



## Cuprins

1	Interpretorul de comenzi .....	1
2	Comenzi relative la directoare și fișiere.....	3
2.1	Nume de fișiere .....	3
2.2	Comenzi pentru gestiunea directoarelor.....	4
2.2.1	Definirea unui disc logic .....	5
3	Comenzi MS-DOS. Fișiere de comenzi .....	6
3.1	Comenzi MS-DOS .....	6
3.1.1	Comenzile Date și Time.....	6
3.1.2	Comenzile dir și tree .....	6
3.1.3	Comanda type .....	7
3.1.4	Comanda echo.....	8
3.1.5	Comanda for.....	8
3.1.6	Etichete .....	8
3.1.7	Comanda goto .....	9
3.1.8	Comanda if.....	9
3.1.9	Comanda exit .....	10
3.2	Fișiere de comenzi.....	10
3.2.1	Apelarea unui fișier de comenzi cu parametri .....	10
3.2.2	Comanda rem .....	10
3.2.3	Comanda call .....	10
3.2.4	Variabile sistem .....	11
3.2.5	Comanda set.....	11
3.3	Lansarea aplicațiilor din linia de comandă.....	12
4	Teme .....	12
5	Intrebări de control.....	15

## 1 Interpretorul de comenzi

Orice sistem de operare are un program ce interpretează și execută comenzi, numit shell. El este o interfață între sistemul de operare și utilizator. În cazul sistemului de operare Windows aceste comenzi se execută într-o fereastră DOS. Pentru a deschide o fereastră DOS se execută un click pe opțiunea

Run
-----

a meniului Start

(Windows + R) .
-----------------

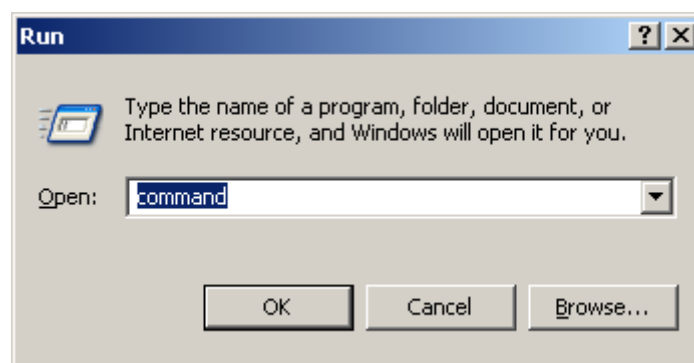
În caseta de dialog care se deschide se tastează opțiunea

command,
----------

ca în Figura 1, sau

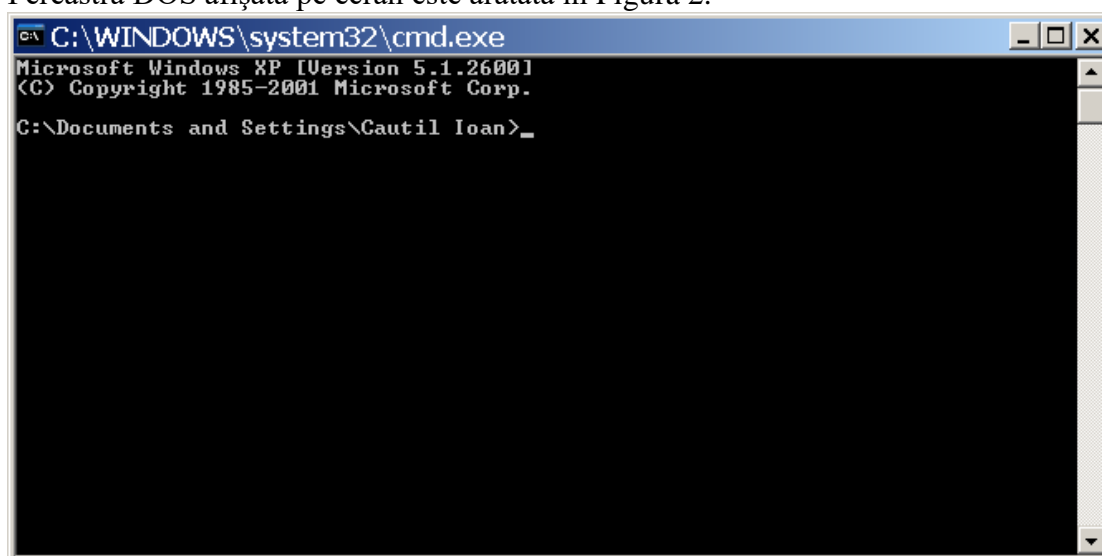
cmd,
------

ceea ce duce la lansarea interpretorului de comenzi.



