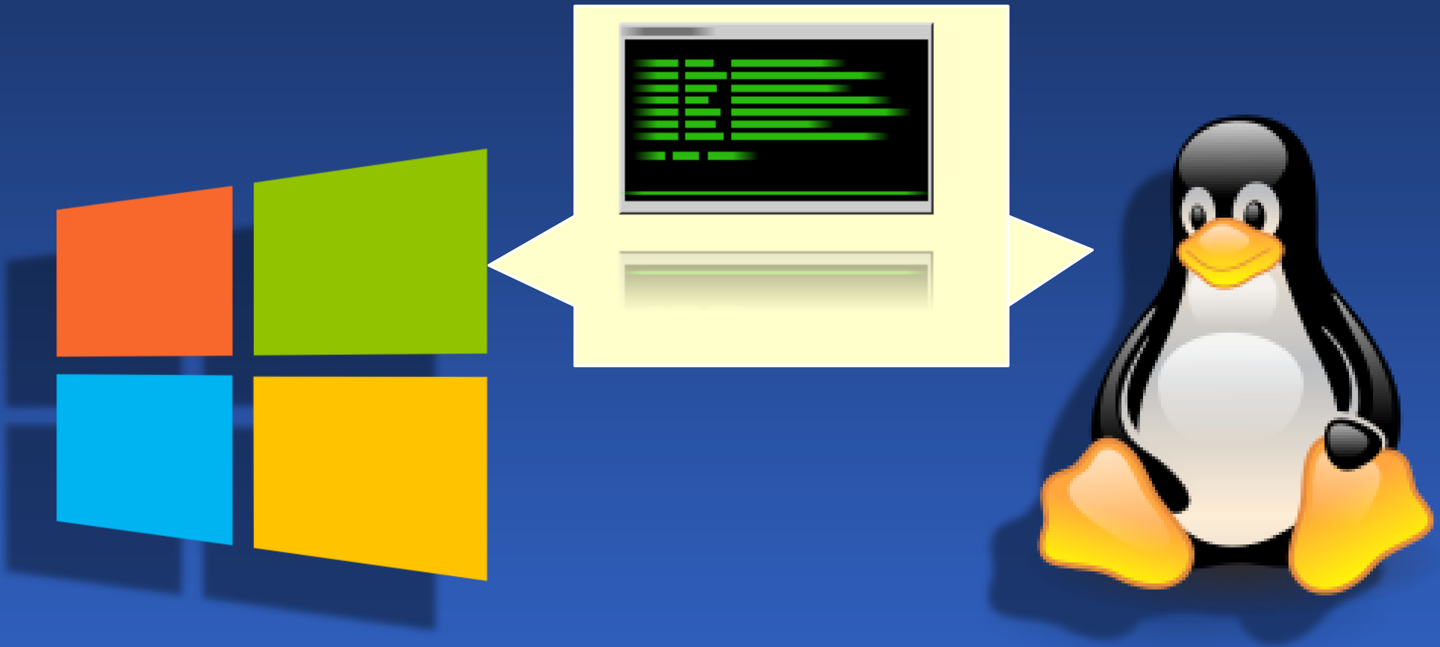
System shells



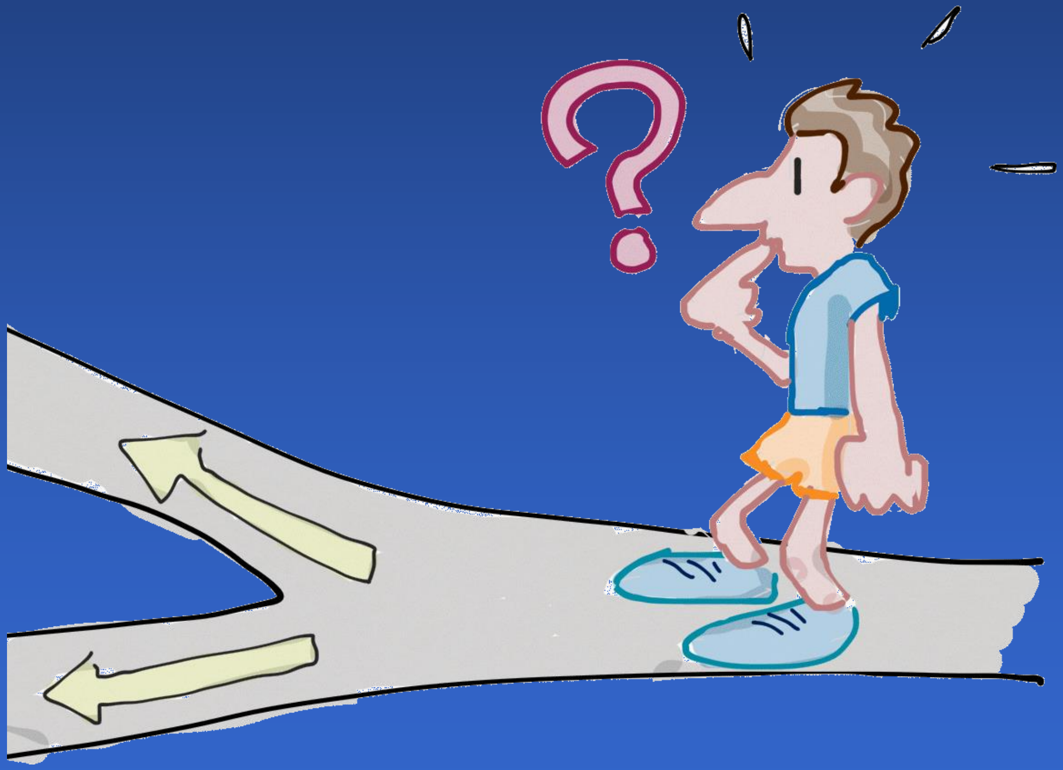
BAT flow control - 4

```
                                    workspace
16/12/2022  12:45              2 295 zaza.txt
                                      123 383 octets
                                  529 472 octets libres

                                  pelle Tweety
 Le numéro de                       5E35-93A3

 Répertoire de C:\unix

05/12/2022  13:34              2 768 goal.o
17/01/2023  11:53              2 568 lab1.o
10/11/2022  10:07              1 472 lab11.o
01/09/2022  16:31              2 416 lab2.o
02/09/2022  10:58              1 968 lab3.o
20/01/2023  12:31              2 880 lateBinding
20/01/2023  11:51              1 976 main
20/01/2023  12:16              1 896
09/11/2022  12:59              2 600
09/11/2022  15:00              2 664
04/01/2023  12:09
              11 fichier(s)                176 octets
               0 Rép(s)                    096 octets libres

