

MS-DOS UTILISATION DES POSSIBILITÉS DE PROGRAMMATION POUR LA CRÉATION DE PROCÉDURES BAT

Michel DEVAUD

I. INSTRUCTIONS SPÉCIFIQUES

1) Instructions

COMMAND

Permet d'appeler depuis une procédure BAT une autre procédure BAT comme s'il s'agissait d'un sous-programme.

Syntaxe : `COMMAND /C nom-de-procédure`

`COMMAND` doit être utilisé conjointement avec `EXIT`.

EXIT

Termine une procédure BAT appelée comme sous-programme. Réalise un retour à la ligne qui suit `COMMAND`.

ECHO

Suivi de `ON` ou `OFF` : autorise ou supprime l'écho des commandes. Normalement une procédure BAT débute par `ECHO OFF`.

Echo devient automatiquement `ON` à la sortie de la procédure.

Seul : donne l'état de l'écho (`ON` ou `OFF`). Suivi d'un texte : affiche ce texte à l'écran.

FOR ... IN ... DO ..".

Instruction d'itération.

Syntaxe

`FOR paramètre IN (liste) DO instruction`

Exemple : Taper directement au clavier :

`FOR %F IN (UN DEUX TROIS) DO ECHO %F`

GOTO

Instruction de branchement.

Syntaxe : GOTO étiquette

Exemple :

```
.
GOTO ETIQ
```

```
.
```

```
.
```

```
:ETIQ
```

```
.
```

```
.
```

IF

Instruction conditionnelle.

Syntaxe : IF condition Instruction

PAUSE

Arrêt de la procédure avec affichage du message : "Tapez une touche pour poursuivre"

ou "Strike a key when ready"

On peut alors arrêter le déroulement de la procédure en tapant Ctrl C. Toute autre touche fait passer à la suite.

REM

Permet d'insérer un commentaire dans le texte de la procédure.

SHIFT

Décalage des paramètres (cf. plus loin).

2) Expressions booléennes

==

Comparaison de chaînes de caractères.

Syntaxe :

Chaîne == Chaîne

Vraie si les chaînes sont identiques. Exemples

Abc==Abc → VRAI

Abc==abc → FAUX

Abc==Xyz → FAUX

EXIST

Syntaxe : **EXIST** nom-de-fichier

Donne le résultat **VRAI** si nom-de-fichier existe dans le répertoire courant (on ne peut pas préciser de chemin d'accès).

ERRORLEVEL

Teste le contenu du registre **AL** (compte rendu de fin du programme précédemment exécuté). Rend le résultat **VRAI** si **AL** est supérieur ou égal au paramètre de **ERRORLEVEL**.

3) Paramètres

%0 %1 ... %9

Dix paramètres représentant la ligne de commande tapée. **%0** représente le nom de la commande, **%1** son premier argument, **%2** le deuxième...

Exemple : **A>TRUC A: TOTO TITI**

%0 %1 %2 %3

L'instruction **SHIFT** permet un décalage (irréversible) vers la gauche. Cela permet, en particulier, de traiter plus de 10 paramètres.

Exemple :

| | %0 | %1 | %2 | %3 ... |
|----------------|-----------------|-------------|-------------|-----------------|
| | TRUC | A : | TOTO | TITI ... |
| SHIFT → | A : | TOTO | TITI | TATA ... |
| SHIFT → | TOTO | TITI | TATA | |
| | ... etc. | | | |

%F

Paramètre à prendre dans une liste (cf. **FOR ... IN ... DO ..**)

%%F

Paramètre de second niveau. Les paramètres du type **%1** ou **%F** (un seul **%**) sont remplacés par leur valeur à l'entrée dans la procédure. Les paramètres de type **%%F** (deux **%**) sont remplacés par leur valeur lors de l'exécution de la procédure.

%NOM%

Paramètre à prendre dans le "string" d'environnement. Voir la commande **SET** de MS-DOS ainsi que l'exemple **COPIE.BAT**.

II. EXEMPLES DE PROCÉDURES BAT

1) Format.bat

But :

Une machine possède deux disquettes A: et B: et un disque dur C:. On souhaite éviter qu'un utilisateur formate le disque dur.