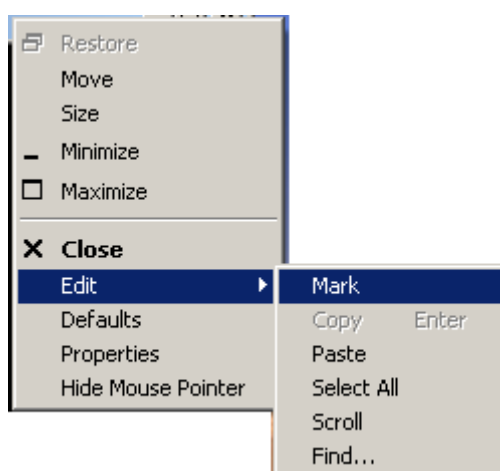
**Figura 1. Caseta de dialog pentru deschiderea unei ferestre DOS.**

Fereastra DOS afișată pe ecran este arătată în Figura 2.



**Figura 2. Fereastra DOS.**

La un click de dreapta pe bara de titlu a ferestrei sau în bara de activități se afișează meniul contextual din Figura 3 cu care se pot stabili proprietățile ferestrei, se pot copia comenzi în fereastră, etc.



**Figura 3. Meniul contextual al ferestrei DOS.**



Menționăm că, interpretorul de comenzi nu face deosebire între litere mari și mici. Comenzile pot fi scrise cu litere mari sau mici.

## 2 Comenzi relative la directoare și fișiere

În sistemul de fișiere Windows și MS-DOS unitățile de disc fizice sunt asociate cu unități de disc logice. Unitățile logice sunt referite printr-o literă urmate de caracterul **:**. De exemplu, unitățile logice asociate discului dur sunt referite ca **c:**, **d:**, unitatea de citit / scris discuri compacte este referită ca **e:**, etc.

Directoarele sunt memorate la fel ca și fișierele. Eticheta unui fișier sau director are următoarea formă:

Nume fișier	Extensie	Atribute	Timp	Data	Numărul primului bloc din fișier	Dimensiune
-------------	----------	----------	------	------	----------------------------------	------------

### 2.1 Nume de fișiere

În sistemul de operare *MS-DOS* numele unui fișier are forma **nume.extensie**. Numele constă din cel mult 8 caractere iar extensia din maxim 3 caractere. Numele de fișiere *MS-DOS* nu pot conține caracterele **\ / : \* ? " < > |**. În sistemul de operare *Windows* se pot utiliza nume de fișiere lungi de până la 255 caractere. Pentru aceasta, orice fișier sau director are două nume: unul de 8+3 caractere pentru compatibilitate *MS-DOS* și unul lung. Dacă numele fișierului este lung, *Windows* crează un nume tip *MS-DOS* format din primele 6 caractere urmat de ~1, ~2, etc, iar extensia se trunchiază la 3 caractere. În cazul în care numele fișierului conține spații, acestea se elimină și se rețin primele 6 caractere urmate de ~1, etc. De exemplu, numele de director All Users, ce conține un spațiu, este transformat în AllUse~1. Numele de fișiere duble care rezultă din transformarea numelor lungi în cele *MS-DOS* sunt urmate de ~2, ~3, etc. De exemplu, numele de director Documents se transformă în Docume~1, numele de director Documents1 se transformă în Docume~2, etc.

Menționăm că fișierele ce conțin programe executabile au extensia **exe**.

Sistemul de fișiere al sistemului de operare Windows este organizat sub forma unui arbore de directoare. Fiecare director poate conține alte directoare și fișiere. Discul dur al calculatorului este divizat în discuri logice ce au denumirile **c :**, **d :**, etc. Pe fiecare disc logic există un sistem de fișiere organizat sub forma unui arbore. Primul nivel al structurii arborescente este reprezentat de directorul rădăcină (Root), care este creat de sistemul de operare. El conține toate celelalte directoare și fișiere. Fiecare director conține la rândul lui alte directoare și fișiere. Directorul rădăcină este reprezentat prin **\**. Un fișier sau un director în ierarhie este specificat prin calea absolută de la directorul rădăcină. Ea constă din lista directoarelor traversate de la directorul rădăcină la directorul ce conține fișierul, cu componentele separate de **\**. **Calea absolută începe cu \**. În orice moment orice proces are un director curent de lucru. Calea de la directorul curent la un subdirector se numește cale relativă. **Calea relativă nu începe cu \**.

