

METODE AVANSATE DE GESTIUNE A DOCUMENTELOR ȘI A SISTEMELOR DE CALCUL - CURS 2 -

Asist. Diana – Florina Șotropa

www.cs.ubbcluj.ro/~diana.sotropa

Cuprins

- Sistemul de Operare Windows.

SCRIPT-URI

- executarea unor grupuri de comenzi -

Argumente in linia de comanda

- Parametrii:
 - %0 = numele programului apelat
 - %1, %2, ...
 - %* = toate argumentele (%1 %2 ... %255)
 - Doar %1, %2, ... %9 pot fi adresate folosind numarul
- Extensii ale parametrilor care contin nume de fisiere:
 - %~f1 – full name
 - %~d1 – disk
 - %~p1 – path
 - %~n1 – nume fisier fara extensie
 - %~x1 – doar extensie fisier
 - %~s1 – nume scurt
 - %~1 – se elimina ghilimele
 - %~a1 – attributele
 - %~t1 – detalii temporale
 - %~z1 - dimensiune

Sintaxă: Operatori

- Operatori aritmetici
- Operatori relaționali
- Operatori logici
- Operatori de atribuire
- Operatori pe biți

Sintaxă:
Operatori
aritmetici

+

-

*

/

%% sau %

Sintaxă: Operatori de atribuire

`+=`

`--=`

`*=`

`/=`

`%%=` sau `%=`

Sintaxă: Operatori relaționali

EQU

NEQ

LSS

GTR

GEQ

Sintaxă: Operatori pe biți

& sau &=

| sau |=

^ sau ^=

!

~

<< sau <<=

>> sau >>=

Operatori

`command1 && command2`

⇒Daca `command1` s-a executat cu success atunci se executa `command 2`

`command1 & command2`

⇒Se executa `command1` si apoi se executa `command2`

`command1 || command2`

⇒Se executa `command2` doar daca `command1` nu se poate executa

Operatori Exemple

```
COPY H:\share\TNSnames.ora  
C:\Oracle\ && ECHO The Copy  
succeeded
```

⇒ Arata mesajul daca copierea s-a
efectuat cu success

```
(Dir C:\Test1 & Dir C:\Test2) ||  
Echo One or more of the commands  
failed.
```

⇒ Se pot folosi paranteze pentru a
se testa succesul mai multor
comenzi

Setarea variabilelor

- Variabilele setate cu SET isi pastreaza valoarea doar in sesiunea curenta

SET - afiseaza toate variabilele

SET variabila – afiseaza toate variabilele care se potrivesc

SET variabila = string

SET /A variabila="expresie" – variabila va contine inclusiv “

SET /A “variabila=expresie” – variabila nu contine “

SET “variabila=”

SET /P variabila=[promptString]

```
SET /a _rand= (%RANDOM%*500/32768) +1
```



Setarea variabilelor Exemplu

```
@echo off
Set /P _dept=Please enter Department || Set _dept=NothingChosen
If "%_dept%"=="NothingChosen" goto :sub_error
If /i "%_dept%"=="finance" goto sub_finance
If /i "%_dept%"=="hr" goto sub_hr
goto:eof
```

```
:sub_finance
echo You chose the finance dept
goto:eof
```

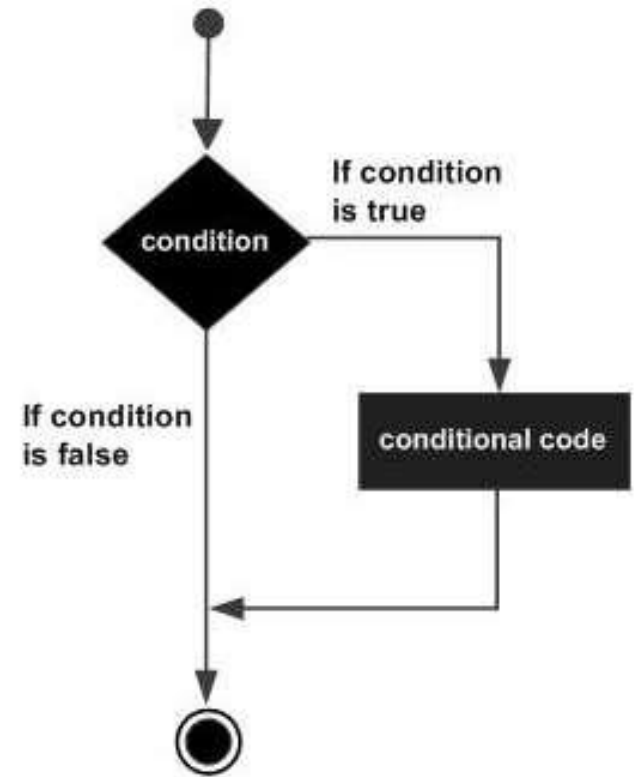
```
:sub_hr
echo You chose the hr dept
goto:eof
```

```
:sub_error
echo Nothing was chosen
```