C:\unix>
C:\unix>
C:\unix>
```

4 – FLOW CONTROL

IF, GOTO etc.

Choices



Check information

- Fill scr5.bat

```
@echo off
REM -- check user input --
SET /P ANSWER= ... Do you feel good (Y/N)?

IF %ANSWER%==Y echo YOU FEEL GOOD :-)
```

- Run the scr5.bat batch script
  - Inform users who typed "Y" (not Yes, y, yes)
  - What about "N" ?
  - What about other answers (Dunno, QWertY) ?
  - Can I have another chance to type ?

IF and NOT

- Modify scr5.bat

```
@echo off
REM -- check user input --
set /P ANSWER= ... Do you feel good (Y/N)?

IF NOT %ANSWER==Y echo YOU DON'T FEEL GOOD
```

- Run the scr5.bat batch script
  - Reverse logic

IF and case

- Modify scr5.bat

```
@echo off
REM -- check user input --
set /P ANSWER= ... Do you feel good (Y/N)?

IF /I %ANSWER:~0,1%==Y echo YOU FEEL GOOD :-)
```

- Run the scr5.bat batch script
  - ~~Inform users who typed "Y" (not Yes, y, yes)~~
  - What about "N" ?
  - What about other answers (Dunno, QWertY) ?
  - Oops! May I have another chance ?