Fie de exemplu următoarea structură de directoare:

- Drx
  - dira
  - dirb
    - dirc



- dird

Calea **\Drx\dirb** este o cale absolută către directorul dirb pentru că începe cu \ (adică începe cu directorul rădăcină).

Directorul **dirc** poate fi specificat prin

- Calea absolută, care este **\Drx\dirb\dirc**
- Calea relativă la directorul părinte **dirb**, care este **dirc** (fără \)

## 2.2 Comenzi pentru gestiunea directoarelor

Sistemul de operare are funcții pentru lucrul cu directoare. Comenzile pentru lucrul cu directoare sunt cele din Tabelul 1.

Comanda	Forma generală
Crearea unui director	<i>mkdir nume</i> <i>md nume</i>
Ștergerea unui director	<i>rmdir nume</i> <i>rd nume</i>
Modificarea directorului curent	<i>chdir nume</i> <i>cd nume</i>
Copierea unui fișier	<i>copy dest sursa</i>
Redenumirea unui fișier	<i>rename nume</i> <i>ren nume</i>
Ștergerea unui fișier	<i>del nume</i>
Compararea a două fișiere	<i>fc nume1 nume2</i>
Mutarea unui fișier	<i>move sursa dest</i>
Listarea unui fișier	<i>type name</i>

**Tabelul 1. Comenzi pentru lucrul cu directoare**

În tabelul de mai sus numele unui fișier sau director poate conține calea relativă sau cea absolută. La crearea unui director sunt create automat două elemente cu numele . (punct) și .. (două puncte). Acestea sunt :

- directorul curent este desemnat prin . (punct),
- directorul părinte al directorului curent este desemnat prin .. (două puncte).

### Exemple:

#### 1. Comanda

```
cd ..
```

face ca directorul părinte să devină directorul curent.

#### 2. Comanda

```
cd \
```

face ca directorul rădăcină să devină directorul curent.

#### 3. Comanda

```
mkdir test
```

crează în directorul curent un subdirector cu numele test.



4. Trecerea de la un disc logic la altul se face specificând numele acestuia. De exemplu, comanda

```
d:
```

face ca directorul rădăcină al discului d: să devină noul director curent

### 2.2.1 Definirea unui disc logic

Este posibil să definim discuri logice. Orice literă, ce nu este utilizată pentru un disc fizic sau logic, poate fi utilizată pentru a defini un nou disc logic. Orice director sau disc fizic poate fi definit ca un disc logic.

Definirea de discuri logice este utilă

- pentru a prescurta scrierea unei căi către un fișier sau director,
- pentru a redefini o cale către un director sau fișier, în anumite aplicații, copiate de pe alte calculatoare, ce au căi predefinite către directoare sau fișiere, ce nu corespund cu cele de pe calculatorul curent,
- pentru a defini o cale ce va fi schimbată ulterior.

Comanda pentru a defini un disc logic care să corespundă unui director sau disc fizic este `subst` sau `substitute`.

Fie directorul **c:\files\docs** pe care vrem să-l referim ca **w:**. Vom scrie următoarea comandă

```
subst w: c:\files\docs.
```

După această comandă, fișierul **c:\files\docs\fis.doc** poate fi înlocuit cu **w:\fis.doc**.

Dacă directorul **c:\files\docs** a fost mutat la o nouă locație, de exemplu,

**d:\newlocfiles\docs**, vom scrie următoarea comandă

```
subst w: d:\newlocfiles\docs.
```

Definiția unui disc logic este valabilă până la ștergerea ei, cu comanda

```
subst w: /D.
```

Pentru a vedea discurile virtuale definite, utilizăm comanda

```
subst.
```

Menționăm că, dacă ștergem un fișier utilizând un disc logic definit cu comanda `subst`, fișierul este șters fizic, fără a fi mutat în directorul `Recycle.bin`.