Sintaxă: IF

```
if(condition) do_something
```



Sintaxă: IF

- /I = comparare case insensitive
- NOT = verifica daca conditia e falsa
- Item = sir de caractere / variabila de mediu
- == = egalitate in compararea sirurilor de caractere
- compare-op: EQU, NEQ, LSS, LEQ, GTR, GEO
(-, !=, <, <=, >, >=)
=> comparare numere

```
IF [/I] [NOT] item1==item2 command
```

```
IF [/I] item1 compare-op item2 command
```

```
IF [/I] item1 compare-op item2 (command) ELSE  
(command)
```



Sintaxă: IF

- EQU vs ==
- Comparare 2 cu 19, "2" cu "19", 026 cu 10, "026" cu "10"

- Comparare numerica:

```
IF 2 GEQ 15 echo "bigger"
```

- Comparare siruri:

```
IF (2) GEQ (15) echo "bigger"
```

```
IF "2" GEQ "15" echo "bigger"
```



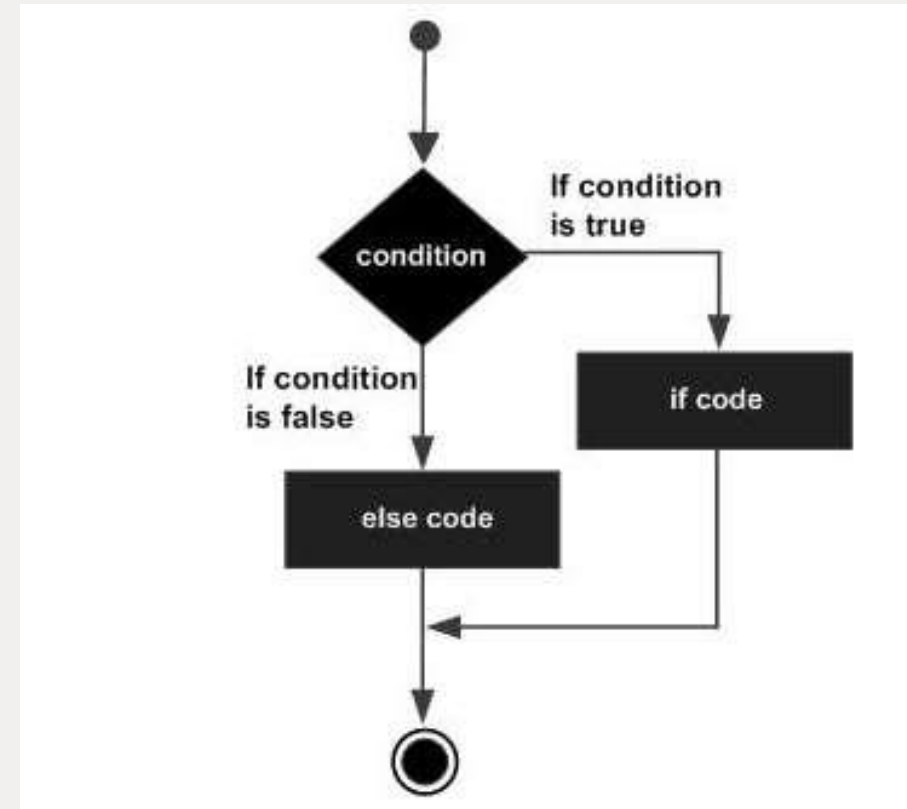
Exemplu: IF

```
@echo off
SET /A a=5
SET /A b=10
SET /A c=%a% + %b%
if %c% == 15 echo "The value of variable c is 15"
if %c% == 10 echo "The value of variable c is 10"
```



Sintaxă: IF / ELSE

```
If (condition) (do_something) ELSE  
(do_something_else)
```



Exemplu: IF / ELSE

```
@echo off
SET /A a=5
SET /A b=10
SET /A c=%a% + %b%
if %c% == 15 (echo "The value of variable c is 15")
else (echo "Unknown value")
if %c% == 10 (echo "The value of variable c is 10")
else (echo "Unknown value")
```



