3 -

PROGRAMARE SHELL

1. NOȚIUNI INTRODUCTIVE

* shell = un program special care furnizează o interfaţă între utilizator şi nucleul sistemului de operare
* shell-uri: *sh (Bourne shell)*, *csh (C shell)*, *ksh (Korn shell)*, *bash (GNU Bourne-again shell)*
* script = un fişier de tip text care conţine comenzi (interne sau externe)
* un prim exemplu de script bash:

|  |
| --- |
| #!/bin/bash  pwd  ls |

Observaţii:

* + secvenţa #! de pe prima linie a script-ului NU ESTE un comentariu (este numită *shebang*)
  + după această secvenţă este definită calea absolută către programul care trebuie rulat pentru toate celelalte linii ale script-ului
* rularea unui script bash:

chmod +x script\_1.sh

./script\_1.sh

* comentarii: încep cu caracterul #(hash)
* variabile:
  + numele unei variabile poate conţine litere, cifre și caracterul „ \_” *(underscore)*
  + primul caracter trebuie să fie o literă
  + cuvintele rezervate nu pot fi utilizate ca nume de variabile
  + shell-ul face deosebire între litere mici și litere mari *(case-sensitive)*
  + exemple:

n=45

name=Ana

msg="Enter a number:"

* cuvinte rezervate *(keywords)*:

if then else elif fi

for while until do done

case in esac

* comenzi interne *(built-in commands)*:
* afişarea listei cu comenzile interne: help
* afişarea informaţiilor despre o comandă: help ***command***
* exemple: echo read printf test
* exemple de script-uri bash:
* citirea şi afişarea unui număr: script\_2.sh
* citirea şi afisarea unui şir de caractere: script\_3.sh
* variabile speciale:

|  |  |
| --- | --- |
| $0 | numele fişierului |
| $1,..., $9 | argumentele furnizate în linia de comandă |
| $# | numărul de argumente furnizate în linia de comandă |
| $\* | sirul de argumente |
| $@ | lista individuala a argumentelor |
| $? | codul de ieşire (exit status) al ultimei comenzi executate |
| $$ | PID-ul procesului curent |
| $! | PID-ul ultimei comenzi lansate în background |

* exemplu: script\_4.sh

2. EXPRESII ARITMETICE CU NUMERE ÎNTREGI

* variabilele shell sunt tratate, în mod implicit, ca şiruri de caractere
* exemplu: script\_5.sh

2.a. Comanda expr

|  |
| --- |
| **expr** *expresie* |

* evaluează şi afişează la ieşirea standard valoarea unei expresii aritmetice întregi
* operatori:

|  |  |
| --- | --- |
| + - \\* / % | Sumă, diferenţă, produs, cât sau modulo |
| = != | Comparaţii numerice: |
| \> \>= | - valoarea 1 dacă relaţia dintre s şi d este adevărată |
| \< \<= | - valoarea 0 în caz contrar |
| \( \) | Paranteze pentru definirea subexpresiilor |
| S \| D | valoarea S dacă S nu este nici NULL nici 0, valoarea D în caz contrar |
| S \& D | valoarea S dacă S cât şi D nu sunt nici NULL nici 0, 0 în caz contrar |
| length S | Lungimea şirului S |
| index S CHARS | Poziţia primei apariţii în S sau 0 (numerotarea începe de la 1) |
| substr S P L | Subşirul care începe în S pe poziţia P şi are lungimea L |

2.b. Comanda let

* evaluează şi afişează la ieşirea standard valoarea unei expresii aritmetice întregi
* operatori: ++ -- ! ~ \*\* \* / % + - << >> <= >= < > == != & ^ | && ||

2.c. Parantezele duble

* exemplu: script\_6.sh

3. COMANDA test

* sintaxă:

|  |
| --- |
| **test** *conditie* sau **[** *conditie* **]** |

* evaluează *conditie* şi returnează 0 dacă condiţia este adevărată sau o valoarea NENULĂ în caz contrar
* permite compararea numerelor întregi, compararea şirurilor de caractere sau opţiuni de verificare a fişierelor