### Exemple:

1. Fie un director L pe discul c:

- c:
- L

Vrem ca directorul L să fie definit ca discul logic L:. Pentru aceasta utilizăm comanda

```
subst L: c:\L.
```

Comanda

```
subst,
```

fără argumente, afișază definiția introdusă.



```
C:\ C:\WINDOWS\system32\command.com
Microsoft(R) Windows DOS
(C)Copyright Microsoft Corp 1990-2001.

C:\DOCUME~1\CAUTIL~1>subst
L:\: => C:\L

C:\DOCUME~1\CAUTIL~1>_
```

Figura 4. Utilizarea comenzii subst

### 3 Comenzi MS-DOS. Fișiere de comenzi

#### 3.1 Comenzi MS-DOS

Comenzile MS-DOS încep totdeauna pe o linie nouă și sunt urmate de caracterul **Enter**. Formatul unei comenzi este:

mnemonica parametri.

Vom prezenta în continuare comenzile cele mai utilizate ale sistemului de operare MS-DOS.

##### 3.1.1 Comenzile Date și Time

Aceste comenzi afișază data curentă sub forma luna/zi/an și timpul curent sub forma hh:mm:ss. Ele permit de asemenea modificarea datei curente și a timpului curent. Un exemplu de utilizare a acestor comenzi este arătat în Figura 5.

```
C:\ C:\WINDOWS\System32\command.com
C:\DOCUME~1>date
The current date is: Sun 11/11/2007
Enter the new date: <mm-dd-yy>

C:\DOCUME~1>time
The current time is: 10:53:57.06
Enter the new time:
```

Figura 5. Comenzile Date și Time

##### 3.1.2 Comenzile dir și tree

Aceste comenzi listează conținutul unui director, fișiere și subdirectoare, sub forma unei liste și respectiv sub forma unui arbore. Exemple de utilizare a acestor comenzi sunt prezentate în Figura 6 și în Figura 7.



```
C:\WINDOWS\System32\command.com
C:\DOCUME~1\IOANCA~1>tree
Folder PATH listing
Volume serial number is 71FAE346 5829:59C1
C:.
├── Desktop
├── Favorites
│   └── Links
├── My Documents
│   ├── My Music
│   ├── My Pictures
│   └── Updater5
├── Start Menu
│   └── Programs
│       ├── Accessories
│       ├── Accessibility
│       └── Entertainment
└── Startup
C:\DOCUME~1\IOANCA~1>
```

Figura 6. Arborele unui director afișat cu comanda tree

```
C:\WINDOWS\System32\command.com
C:\DOCUME~1>cd..
C:\>dir
Volume in drive C has no label.
Volume Serial Number is 5829-59C1

Directory of C:\

07/09/2005  11:20 AM    <DIR>          Background
03/26/2005  12:30 AM    <DIR>          0 CONFIG.SYS
11/28/2007  05:43 PM    <DIR>          Documents and Settings
12/01/2005  03:02 PM    <DIR>          Drivers
09/13/2007  06:14 AM    <DIR>          283 jpb.exe
03/10/2007  05:31 PM    <DIR>          NVIDIA
11/28/2007  10:02 PM    <DIR>          Program Files
10/12/2006  07:58 PM    <DIR>          129 Shortcut to 3½ Floppy (A).lnk
07/26/2005  09:48 PM    <DIR>          software
10/14/2005  08:13 PM    <DIR>          97 test
11/27/2007  08:20 AM    <DIR>          WINDOWS
10/07/2007  03:47 AM    <DIR>          1,777 wininit.exe
11/27/2007  08:18 AM    <DIR>          150 YServer.txt
        6 File(s)                2,436 bytes
        7 Dir(s)              2,059,448,320 bytes free
C:\>
```

Figura 7. Conținutul directorului rădăcină al discului c :

În cazul fișierelor, comanda dir afișează numele fișierului, dimensiunea în octeți și timpul și data ultimei modificări. Numele directoarelor sunt precedate de eticheta <DIR>. Comanda dir poate avea forma generală

```
dir [disc :] [cale] [nume] [opțiuni],
```

unde disc: este numele discului logic, c :, d :, etc., cale este calea relativă sau absolută, nume este numele directorului sau fișierului. Numele poate conține caracterele \* și ?. Caracterul ? specifică un caracter admisibil în acel loc, caracterul \* specifică oricâte caractere admisibile în acel loc.