Sintaxă: IF defined

```
If [NOT] defined somevariable somecommand
```

Exemplu: IF defined

```
@echo off
SET str1=String1
SET str2=String2
if defined str1 echo "Variable str1 is defined"
if defined str3 (echo "Variable str3 is defined")
else (echo "Variable str3 is not defined")
```



Sintaxă: IF exists

```
If [not] exist somefile.ext do_something
```

```
If [not] exist somefile.ext do_something else  
do_something_else
```



Exemplu: IF exists

```
@echo off  
if exist C:\set2.txt echo "File exists"  
if exist C:\set3.txt (echo "File exists") else  
(echo "File does not exist")
```



Sintaxă: IF – condiții multiple

```
if(condition1) if (condition2) do_something
```



Exemplu: IF – condiții multiple

```
@echo off
SET /A a=5
SET /A b=10
if %a% == 5 if %b% == 10 echo "The value of the
variables are correct"
```



Exemple: IF

- Testare daca variabila exista

```
IF [%1]==[] ECHO Value Missing
```

```
IF [%1] EQU [] ECHO Value Missing
```

- Utilizarea parantezelor pentru a grupa comenzile:

```
IF EXIST filename.txt (  
    Echo deleting filename.txt  
    Del filename.txt  
) ELSE (  
    Echo The file was not found.  
)
```



Exemple: IF

- Pipe

```
IF SomeCondition Command1 | Command2
```



```
(IF SomeCondition Command1 ) | Command2
```

```
IF SomeCondition (Command1 | Command2)
```

- Comanda IF considera parantezele din jurul conditiilor ca fiind caractere care trebuie comparate

```
IF (%_var1%==(demo Echo the variable  
_var1 contains the text demo
```

- IF poate sa se regaseasca si in dreapta pipe-ului

```
Echo Y | IF red==blue del *.log
```



Exemple: IF

- Singurul operator logic suportat de IF este NOT
- AND:

```
IF SomeCondition (  
    IF SomeOtherCondition (  
        Command_if_both_are_true  
    )  
)
```

- OR:

```
Set /A _tempvar=0  
If SomeCondition Set /A _tempvar=1  
If SomeOtherCondition Set /A _tempvar=1  
if %_tempvar% EQU 1  
Command_to_run_if_either_is_true
```



Exemple: IF

- Daca sirurile de caractere contin delimitatori, cum ar fi [] (spatiu) sau [,] (virgula):
 - Delimitatorii trebuie evitati folosind ^
SAU
 - Sirurile de caractere trebuie incapsulate intre ""



Sintaxă: CHOICE

```
CHOICE [/c [choiceKeys]] [/N]  
[/CS] [/t Timeout /d Choice] [/m  
Text]
```

*/C[:]*choiceKeys = Una sau mai multe taste de apasat. Implicit: YN.

/N = Nu afiseaza valorile posibile

/CS = optiuni case sensitive

/d choice = Optiunea implicita – se poate folosi doar impreuna cu */T*

/m text = Textul care descrie optiunile

/T Timeout = Timeout in secunde. Daca acesta e o se va selecta varianta implicita
=> se seteaza variabila de mediu **errorlevel** cu numarul corespunzator optiunii

```
CHOICE /C CH /M "Select [C] CD or [H] Hard drive"
```



Sintaxă: GOTO

```
...  
goto :label  
...some commands  
:label  
...some other commands
```

GOTO:eof

Exemplu: GOTO

```
@echo off
SET /A a=5
SET /A b=10
if %a% == 5 goto :labela
if %b% == 10 goto :labelb
:labela
echo "The value of a is 5"
:labelb
echo "The value of b is 10"
SET /A c=%a% + %b%
echo %c%
```



Exemplu: **GOTO**

```
IF %1==12 GOTO specialcase
    Echo the input was NOT 12
goto:eof

:specialcase
    Echo the input was 12
goto:eof
```



Exemplu: GOTO

Folosirea unei variabile pentru definirea unei etichete

```
@echo off
CHOICE /C:01 /m "choose [0]yes or [1]No"
    goto sub_%ERRORLEVEL%

:sub_1
Echo You typed 0 for yes
goto:eof

:sub_2
Echo You typed 1 for no
goto:eof
```



Exemplu: GOTO

Folosirea unei variabile pe post de comentariu

```
@Echo Off
Setlocal
SET _skip=
IF /I NOT %1==Update SET _skip=::

%_skip% COPY demo.bat demo1.bat
%_skip% echo Update applied
```

GOTO nu se poate folosi intre () inclusiv in IF si FOR deoarece se iese din context