IF and ELSE

- Modify scr5.bat

```
@echo off
REM -- check user input --
set /P ANSWER= ... Do you feel good (Y/N)?

IF /I %ANSWER:~0,1%==Y (echo YOU FEEL GOOD :-^))^
 ELSE (echo BAD NEWS :-^()
```

One line command

- Run the scr5.bat batch script
  - ~~Inform users who typed "Y" (not Yes, y, yes)~~
  - ~~What about "N" ?~~
  - What about other answers (Dunno, QWertY) ?
  - Oops! May I have another chance ?

Nested IF ... ELSE

- Fill scr6.bat

```
@echo off
set /P ANSWER= ... Do you feel good (Y/N)?

IF /I %ANSWER:~0,1%==Y (
echo YOU FEEL GOOD :-^)
) ELSE (
   IF /I %ANSWER:~0,1%==N (
   echo BAD NEWS :-^(
   ) ELSE (
   echo WONG ANSWER ???
   )
)
```

Escape special char !!

- Run the scr6.bat batch script
  - ~~Inform users who typed "Y" (not Yes, y, yes)~~
  - ~~What about "N" ?~~
  - ~~What about other answers (Dunno, QWertY) ?~~
  - Oops! May I have another chance ?

GOTO and labels

- Modify scr6.bat

```
@echo off
:question
set /P ANSWER= ... Do you feel good (Y/N)?
IF /I %ANSWER:~0,1%==Y (
echo YOU FEEL GOOD :-^)
) ELSE (
   IF /I %ANSWER:~0,1%==N (
   echo BAD NEWS :-^(
   ) ELSE (
   echo WONG ANSWER -- RETRY...
   GOTO :question
   )
)
```

Label

JUMP

- Run the scr6.bat batch script
  - ~~Inform users who typed "Y" (not Yes, y, yes)~~
  - ~~What about "N" ?~~
  - ~~What about other answers (Dunno, QWertY) ?~~
  - ~~Oops! May I have another chance ?~~

CALL and labels

- Fill scr7.bat

```
@echo off
echo before routine
call :routine AA BB
echo after routine
goto :eof

:routine
echo i am in routine %1 %2
goto :eof
```

Call with arguments

BAT will exit
(or do next lines)

return

- Run the scr7.bat batch script
  - Routine is called and returns
  - `:eof` is a reserved virtual label
  - Main program has to exit, or it will execute remaining lines

Check files & folders

- Fill scr8.bat

```
@echo off
IF NOT EXIST %HOMEPATH%\LABS md %HOMEPATH%\LABS
pushd %HOMEPATH%\LABS
echo BEEN HERE at %TIME% >> info.txt
popd

IF EXIST %HOMEPATH%\LABS\info.txt (
type %HOMEPATH%\LABS\info.txt
del %HOMEPATH%\LABS\info.txt )
```

- Run the scr8.bat batch script
  - Will create LABS if necessary
  - Here we use reverse logic
  - Only prints the file if it exists

Check Variables

- Modify if3.bat

```
@echo off

IF NOT DEFINED LABFOLDER set LABFOLDER=HOMEPATH\LABS

echo %LABFOLDER%
```

- Run the scr8.bat batch script
  - Note that `%...%` is not used in the expression

Check ERRORS

- Modify scr8.bat

```
@echo off
MD %HOMEPATH%\AB:C
IF ERRORLEVEL 1 goto :error

Set /a VAL=09
IF ERRORLEVEL 1 goto :error

Goto :eof

:error
Echo A PROBLEM OCCURED
```

- Run the scr8.bat batch script
  - Does not evaluate Set expression
  - ERRORLEVEL will be >0 in case of an error

Comparison == operator

- Type scr9.bat

```
@echo off
set VAR1=AAA
set VAR2=AAA
set VAR3="AAA"

IF    %VAR1%==%VAR2%     (echo %VAR1%=%VAR2%
                        ) else (echo %VAR1% ! %VAR2%)
IF "%VAR1%"=="%VAR2%"  (echo "%VAR1%"="%VAR2%”
                        ) else (echo "%VAR1%" ! "%VAR2%")
IF "%VAR1%"==%VAR2%    (echo "%VAR1%"%VAR2%
                        ) else (echo "%VAR1%" ! %VAR2%)
IF "%VAR1%"==%VAR3%    (echo "%VAR1%"=%VAR3%
                        ) else (echo "%VAR1%" ! %VAR3%)
```