### 3.1.3 Comanda type

Comanda

```
type
```

listează un fișier pe ecran. Forma comenzii este

```
type nume
```

unde nume este numele fișierului ce va fi listat.



### 3.1.4 Comanda echo

Comanda

```
echo
```

afisează un text pe ecran. Forma comenzii este:

```
echo text.
```

Comanda

```
echo
```

poate fi utilizată pentru a comanda afișarea sau suprimarea afișării comenzilor pe ecran.

Comanda

```
echo off
```

suprimă afișarea comenzilor următoare pe ecran.

Comanda

```
echo on
```

reia afișarea comenzilor următoare pe ecran.

Comanda

```
@echo off
```

suprimă atât afișarea comenzilor următoare pe ecran cât și a comenzii însăși.

### 3.1.5 Comanda for

Permite execuția repetată a unei comenzi. Forma comenzii într-o linie de comandă este

```
for %var in (mulțime de valori) do comandă,
```

iar într-un fișier de comenzi

```
for %%var in (mulțime de valori) do comandă.
```

Mulțimea de valori conține valori separate de spații sau virgule. Comanda se execută cu variabila var luând pe rând toate valorile mulțimii.

**Exemple:**

1. Să se listeze directoarele rădăcină de pe unitățile de disc a: și c:

```
for %x in (a:, c:) do dir %x
```

In exemplul de mai sus variabila x ia valorile a: și c :.

2. Fie trei fișiere file1.txt, file2.txt, file3.txt în directorul curent. Pentru a lista aceste trei fișiere utilizăm comanda

```
for %a in (file1.txt, file2.txt, file3.txt) do type %a
```

In exemplul de mai sus variabila a ia valorile: file1.txt, file2.txt, file3.txt.

### 3.1.6 Etichete

Pentru a transfera controlul execuției unui fișier de comenzi în diverse puncte utilizăm etichete. O etichetă are unu la opt caractere, litere sau cifre, primul caracter fiind o literă. O etichetă este scrisă singură pe o linie după caracterul:

**Exemple:**

- 1.

```
:labelx
```

este comanda ce declară eticheta labelx.





### 3.1.7 Comanda goto

Comanda **goto** are forma:

```
goto eticheta.
```

Interpretorul de comenzi trece la execuția comenzii din linia ce urmează după eticheta.

### 3.1.8 Comanda if

Permite execuția unei comenzi dacă o anumită condiție este adevărată. Forma comenzii este:

```
if [not] condiție comanda
```

sau

```
if [not] condiție (comanda) else (comanda).
```

În forma a doua a instrucțiunii argumentele comanda se scriu obligatoriu în paranteze. Condițiile ce se pot testa sunt:

- Existența unui fișier.

În acest caz comanda if are forma:

```
if [not] exist filename comanda
```

sau

```
if [not] exit filename (comanda) else (comanda).
```

În forma a doua a instrucțiunii argumentele comanda se scriu obligatoriu în paranteze.

#### Example:

Să testăm dacă în directorul curent există fișierul file.txt Pentru a afișa un mesaj putem utiliza una din cele două forme ale comenzii:

```
if exist file.txt (echo file.txt exists) else (echo
file.txt missing)
if not exist file.txt (echo file.txt missing) else (echo
file.txt exists)
```

- Egalitatea a două șiruri.

Testarea egalității a două șiruri se face cu următorii operatori:

$$sir1 \left\{ \begin{array}{c} EQU \\ NEQ \\ LSS \\ LEQ \\ GTR \\ GEQ \\ = \end{array} \right\} sir2$$

Forma comenzii if este în acest caz:

```
if [not] șir1 operator șir2 comanda
```

sau

```
if [not] șir1 operator șir2 (comanda) else (comanda).
```

În forma a doua a instrucțiunii argumentele comanda se scriu obligatoriu în paranteze.

#### Example.



1. Dacă un șir dintr-o mulțime de șiruri are o anumită valoare, vom afișa această valoare:

```
for %b in (a, b, c) do if "%b" == "a" echo %b
```

### 3.1.9 Comanda exit

Comanda

```
exit
```

duce la părăsirea interpretorului de comenzi. Forma comenzii este

```
exit.
```

## 3.2 Fișiere de comenzi

Interpretorul de comenzi are tastatura și monitorul ca fișiere de intrare și ieșire standard, dar este posibilă și execuția unor comenzi citite dintr-un fișier de comenzi. Fișierele cu comenzi au extensia bat.