Mise en œuvre :

Il faut tout d'abord changer le nom de FORMAT.COM. En effet, lors de la frappe d'une commande, MS-DOS la cherche d'abord dans les commandes résidentes, puis dans les fichiers .COM, puis dans les fichiers EXE et, enfin, dans les fichiers BAT (un fichier BAT ne peut donc porter le même nom qu'une commande interne ou COM ou .EXE de MS-DOS).

Puisqu'il s'agit de rendre le programme FORMAT.COM difficilement accessible, je propose de le renommer FORMAT_.COM où le caractère _ est le caractère de code 255 (il a la couleur et le goût de l'espace mais ce n'est pas un espace - on l'obtient en appuyant sur la touche ALT tout en tapant 255). Nous allons donc lancer la commande :

```
REN C:FORMAT.COM C:FORMAT_.COM
```

Liste commentée de FORMAT.BAT :

```
ECHO OFF                               Supprime l'écho
IF %1x==x GOTO SYNT                     Si %1 est vide ...
IF %1==A: GOTO FORM
IF %1==a: GOTO FORM                     Formatage de A: ou B:
IF %1==B: GOTO FORM
IF %1==b: GOTO FORM
IF %1==C: GOTO DDUR                     On demandé le formatage du disque dur
IF %1==c: LOTO DDUR
:SYNT                                    Rappel de la syntaxe
ECHO Tapez FORMAT A: ou FORMAT B:
ÉCHO Merci.
GOTO FIN
:FORM                                    Formatage de A: ou B:
FORMAT %1                                FORMAT(ALT 255)
GOTO FIN
:DDUR                                    Refus de formater
ECHO                                     Vous n'avez pas à formater le disque dur !
ECHO                                     Au revoir.
```

:FIN

Fin de la procédure

2) Copie.bat

But : Cet utilitaire permet de copier une liste de fichiers (COPY n'accepte qu'un seul paramètre).

Syntaxe : Pour copier des fichiers de A: vers B:

```
A>COPIE A: B: TOTO.TRC *.COM SI*.* ... etc.
```

Liste commentée de COPIE.BAT :

```
ECHO OFF
SET DSKS=%1           Sauvegarde dans le "string"
SET DSKB=%2           d'environnement des noms des disques
                       source et but.
ECHO Placez le disque source en %1 et le disque but en %2
PAUSE
:DEBUT
IF %3x==x GOTO FIN    Pas ou plus de paramètre
ECHO ON
COPY %DSKS%%3 %DSKB% Copie du fichier
ECHO OFF
SHIFT                 Décalage LOTO DEBUT
:FIN
SET DSKS=              Nettoyage du "string"
SET DSKB=              d'environnement.
```

3) Copyc.bat

But : Copier tous les fichiers d'un disque sur un autre, à condition qu'ils n'existent pas déjà.

Syntaxe : COPYC B : A :

Copie tous les fichiers de B : sur A : s'ils n'existent pas déjà.

Liste commentée de COPYC.BAT :

```
%1                   On se loge sur le disque source
FOR %%F IN (*.*) DO IF NOT EXIST %%F COPY %%F %2
```

Remarque : ECHO n'est pas mis OFF afin que l'on puisse suivre le déroulement des opérations.

4) Menu.bat

But : Illustrer l'utilisation de ERRORLEVEL dans une procédure proposant un menu à l'utilisateur (par exemple, le choix entre deux programmes).

Utilisation : Cette procédure affichera à l'écran le menu suivant :

```

MENU:
-----

1 ... Programme 1
2 ... Programme 2
3 ... Terminer

Votre choix ?

```

Utilitaire de saisie :

Pour que cette procédure puisse fonctionner, il nous faut écrire un petit utilitaire permettant la lecture d'un caractère au clavier et rendant son code ASCII dans AL. Cela peut se faire facilement sous DEBUG.

```

C>DEBUG
-a 100                                offset début du programme
46E8:0100 MOV AH, 00)
46E8:0102 INT 16)                      Instructions du programme
46E8:0104 MOV AH, 4C )                 de saisie clavier
46E8:0106 INT 21 )
46E8:0108                               Taper juste RETURN
-r CX                                  CX doit contenir
CX 0000                                la longueur en octets
:8                                     du programme : 8
-n getcar.com                          Nom donné au programme
-w                                       Ecriture sur le disque
Ecriture 0008 octets
-q                                     Sortie de DEBUG
C>