Exemplu: GOTO

GOTO nu se poate folosi intre () inclusiv in IF si FOR deoarece se iese din context
=> in aceste cazuri se recomanda folosirea CALL

```
@echo off
if A equ A (
    GOTO :EXAMPLE_LABEL
    :EXAMPLE_LABEL
    rem
) else (
    echo You didn't expected to see this,did you?
)
```



Sintaxă: CALL

```
CALL [drive:][path]filename [parameters]
```

```
CALL :label [parameters]
```

```
CALL internal_cmd
```



Exemplu: CALL

Transmiterea parametrilor

```
CALL MyScript.cmd "1234"  
CALL OtherScript.cmd %_MyVariable%
```



Exemplu: CALL

apel catre alt script

```
::-----start main.cmd-----  
@Echo off  
CALL function.cmd 10 first  
Echo %_description% - %_number%  
  
CALL function.cmd 15 second  
Echo %_description% - %_number%  
  
::-----start function.cmd-----  
@Echo off  
:: Add 25 to %1  
SET /a _number=%1 + 25  
:: Store %2  
SET _description=[%2]
```



Exemplu: CALL

apel catre eticheta

```
CALL :sub_display 123
CALL :sub_display 456
ECHO All Done
GOTO :eof

:sub_display
ECHO The result is %1
EXIT /B
```



Exemplu: CALL

Transmiterea prin referinta

```
@Echo off
Echo:
Set var1="Red Pippin"
Set var2="St Edmunds Pippin"
Set var3="Egremont Russet"

Echo: before: var1=%var1% var2=%var2% var3=%var3%
call :myGetFunc var1 var2 var3
Echo: after: var1=%var1% var2=%var2% var3=%var3%
Echo:&pause&goto:eof

:myGetFunc
Set %~1="return64"
Set %~3="return65"
EXIT /B
```



Exemplu: CALL

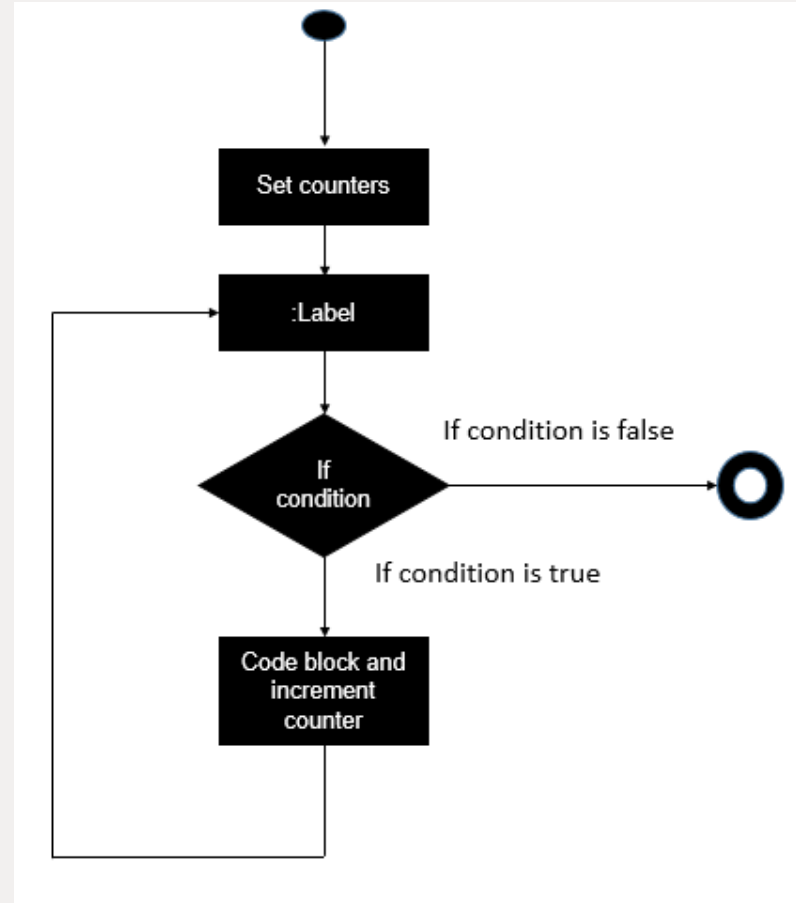
apelul unor alte comenzi

```
@Echo off
SETLOCAL
set _server=frodo
set _var=_server
CALL SET _result=%%%_var%%%
echo %_result%

=> SET _result=frodo
```