### 3.2.1 Apelarea unui fișier de comenzi cu parametri

Un fișier de comenzi poate fi apelat cu unu până la nouă parametri, specificați după numele său. Parametrii sunt șiruri de caractere. Ei pot fi referiți în instrucțiuni ca %1, %2, ..., %9. Pentru a putea utiliza simplu acești parametri există comanda

```
shift
```

care deplasează valorile parametrilor. Astfel după o comandă *shift* avem: %1 primește valoarea %2, ..., %8 primește valoarea %9

Pentru a apela un fișier de comenzi cu parametri, linia de comandă are forma:

```
nume.bat parametrul1 parametru2 ...
```

### Exemple.

1. Să se afișeze valorile tuturor parametrilor cu care apelăm un fișier de comenzi. Pentru aceasta vom afișa repetat parametrul %1 și vom deplasa parametrii cu comanda *shift*.

```
:loop
    if "%1" == "" goto stop
    echo %1
    shift
    goto loop
:stop
    echo stop
```

### 3.2.2 Comanda rem

Forma comenzii este:

```
rem șir de caractere.
```

Comanda rem conține comentarii.

### 3.2.3 Comanda call

Comanda *call* apelează un fișier de comenzi. Forma ei este: **call nume fișier**

### Exemple.



1. Presupunem că avem două fișiere de comenzi `aaa.bat` și `bbb.bat` în același director și vrem să apelăm fișierul `aaa.bat` în interiorul fișierului `bbb.bat`.  
Fișierul `bbb.bat` va fi următorul:

```
echo apeleaza aaa.bat  
call aaa.bat  
echo s-a executat aaa.bat
```

### 3.2.4 Variabile sistem

Sistemul de operare are niște variabile predefinite numite variabile sistem, utilizate în scopuri speciale. Tabelul următor conține câteva dintre aceste variabile și semnificația lor.

Variabila	Semnificație
<code>PATH</code>	Indică sistemului de operare unde să caute programele pe care să le execute, dacă ele nu se află în directorul curent
<code>OS</code>	Conține numele sistemului de operare
<code>homepath</code>	Conține calea directorului curent
<code>username</code>	Conține numele calculatorului
<code>systemdrive</code>	Conține unitatea de disc pe care se află sistemul de operare
<code>systemroot</code>	Conține directorul în care se află sistemul de operare

**Tabelul 2. Variabile sistem**

În variabila `PATH` putem specifica mai multe directoare separate de caracterul `;`. Variabila `PATH` este importantă, astfel încât există instrucțiunea `PATH` pentru a da valori acestei variabile. De exemplu, comanda

```
PATH;
```

șterge valoarea variabilei `PATH`, iar comanda

```
PATH=valoare
```

atribuie valoare variabilei `PATH`.

### 3.2.5 Comanda set

În fișierele de comenzi putem defini noi variabile și putem atribui valori acestora cât și variabilelor sistem cu comanda `set`. Forma acestei comenzi este:

```
set nume=valoare
```

Pentru a utiliza valoarea unei variabile `numevar` vom scrie

```
%numevar%.
```

#### Exemple:

Să presupunem următoarea structură de directoare

- `C:`
  - Documents and Settings
  - Administrator
  - Users

Presupunem că în directorul *Administrator* avem fișierul de comenzi `aaa.bat`, iar în directorul *Users* avem fișierul de comenzi `bbb.bat`. În fișierul de comenzi `bbb.bat` avem comanda

```
call aaa.bat
```

care apelează fișierul `aaa.bat`.



Pentru execuția acestei comenzi este nevoie ca variabila *PATH* să conțină calea **C:\Docume~1\Administrator**. Pentru aceasta în fișierul de comenzi bbb.bat vom adăuga la variabila *PATH* această cale. Fișierul de comenzi bbb.bat va fi următorul:

```
rem salvam valoarea variabilei PATH
set PATH1=%PATH%
echo %PATH1%
rem adaugam calea c:\ Docume~1\Administrator la
variabila PATH
set PATH=%PATH%;c:\ Docume~1\Administrator
echo %PATH%
call aaa.bat
rem refacem vechea valoare a variabilei PATH
set PATH=%PATH1%
echo %PATH%
```

De remarcat în acest fișier faptul că, pentru a utiliza valoarea unei variabile numevar, am scris %numevar%. Dacă de exemplu am scrie în fișierul de mai sus

```
echo PATH
```

s-ar afișa șirul PATH și nu valoarea variabilei PATH.