- The " separator goes in the Sting
  - Useful when testing the empty string ""

Operator == and numbers

- Type cmp2.bat

```
@echo OFF
set /a ELEVEN=11
set /a OELEVEN=013
set /a HELEVEN=0xB

IF "%ELEVEN%"=="%OELEVEN%" echo "1. == OCTAL ..."
IF %ELEVEN%==%OELEVEN% echo "2. == OCTAL ..."
IF %ELEVEN%==%HELEVEN% echo "3. == HEXA..."
```

- The « number behind » is evaluated

Extended operators

- Comparison operators

`EQU`	=
`NEQ`	!=
`LSS`	<
`LEQ`	<=
`GTR`	>
`GEQ`	>=

- Apply on numbers and strings
  - Non-quoted strings made of digits are evaluated as numbers (includin `0`… and `0x`…)

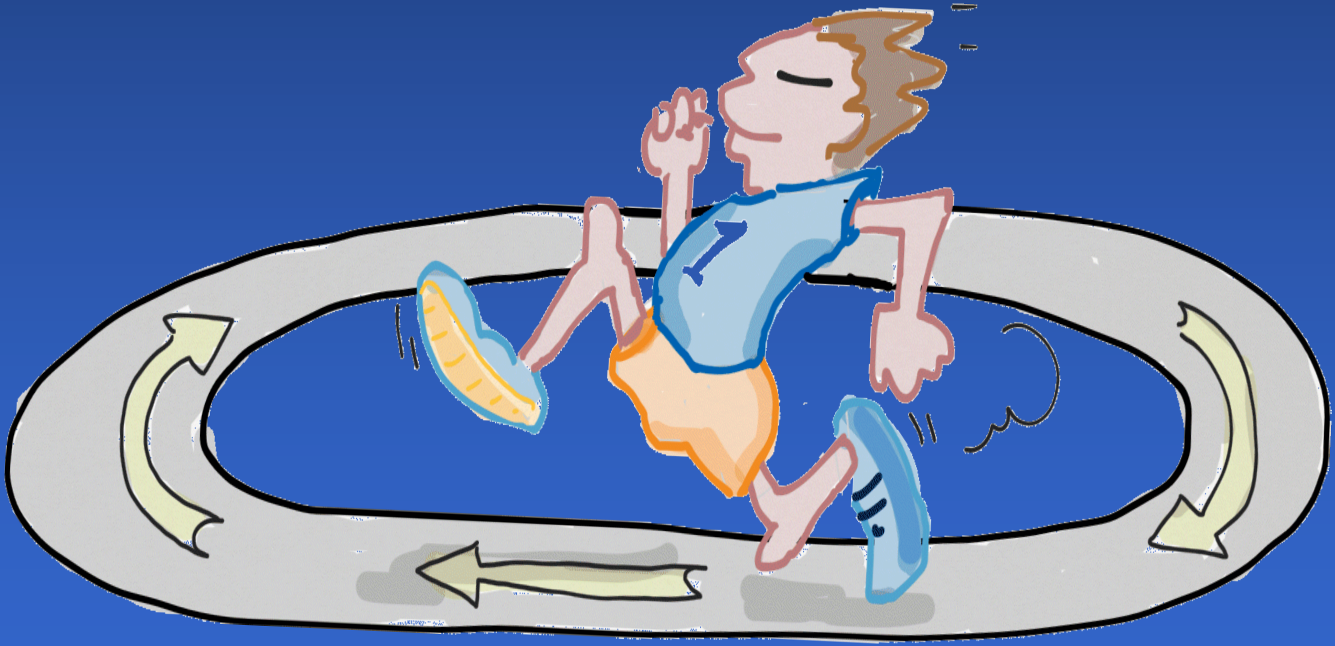
Operator EQU

- Type cmp8.bat

```
echo =====NUMBERS=====
set /a ELEVEN=11
set /a OELEVEN=013
set /a HELEVEN=0xB
IF %ELEVEN% EQU %OELEVEN% echo "1. EQU OCTAL"
IF "%ELEVEN%" EQU "%OELEVEN%" echo "2. EQU quoted"
IF %ELEVEN% EQU %HELEVEN% echo "3. EQU HEXA"

echo =====STRINGS=====
set ELEVEN=11
set OELEVEN=013
set HELEVEN=0xB
IF %ELEVEN% EQU %OELEVEN% echo "4. EQU OCTAL ..."
IF NOT "%ELEVEN%" EQU "%OELEVEN%" echo "5. NOT EQU quoted"
IF %ELEVEN% EQU %HELEVEN% echo "6. EQU HEXA..."
```

- `EQU` evaluates strings `013` and `0xB` as number `11`
  - NOT `"013"` ... beware

Loops



The FOR command

`FOR` *each-item* `IN` *collection* `DO` *command*

- **General structure**
  - *each-item* must be a letter variable `%%a` `%%B` etc.
  - `%%` only applies inside scripts not on the command line *(use %)*
  - Letter variables are case sensitives *(max = 52)*
  - If *collection* is a command or a list, use `(  )`
  - *collection* may be *regex* using `*` and `?` to filter file names

- **Type for1.bat**

```
@echo off
FOR %%c IN (reg green blue) DO echo %%c
FOR %%c IN (reg,green;blue) DO echo %%c
FOR %%c IN (for?.bat if*.bat) DO echo %%c
```

FOR command options

FOR [options] *each-item* **IN** *collection* **DO** *command*

- Command structure
  - **/D** to filter folder names *(i.e. not file names)*
  - **/R** to explore recursively a following root folder *(i.e. /R root)*