Sintaxă: *WHILE*



Sintaxă: **WHILE**

```
Set counters  
:label  
If (expression)  
(  
Do_something  
Increment counter  
Go back to :label  
)
```



Exemplu: WHILE

```
@echo off
SET /A index=1
SET /A "count=5"
:while
if %index% leq %count% (
echo The value of index is %index%
SET /A index = index + 1
goto :while
)
```



Sintaxă: Șiruri , de caractere

- Șir gol

```
@echo off  
SET a=
```

- Verifică șir gol

```
[%a%] == []
```



Exemplu: Concatenarea șirurilor de caractere

- Folosind operatorul SET

```
@echo off
SET a=Hello
SET b=World
SET c=%a% and %b%
echo %c%
```



Sintaxă: Subșiruri de caractere

```
%variable:~num_chars_to_skip%
```

```
%variable:~num_chars_to_skip,num_chars_to_keep%
```

```
%variable:~num_chars_to_skip, -num_chars_to_keep%
```

```
%variable:~-num_chars_to_skip,num_chars_to_keep%
```

```
%variable:~-num_chars_to_skip,-num_chars_to_keep%
```



Exemple: Subșiruri de caractere

```
SET _test=123456789abcdef0
```

```
SET _result=%_test:~0,5%
```

⇒ 12345 (se extrag primele 5 caractere)

```
SET _result=%_test:~7,5%
```

⇒ 89abc (skip 7 caractere, se extrag urmatoarele 5)

```
SET _result=%_test:~7%
```

⇒ 89abcdef0 (skip 7 caractere, se extrag restul)

```
SET _result=%_test:~-7%
```

⇒ abcdef0 (se extrag ultimele 7 caractere)

Exemple: Subșiruri de caractere

```
SET _test=123456789abcdef0
```

```
SET _result=%_test:~0,-7%
```

⇒ 123456789 (se extrag toate caracterele in afara de ultimele 7)

```
SET _result=%_test:~7,-5%
```

⇒ 89ab (se extrag toate caracterele incepand cu al 8-lea in afara de ultimele 5)

```
SET _result=%_test:~-7,5%
```

⇒ abcde (din ultimele 7 caractere se extrag primele 5 caractere)

```
SET _result=%_test:~-7,-5%
```

⇒ ab (din ultimele 7 caractere se extrag ultimele 5 caractere)

Sintaxă: Subșiruri de caractere

```
%variable:to_remove_substring=%
```

```
%variable:old_to_replace=new%
```



Exemple: Subșiruri de caractere

```
SET _test=12345abcabc  
SET _result=%_test:12345=Hello %  
=> Hello abcabc - Inlocuieste '12345' cu 'Hello '
```

```
SET _test=12345abcabc  
SET _result=%_test:ab=xy%  
=> 12345xycxyc - Inlocuieste 'ab' cu 'xy'
```

```
SET _test=12345abcabc  
SET _result=%_test:ab=%  
=> 12345cc - Sterge 'ab'
```

::Delete the character string 'ab' and everything before it

```
SET _test=12345abcabc  
SET _result=%_test:*ab=%  
=> cabc - Sterge 'ab' si toate caracterele care apar inainte  
de prima aparitie
```



Exemple: Subșiruri de caractere

Sa se verifice daca un subsir exista intr-un sir:

```
Set _cities="Aberdeen, London, Edinburgh"
```

```
Set _dummy=%_cities:London=%
```

```
IF NOT %_dummy% == %_cities% (ECHO London was found.)
```

```
ELSE (ECHO London was not found.)
```



Sintaxă: Șiruri

- Fiecare element dintr-un șir se definește folosind SET
- Bucla FOR se utilizează pentru a parcurge șirul

```
SET a[0]=1
```

```
@echo off  
set lista=1 2 3 4  
(for %%a in (%lista%) do (  
echo %%a  
))
```



Sintaxă: Șiruri

- Accesare elemente

```
@echo off  
set a[0]=1  
echo %a[0]%
```



Sintaxă: Șiruri

- Modificare elemente

```
@echo off
set a[0]=1
set a[1]=2
set a[2]=3
set a[1]=5
echo %a[1]%
```



Sintaxă: Șiruri

- Parcurgere șiruri

```
@echo off
setlocal enabledelayedexpansion
set topic[0]=comments
set topic[1]=variables
set topic[2]=Arrays
set "topic[3]=Decision making"
set "topic[4]=Time and date"
set topic[5]=Operators
for /l %%n in (0,1,5) do (
echo !topic[%%n]!
)
```