### 3.3 Lansarea aplicațiilor din linia de comandă

Programele pot fi lansate din linia de comandă cu forma **numeaplicație parametri**.

Implicit, sistemul de operare caută programele în directorul

**c:\WINDOWS\System32**. Pentru a lansa aplicații memorate în alte directoare, trebuie scrisă calea absolută spre directorul în care este memorată aplicația. De exemplu, pentru a lansa aplicația Notepad vom scrie

```
Notepad.
```

Pentru a lansa aplicația Notepad care să deschidă un fișier existent vom scrie

```
Notepad numefișier,
```

unde numefișier va include și calea spre directorul fișierului.

## 4 Teme

În directorul My Documents se va crea un director nou cu numele exemple. În timpul lucrării vor fi afișate simultan pe ecran fereastra DOS și fereastra directorului exemple afișată în Windows Explorer, ca în Figura .

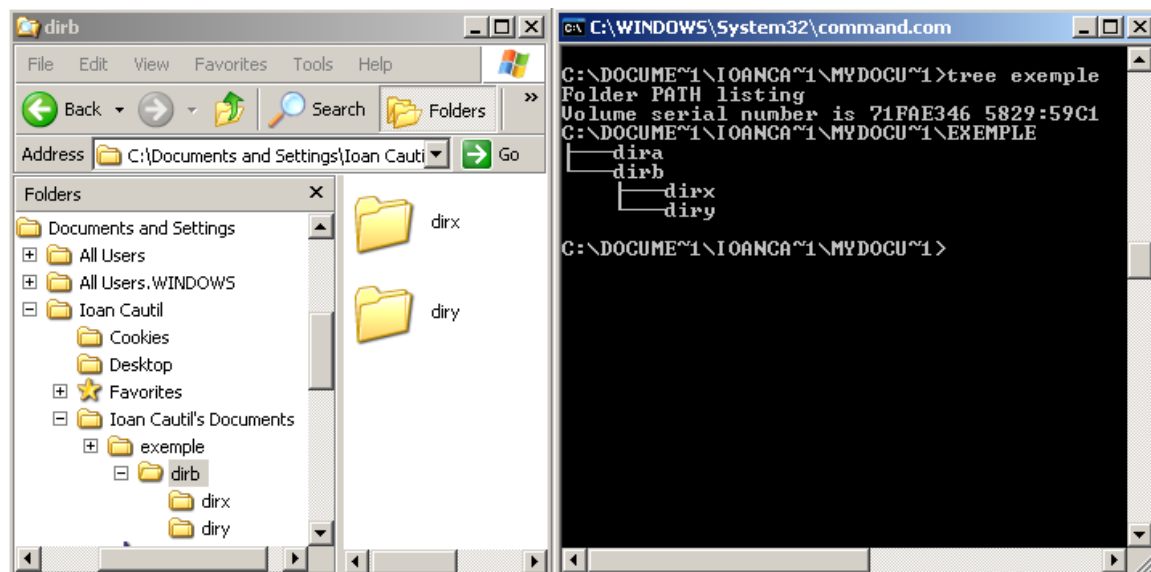


Figura 8. Fereastra Windows și DOS afișate pe ecran.

a) se va crea în fereastra DOS următoarea structură de directoare:

- exemple
  - dira
  - dirb
    - dirx
    - diry

Se vor identifica directoarele create în fereastra DOS și în fereastra directorului exemple afișată în Windows Explorer. Se va lista conținutul directorului exemple cu comenzile dir și tree.

b) se vor crea trei fișiere cu editorul Notepad în directorul dirx. Se vor lista aceste trei fișiere cu instrucțiunea for în fereastra DOS.