  - **/L** to iterate on numbers, collection is *(start, step, stop)*
  - **/F** to set the collection as a file or a command output
- modifiers
  - **/A** to genetate each-item as a number
  - **/I** to make processing NOT case sensitive
  - **/Q** no error messages *(i.e Quiet)*
  - **/T** introduces a delay at each iteration
  - **/C** command is a string containing spaces

FOR /R as a tree filter

- Type for2.bat

```
@echo off
Set MYLABS=%USERPROFILE%\LABS
Echo. > mybat.txt
FOR /R %MYLABS% %%a in (*.bat) do echo %%~fa >> mybat.txt
```

- `FOR /R` is exploring recursively
  - Collection must be a an expression
  - This command may take some time to complete

FOR /L as an iterator

- Type for5.bat

```
@echo off
set /p nb=which number do you want to sum-up :
set /a formula=(%nb%*(%nb%+1))/2
set /a sum=0
FOR /L %%i in (1,1,%nb%) do (
    set /a sum+=%%i
)
echo sum(%nb%)=%sum%  (Nx(N+1))/2=%formula%
```

- `FOR /L` is efficient for iterating

FOR /F command filters

`FOR /F` [filters] *each-item* `IN` *collection* `DO` *command*

- Filter expressions

`eol=c`	eliminate characters after « c » in the line
`skip=n`	eliminate n first lines
`delims=,`	defines several items (`%%a, %%b, %%c` …)
`usebacq`	allow folder/files names with spaces

→ use back-quotes

FOR as a string parser

- Type for2.bat

```
@echo off

FOR /F "tokens=1-3 delims=^/" %%A in ("%DATE%") do (
    echo %%A-%%B-%%C )

FOR /F "tokens=1-4 delims=:," %%A in ("%TIME%") do (
   echo %%Ah%%B^'%%C^"
   echo %%D ms )
```

- `FOR /F` is efficient for parsing strings
  - `tokens` will define multiple variables
  - `delims` will extend the standard separators

FOR as a file parser

- Type for4.bat

```
@echo off
FOR /F "tokens=1-3 eol=; skip=5 delims=," %%A in (%~nx0) do (
    echo %%A note %%C)
exit /b 0

;LAST-FIRST-SCORE
DURAND,Claude,12
DAVID,John,15
MARTIN,Elsa,17
```

- `FOR /F` is efficient for parsing files
  - `eol` will eliminate commented lines
  - `skip` will eliminate first lines