

Documentație și Utilizare

Alfirii Flavius

Introducere

Proiectul supranumit de noi TermiX a fost implementat si testat in mediul de programare Code Bloks, folosindu-se limbajul de programare C.

Comenzi Implementate

- **Comenzi standard :**

1. Cd (implementat)
2. Man (implementat)
3. Ls
4. Nano

Aceste 4 comenzi ajuta la functionalitatea celor 6 comenzi mai jos prezentate. Noi am implementat 2 dintre ele iar pe celalate le folosim gata implementate in terminalul OS-ulli.

1. **Cd:**

DESCRIPTION: Change the shell working directory.

cd .. (go to home space)

Ex.

cd 2016
2016 – Directory

2. **Man:**

DESCRIPTION: Print information about comand.

--help display this help and exit

--version output version information and exit

Ex.

man cd

3. **Ls:**

DESCRIPTION: show list directory contents.

Ex.

ls

4. **Nano:**

DESCRIPTION: create a file (.txt / .sh / .c / .awk / etc).

Ex.

nano namefile.type

• **Comenzile alese:**

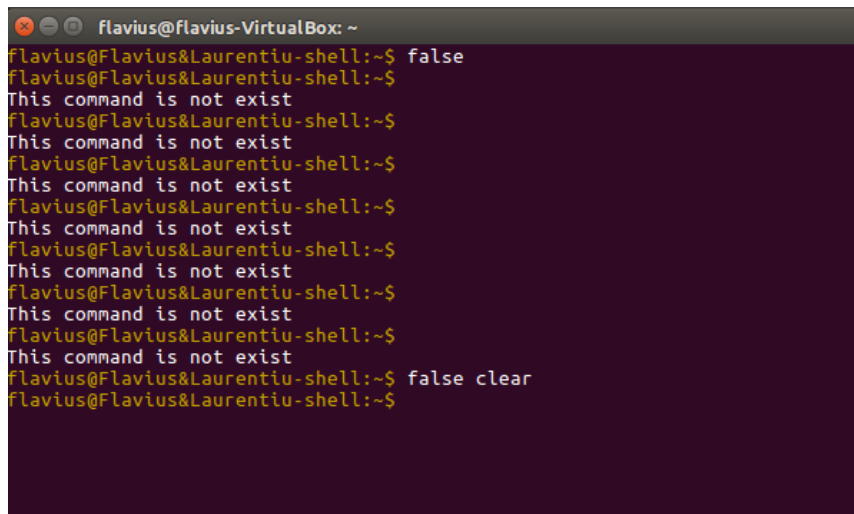
1. False
2. Cat
3. Logname
4. File
5. Expr
6. Clear

1. **False:**

DESCRIPTION: do nothing, (ignored command line arguments).

Ex.

false cd



```
flavius@flavius-VirtualBox: ~  
flavius@Flavius&Laurentiu-shell:~$ false  
flavius@Flavius&Laurentiu-shell:~$  
This command is not exist  
flavius@Flavius&Laurentiu-shell:~$  
This command is not exist  
flavius@Flavius&Laurentiu-shell:~$  
This command is not exist  
flavius@Flavius&Laurentiu-shell:~$  
This command is not exist  
flavius@Flavius&Laurentiu-shell:~$  
This command is not exist  
flavius@Flavius&Laurentiu-shell:~$  
This command is not exist  
flavius@Flavius&Laurentiu-shell:~$  
This command is not exist  
flavius@Flavius&Laurentiu-shell:~$ false clear  
flavius@Flavius&Laurentiu-shell:~$
```

2. **Cat:**

DESCRIPTION: concatenate files and print.

-c, -- counts lines and print the result

-b, -- put before each line (1,2,3,...,n)

-d, -- put after each line „,\$”

Ex.

cat first_file.txt second.sh

```
cat -c first_file.txt second.sh
```

```
flavius@Flavius-Laurentiu-shell:~$ cat -b ss.txt lab6.txt
1      a
2      b
3      c
4      d
5      e
6
7      f
8      g
9      1
10
11     2
12     2
13     123456
14     abcdefg
15     7891
16     ZXVNM
17     ZXVNM
flavius@Flavius-Laurentiu-shell:~$
```

3. Logname:

DESCRIPTION: print user's login name.