```

Liste commentée de MENU.BAT :

```

ECHO OFF
:DEBUT
CLS
ECHO MENU:
ECHO -----
ECHO .
ECHO 1 ... Programme 1
ECHO 2 ... Programme 2
ECHO 3 ... Terminer
ECHO .
:CHOIX
ECHO Votre choix ?
GETCAR
IF ERRORLEVEL 52 GOTO ERREUR
IF ERRORLEVEL 51 GOTO FIN
IF ERRORLEVEL 50 LOTO CAS2
IF ERRORLEVEL 49 LOTO CAS1
:ERREUR
ECHO Choix incorrect !
GOTO CHOIX
:CAS1
ECHO Lancement du programme 1
PRG1
GOTO DEBUT
:CAS2
ECHO Lancement du programme 2
PRG2
GOTO DEBUT
:FIN
ÉCHO Au revoir ...

```

Saisie caractère tapé
52 = code ASCII de 4
51 = code ASCII de 3
50 = code ASCII de 2
49 = code ASCII de 1

Par exemple !

Par exemple !

III. AUTOEXEC.BAT

Cette procédure BAT est particulière : elle est systématiquement exécutée lors du lancement du système MS-DOS. Elle doit être dans le répertoire *racine* du disque système.

AUTOEXEC.BAT va nous permettre de configurer automatiquement notre système et d'en augmenter la "convivialité".

Pour ma part, j'utilise AUTOEXEC.BAT conjointement avec une autre procédure : DEPART.BAT. Cette deuxième procédure est lancée par AUTOEXEC.BAT et présente un menu des applications disponibles. DEPART.BAT est relancée à la fin de l'exécution de chacune des applications.

Voyons comment peuvent se présenter ces procédures :

1) Autoexec.bat

| | |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ECHO OFF | |
| PATH C :\ ;\UTIL ; ... | Définition de chemins d'accès par défaut. Dans mon système, le répertoire racine ne contient, en plus des fichiers qui doivent obligatoirement s'y trouver (CONFIG.SYS, AUTOEXEC.BAT...), que des sous-répertoires. Le sous-répertoire UTIL contient tous les utilitaires du système (FORMAT.COM, EDLIN.COM, ..., fichiers BAT...) |
| KEYBFR | Chargement du "driver" de clavier |
| PROMPT \$P\$G | Affichage du répertoire courant dans le "prompt" de MS-DOS. |
| DATE | Affichage et mise à jour de la date (peu intéressant si le système dispose d'une horloge sauvegardée) |
| TIME | Affichage et mise à jour de l'heure (peu intéressant si le système dispose d'une horloge sauvegardée) |
| DEPART | Lancement de DEPART.BAT |

2) Départ.bat

| | |
|----------------|-------------------------------------------------------------------------------------------------|
| ECHO OFF | |
| CD C :\ | Sélection du répertoire racine. |
| CLS | Efface écran |
| VER | Affiche numéro de version MS-DOS |
| ÉCHO _ | "_"représente ALT 255 (pour sauter une ligne). ÉCHO tout seul ferait afficher l'état de l'écho. |
| ÉCHO _ | ÉCHO _ |
| ÉCHO Blablabla | En-tête personnalisé |
| ÉCHO _ | |

```

ÉCHO _
                                         Affichage menu des applications :
ÉCHO MENU :
ECHO -----
ÉCHO _
ÉCHO _
ÉCHO TDT :                               Logiciel de traitement de texte
ÉCHO BD :                                Gestionnaire de bases de données
ÉCHO TAB :                               Tableur
                                         ... etc.

ÉCHO _
ÉCHO _
ÉCHO Votre choix ?
ÉCHO

```

L'utilisateur tapera alors le code de l'application désirée. Ce code correspond à une procédure BAT (placée par exemple dans le sous-répertoire UTIL) qui pourrait avoir l'allure suivante (pour le traitement de texte, TDT.BAT, par exemple) :

```

ÉCHO OFF
CLS
ÉCHO Chargement du logiciel de traitement de texte
ÉCHO Patience ...
CD TEXTE                               Changement de répertoire
PRGTDT                                 Nom du programme de T.D.T.
DÉPART                                 On relance DEPART.BAT à la sortie de
                                         l'application.

```

Michel DEVAUD