L'ENCHAÎNEMENT DES COMMANDES SOUS MS-DOS

LES REDIRECTIONS

A) DÉFINITION

A moins de lui indiquer le contraire, MS-DOS saisit l'entrée d'une commande au clavier et envoie la sortie sur l'écran. Or il est parfois utile de diriger la sortie d'une commande vers un fichier ou vers l'imprimante. Par exemple la fonction **tree** ne possède pas de paramètre permettant d'afficher son résultat page par page, il peut donc être intéressant d'envoyer le résultat de cette commande directement sur l'imprimante ou dans un fichier que l'on visualisera ensuite sous un éditeur.

B) LES CARACTÈRES DE REDIRECTION

- redirige la sortie d'une commande vers un fichier ou vers un périphérique ⇨**dir** a: >listerep.txt , **tree** a: >arbre.txt
- < extrait l'entrée de la commande d'un fichier plutôt que d'un clavier ⇨**sort** <listerep.txt.
- >> ajoute la sortie de la commande à la fin d'un fichier sans supprimer ce qui était déjà dans le fichier .⇨**dir** c: >>listerep.txt

C) LES COMMANDES FILTRE

Les commandes filtre retraitent des portions de données qui passent par elles. MS-DOS possède trois commandes filtre :

- **More** : qui affiche le contenu d'un fichier ou le résultat d'une commande un écran à la fois.
- **Find** : qui recherche des caractères spécifiés dans un fichier ou dans les sorties d'une commande
- **Sort** : qui tri par ordre alphabétique des fichiers et des sorties de commandes.

D) UTILISATION DES COMMANDES FILTRE

On veut afficher l'arborescence des répertoires page par page :
Tree a: >arbre.txt
More < arbre.txt
On veut afficher uniquement les fichiers d'un répertoire :
Dir c: >liste.txt
Find /v «REP» <liste.txt >listefic.txt
More <listefic.txt
On veut connaître tous les fichiers de la disquette et du disque dur et les voir par ordre alphabétique.
Dir c: >listefic.txt
Dir a: >>listefic.txt
Find /v «REP» <listefic.txt >pasrep.txt
Sort <pasrep.txt >fictri.txt
More < fictri.txt

Attention à l'issue de ces commandes les fichiers existent toujours. Il faut penser à les détruire.