3.a. Compararea numerelor întregi

* operatori relaţionali: -lt -le -eq -ne -ge -gt

3.b. Compararea şirurilor de caractere

|  |  |
| --- | --- |
| -z sir | Verifică dacă şirul are lungimea 0 |
| -n sir | Verifică dacă şirul are lungime nenulă |
| s1 = s2 | Verifică dacă cele două siruri sunt egale |
| s1 != s2 | Verifică dacă cele două siruri sunt diferite |

3.c. Opţiuni de verificare a fişierelor

|  |  |
| --- | --- |
| -e fisier | Verifică dacă fişierul există |
| -s fisier | Verifică dacă fişierul există şi are lungimea nenulă |
| -r fisier | Verifică dacă fişierul există şi dacă se poate citi |
| -w fisier | Verifică dacă fişierul există şi dacă se poate scrie |
| -x fisier | Verifică dacă fişierul există şi este executabil |
| -f fisier | Verifică dacă fişierul există şi este un fişier obişnuit |
| -d fisier | Verifică dacă fişierul există şi este un director |
| -L fisier | Verifică dacă fişierul există şi este o legătură simbolică |
| -p fisier | Verifică dacă fişierul există şi este un pipe |
| -c fisier | Verifică dacă fişierul există şi este un fişier special de tip caracter |
| -b fisier | Verifică dacă fişierul există şi este un fişier special de tip bloc |

4. IF/THEN/ELIF/ELSE/FI

* sintaxă:

|  |
| --- |
| **if** *condition*  **then**  statement(s) to be executed  **elif** *condition*  **then**  statement(s) to be executed  **elif** *condition*; **then**  statement(s) to be executed  **else**  statement(s) to be executed  **fi** |

* exemple: if\_1.sh, if\_2.sh

5. FOR/DO/DONE

* sintaxă:

|  |
| --- |
| **for** *var* **in** *list*  **do**  statement(s) to be executed  **done** |

* exemple: for\_1.sh, for\_2.sh, for\_3.sh, for\_4.sh, for\_5.sh
* specificarea unui pattern pentru numele unui fişier (*filename wildcards*)

|  |  |
| --- | --- |
| \* | Orice secvenţă de caractere, chiar şi vidă (nu primul punct din numele fişierului) |
| ? | Un singur caracter (nu primul punct din numele fişierului) |
| [abc] | Orice caracter care se găseşte în lista de caractere |
| [!abc] | Orice caracter care nu se găseşte în lista de caractere |

* exemplu:
* afişarea fişierelor care încep cu o literă şi au o extensie formată din exact 2 caractere: ls [a-zA-Z]\*.??

6. WHILE/DO/DONE, UNTIL/DO/DONE

* sintaxă:

|  |  |
| --- | --- |
| **while** *condition*  **do**  statement(s) to be executed  **done** | **until** *condition*  **do**  statement(s) to be executed  **done** |

* exemple: while\_1.sh, while\_2.sh, while\_3.sh, while\_4.sh, while\_5.sh

7. CASE/ESAC

* sintaxă:

|  |
| --- |
| **case** *var* **in**  ***pattern\_1*)**  statement(s) to be executed if***patern\_1*** is matched**;;**  ***pattern\_2*)**  statement(s) to be executed if***patern\_2*** is matched**;;**  **...**  **\*)**  default condition to be executed**;;**  **esac** |

* exemple: case\_1.sh

8. ALTE COMENZI UTILE

8.a. Comanda cut

cut -d: -f 1 /etc/passwd

cut -d ":" -f 1 /etc/passwd

who | cut -d " " -f 1

8.b. Comanda find

find . -type f -name "\*.sh"

find /tmp -type d -empty

8.c. Comanda shift

shift [n] deplasarea spre stânga cu n poziţii a argumentelor furnizate în linia de comandă

8.d. Comanda sleep

sleep [n] suspendarea execuţiei procesului curent pentru n secunde

8.e. Comanda exit

exit [n] terminarea execuţiei şi revenirea la procesul din care a fost lansat

REFERINȚE:

* Programare shell: https://ryanstutorials.net/bash-scripting-tutorial/