Ex.

logname (print user)

```
flavius@Flavius-VirtualBox: ~
flavius@Flavius-Laurentiu-shell:~$
This command is not exist
flavius@Flavius-Laurentiu-shell:~$
This command is not exist
flavius@Flavius-Laurentiu-shell:~$
This command is not exist
flavius@Flavius-Laurentiu-shell:~$
This command is not exist
flavius@Flavius-Laurentiu-shell:~$
This command is not exist
flavius@Flavius-Laurentiu-shell:~$ logname
No user logged on.
flavius@Flavius-Laurentiu-shell:~$
```

4. File:

DESCRIPTION: determine file type

-b, -- print file type without file_name

Ex.

file fisier.txt Result: fisier.txt : ASCII text

file -b fisier.txt Result: ASCII text

```

flavius@flavius-VirtualBox: ~
flavius@Flavius&Laurentiu-shell:~$ ls
2015          folder_1  Music      terminal
2016          folder_2  Pictures   terminal.c
AlfiriiFlavius_lab6.sh  l2.sh    Poze       terminal.o
Desktop       lab6.txt  Public     vbox
Documents    lab9.txt  ss.txt     Videos
Downloads    lalal    Templates  xxx
examples.desktop  lalax    term
flavius      magic.mgc termi.c
flavius@Flavius&Laurentiu-shell:~$
This command is not exist
flavius@Flavius&Laurentiu-shell:~$
This command is not exist
flavius@Flavius&Laurentiu-shell:~$ file l2.sh
l2.sh: shell script
flavius@Flavius&Laurentiu-shell:~$
This command is not exist
flavius@Flavius&Laurentiu-shell:~$ file 2016
2016: Directory
flavius@Flavius&Laurentiu-shell:~$

```

5. Expr:

DESCRIPTION: evaluate expresion

+, -- sum

-, -- decrease

*, // etc.

Ex.

expr 2 + 3 Result: 5

expr 3 > 2 Result: 1

```

flavius@flavius-VirtualBox: ~
flavius@Flavius&Laurentiu-shell:~$
This command is not exist
flavius@Flavius&Laurentiu-shell:~$
This command is not exist
flavius@Flavius&Laurentiu-shell:~$
This command is not exist
flavius@Flavius&Laurentiu-shell:~$
This command is not exist
flavius@Flavius&Laurentiu-shell:~$
This command is not exist
flavius@Flavius&Laurentiu-shell:~$ expr 4 + 5
9
flavius@Flavius&Laurentiu-shell:~$ expr 4 g 10
expr : syntax error
flavius@Flavius&Laurentiu-shell:~$

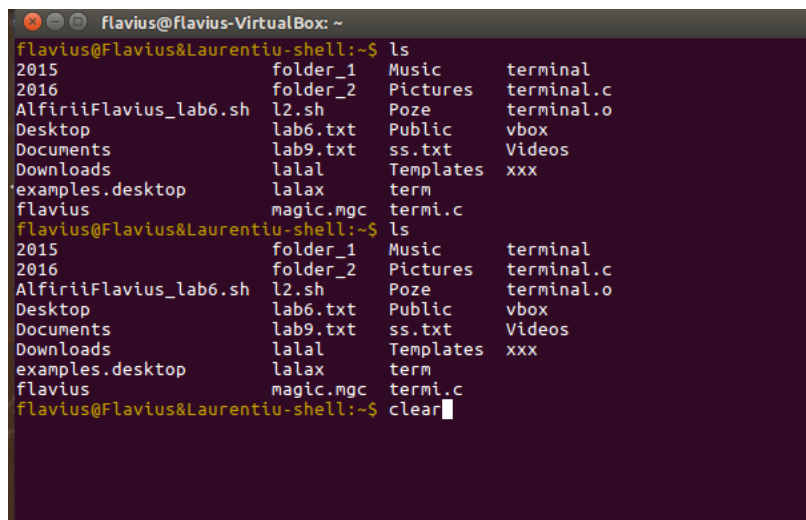
```