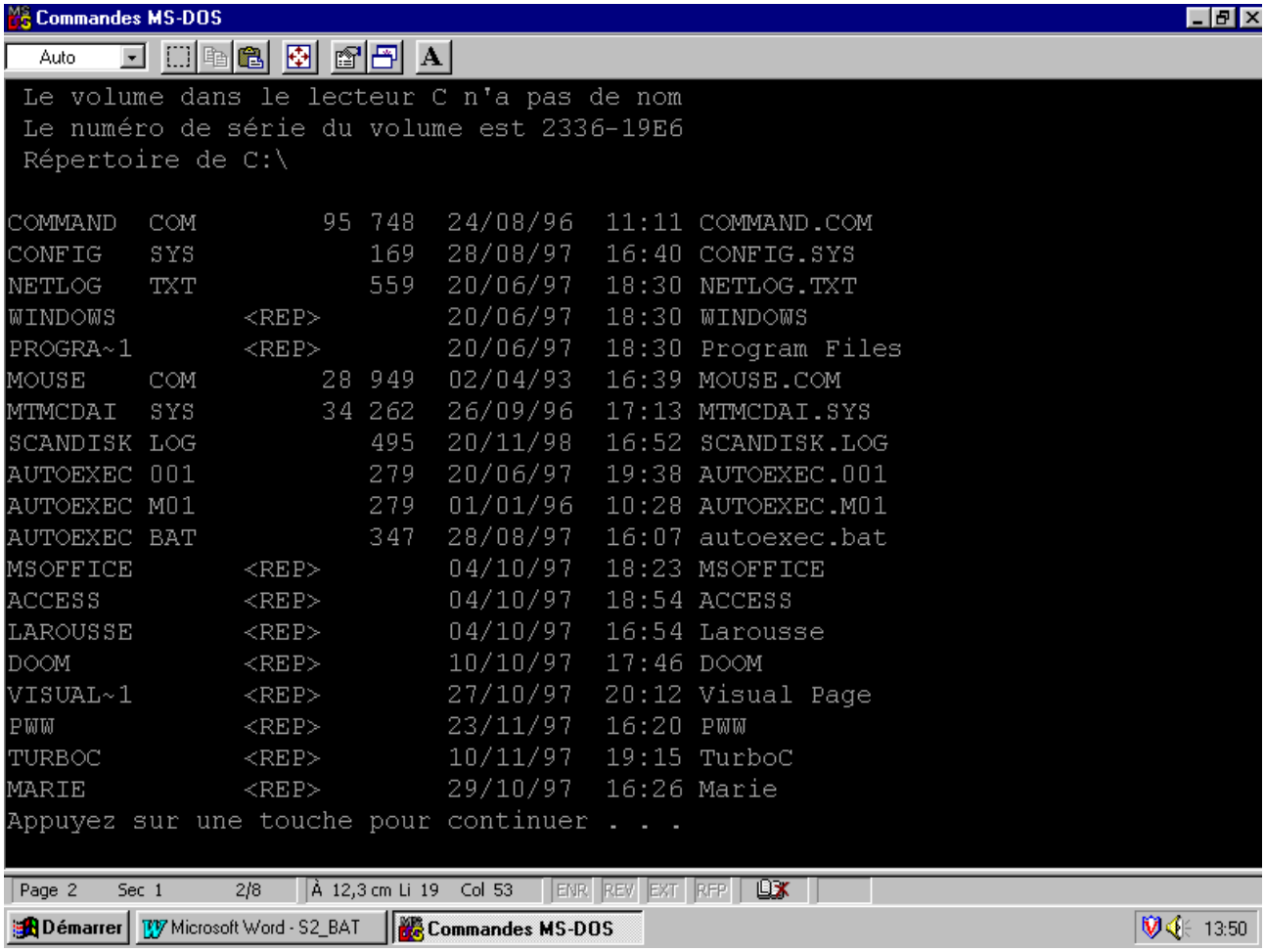
E) LA REDIRECTION DIRECTE

Pour éviter le création de nombreux fichiers intermédiaires on peut faire de la redirection directe entre plusieurs commandes, en utilisant le symbole « | »

Tree a: | More
Dir c: | Find /v «REP» | more
Dir c: >listefic.txt
Dir a: >>listefic.txt
Find /v «REP» <listefic.txt | Sort | More

F) EXERCICES

Afficher écran par écran tous les fichiers du disque dur crées en 98, hors répertoire.
C:\> **dir** c : | **find** « /98 » | **find** /v « REP »
Idem mais les afficher triés par heure.
C:\> **dir** c : /O:d | **find** « /98 » | **find** /v « REP »
C:\> **dir** c : | **find** « /98 » | **find** /v « REP » |sort /+39
Idem mais les afficher triés par taille décroissante.
C:\> **dir** c : | **find** « /98 » | **find** /v « REP » |sort /r /+14
C:\> **dir** c : /O :-s| **find** « /98 » | **find** /v « REP »



A) DÉFINITION - UTILITÉ

Ces fichiers se composent de lignes de commande de MS-DOS telles que l'utilisateur les frappe au clavier.
MS-DOS lit séquentiellement chaque ligne , donc chaque commande du fichier et exécute ces commandes au fur et à mesure. Lorsque toutes ces commandes sont épuisées , le système d'exploitation reprend la main.
Ces fichiers de traitement par lot fournissent à l'utilisateur un moyen souple et puissant pour gérer l'utilisation du système.
Entre autre ils évitent la répétition de frappe fastidieuse de commandes indispensables.
Les fichiers de traitement par lot sont reconnus par MS-DOS par leur extension . Celle-ci est **.BAT** (abréviation de BATCH se traduisant par lot)

Exemple de fichier de commande

```
C:\> TYPE EXEMP1.BAT
    date
    time
    dir /W
```

Une fois constitués, ces fichiers sont considérés comme des commandes par DOS. Pour les lancer en exécution il est inutile de préciser leur extension.