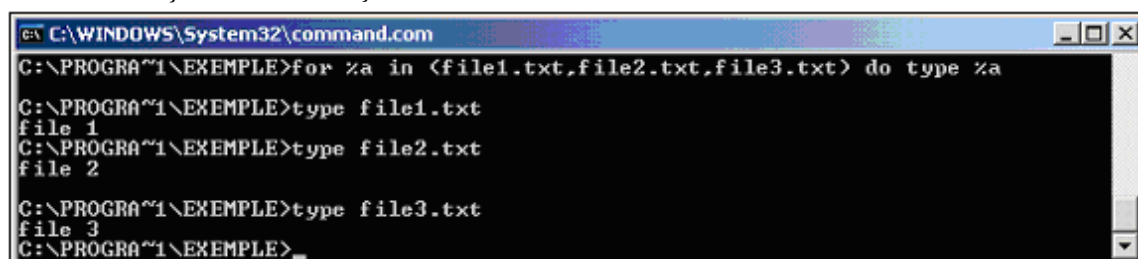


Figura 9. Listarea a trei fișiere

c) se va crea un fișier de comenzi pentru a rezolva problema de la punctul b.

Fișierul va conține :

- o instrucțiune rem ce va specifica funcția fișierului,
- o instrucțiune echo ce va afișa funcția fișierului pe ecran,
- o instrucțiune for ce va lista fișierele.



```
C:\WINDOWS\System32\command.com
C:\PROGRA~1\EXAMPLE>bat1
C:\PROGRA~1\EXAMPLE>ren Fisier de comenzi care listeaza trei fisiere
C:\PROGRA~1\EXAMPLE>echo --- listarea a trei fisiere ---
--- listarea a trei fisiere ---
C:\PROGRA~1\EXAMPLE>for %a in <file1.txt file2.txt file3.txt> do type %a
C:\PROGRA~1\EXAMPLE>type file1.txt
file 1
C:\PROGRA~1\EXAMPLE>type file2.txt
file 2
C:\PROGRA~1\EXAMPLE>type file3.txt
file 3
C:\PROGRA~1\EXAMPLE>
```

Figura 10. Fișier de comenzi pentru listarea a trei fișiere

- d) se va crea un fișier de comenzi pentru a rezolva problema de la punctul b ce va apela fișierul de comenzi de la punctul c.

```
C:\WINDOWS\System32\command.com
C:\PROGRA~1\EXAMPLE>bat2
C:\PROGRA~1\EXAMPLE>ren apelarea unui fisier bat
C:\PROGRA~1\EXAMPLE>echo --- apelarea unui fisier bat ---
--- apelarea unui fisier bat ---
C:\PROGRA~1\EXAMPLE>call bat1.bat
C:\PROGRA~1\EXAMPLE>ren Fisier de comenzi care listeaza trei fisiere
C:\PROGRA~1\EXAMPLE>echo --- listarea a trei fisiere ---
--- listarea a trei fisiere ---
C:\PROGRA~1\EXAMPLE>for %a in <file1.txt file2.txt file3.txt> do type %a
C:\PROGRA~1\EXAMPLE>type file1.txt
file 1
C:\PROGRA~1\EXAMPLE>type file2.txt
file 2
C:\PROGRA~1\EXAMPLE>type file3.txt
file 3
C:\PROGRA~1\EXAMPLE>
```

Figura 11. Fișier de comenzi ce apelează un alt fișier de comenzi

Fișierul va conține :

- o instrucțiune rem ce va specifica funcția fișierului,
- o instrucțiune echo ce va afișa funcția fișierului pe ecran,
- o instrucțiune call ce va apela fișierul de comenzi creat la punctul c.

- e) în directorul diry se vor crea 4 fișiere cu numele alfa.txt, alf.txt, alb.txt și beta.txt cu editorul Notepad.

- se va afișa conținutul directorului diry cu comanda dir,
- se vor afișa fișierele ce încep cu a cu comanda dir,
- se vor afișa fișierele ce încep cu b cu comanda dir,
- se vor afișa fișierele alf.txt și alb.txt cu comanda dir,
- se vor lista fișierele ce încep cu alf cu comanda type.

Pentru a afișa doar fișierele cerute vezi o lucrare anterioară.

- f) se va lansa aplicația Notepad de la linia de comandă și se va crea un fișier în directorul dirx. Se vor examina atributele fișierului exe al aplicației Notepad.



## 5 Intrebări de control

1. Cum se descriu numele fișierelor și directoarelor la căutarea fișierelor și directoarelor pe disc ?
2. Care este diferența între calea relativă și cea absolută la specificarea unui fișier?
3. Care este comanda de afișare a conținutului unui director?
4. Care sunt comenzile DOS pentru lucrul cu directoare ?
5. Care este forma comenzii for ?
6. Care este rolul fișierelor de comenzi ?