6. Clear:

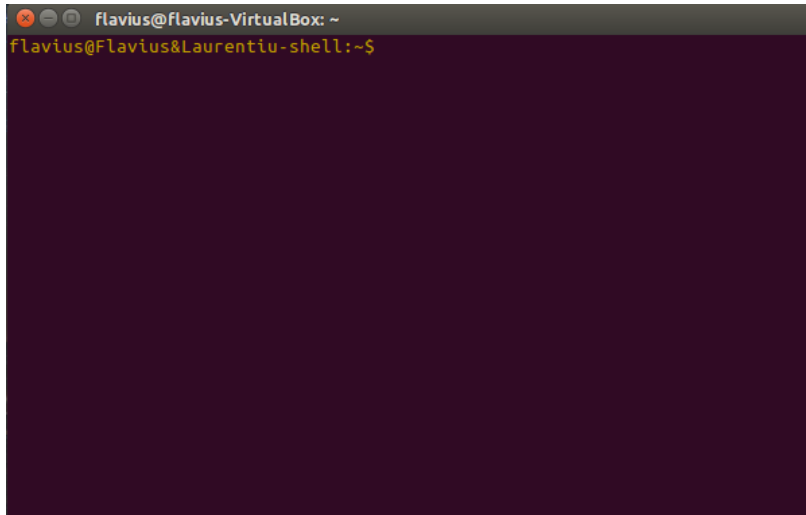
DESCRIPTION: clear the terminal screen

Ex.

Clear



```
flavius@flavius-VirtualBox: ~  
flavius@Flavius&Laurentiu-shell:~$ ls  
2015          folder_1  Music      terminal  
2016          folder_2  Pictures   terminal.c  
AlfiriiFlavius_lab6.sh l2.sh    Poze       terminal.o  
Desktop       lab6.txt  Public     vbox  
Documents     lab9.txt  ss.txt     Videos  
Downloads     lalal    Templates  xxx  
examples.desktop lalax    term  
flavius       magic.mgc termi.c  
flavius@Flavius&Laurentiu-shell:~$ ls  
2015          folder_1  Music      terminal  
2016          folder_2  Pictures   terminal.c  
AlfiriiFlavius_lab6.sh l2.sh    Poze       terminal.o  
Desktop       lab6.txt  Public     vbox  
Documents     lab9.txt  ss.txt     Videos  
Downloads     lalal    Templates  xxx  
examples.desktop lalax    term  
flavius       magic.mgc termi.c  
flavius@Flavius&Laurentiu-shell:~$ clear
```



Prezentare implementare

1. False:

Comanda false ignora orice alta comanda sau simbol introdus dupa.

Mod de implementare: am construit o functie de tip void() care realizeaza in interior un proces dar in terminal nu se intampla nimic.

2. Cat:

La cat folosesc un pointer char in care se memoreaza pe rand, byte cu byte informatia din cele doua fisiere. Fisiere deschise cu functia open() si fopen().

3. Logname:

Pentru logname am folosit o functie in care am apelat functia getlogin() din biblioteca #include<unistd.h>.

4. File:

La comanda file in primul rand am verificat daca este director sau fisier. Daca este director si exista returneaza mesajul specific, iar daca este fisier am cautat sa vad daca exista in directorul cu pricina si dupa am verificat sa vad de ce tip este.

5. Expr:

Aceasta functie realizeaza operatii logice, aritmetice si relationali intre operanzi de de tip int (cifre).

Se preiau parametri ca si sir de caractere iar apoi sunt prelucrati diferit: cifrele sunt transformate in int cu functia atoi() iar semnele sunt cautate cu functia strstr() in sirul de semne creat implicit in functie iar apoi comparam cu strcmp() sa vedem ce operatie o sa aplicam pe cele doua int-uri(cifre).

6. Clear:

Elibereaza casuta, iar pentru asta s-a folosit ----- \033[2J\033[1;1H, care semnifica:

- a. 33 is the octal of ESC
- b. 2J is for clearing the entire console/terminal screen (and moves cursor to upper left on DOS ANSI.SYS)
- c. 1;1H moves the cursor to row 1 and column 1