L'éxécution de ces fichiers peut s'interrompre à tout moment par **Ctrl-C**. DOS affiche le message :

Terminer le fichier de commandes (O/N) ?
La réponse : O interrompt l'exécution et redonne la main au système
N laisse l'exécution se poursuivre normalement

Un fichier de commande peut en contenir un autre : ils sont emboîtables. Mais toutefois, il faut savoir qu'en fin d'exécution du fichier de commande emboîté, la main est redonnée au système d'exploitation et non au fichier de commande primitif (sauf instruction CALL).

B) QUELQUES INSTRUCTIONS SPÉCIFIQUES AUX FICHIERS .BAT

L'instruction ECHO

MS-DOS affiche sur l'écran les commandes des fichiers de commandes avant de les exécuter. L'instruction ECHO commande cette visualisation.

ECHO OFF	inhibe cette visualisation
ECHO ON	Rétablit cette visualisation (le mode normal)
ECHO	sans argument ,délivre à l'écran le mode actuel
ECHO<msg>	dirige le contenu du message vers l'écran
@ECHO OFF	@ inhibe l'affichage de ECHO OFF

L'instruction REM (remarque)

REM<message> agit comme ECHO sauf que REM n'est pris en compte que dans le mode ECHO ON.

En fait ECHO s'utilise pour renseigner l'opérateur et REM pour documenter le fichier de commande.

L'instruction PAUSE

L'instruction suspend l'exécution du fichier de commande et affiche "Appuyer sur une touche pour continuer « Strike a key when ready ... »

PAUSE sert à donner à l'opérateur :

→ le temps de réfléchir
→ la possibilité de changer une disquette
→ la possibilité d'interrompre l'exécution par Ctrl-C

Le Label et GOTO

Le label est une chaîne de 1 à 8 car alphanumériques.
Il apparaît en position d'argument dans l'instruction GOTO.
Utilisé comme étiquette sur une ligne de commande , il doit être précédé de deux points :

Exemple : **GOTO** FIN
...
:FIN
DATE

Si le label n'est pas défini en position d'étiquette lors d'un GOTO, l'exécution du fichier est stoppée. Le message "label not found" s'affiche.

C) LES MENUS

L'instruction CHOICE

Elle permet de choisir parmi des valeurs proposées et de définir une valeur par défaut qui sera prise en compte au bout d'un temps défini. Le choix effectué modifie la variable ERRORLEVEL qui sera testée grâce à l'instruction IF.

Exemple

Echo *
Echo Faites un choix parmi les options du menu
Echo Choix 1 : Lancer Word
Echo Choix 2 : Revenir à MS-DOS
Echo *
CHOICE /C :12 /N /T :2,30
Permet de choisir entre la valeur 1 et la valeur 2. La valeur 2 est choisie par défaut au bout de 30 secondes.

L'instruction IF

Elle permet dans le cas d'un menu créé avec l'instruction **CHOICE**, d'orienter le traitement en fonction du choix de l'utilisateur.

IF ERRORLEVEL <valeur> GOTO <label>
Attention il s'agit ici d'un test >= et non pas d'un test =, il faut donc classer les valeurs possibles par ordre décroissant.
Pour notre MENU
IF ERRORLEVEL 2 GOTO Word

```
IF ERRORLEVEL 1 GOTO Fin
:Word
etc.
:Fin
```

D) EXERCICE

Ecrire un fichier de commande permettant de copier vos fichiers WORD et EXCEL sur une disquette, sur laquelle on retrouvera un répertoire pour WORD et un autre pour EXCEL. Ce fichier de commande permettra de créer plusieurs fois cette disquette si l'utilisateur le désire.

```
@echo off
cls
rem Procédure pour copier les fichiers Word et excel des répertoires du disque dur
rem vers des répertoires de la disquette.
rem pas de paramétrage.
:debut
echo Insérez une disquette dans le lecteur a:
pause
cd c:\marie\msoffice\btsig\s2\word
mkdir a:\word
copy *.* a:\word\*.*
cd c:\marie\msoffice\btsig\s2\excel
mkdir a:\excel
copy *.* a:\excel\*.*
c:
cd ..\exploit
choice /c:on /n /t:n,30 "Voulez-vous faire une autre disquette ?"
if errorlevel 2 goto fin
if errorlevel 1 goto debut
:fin
echo.
echo Fin de la procedure
echo.
```

FICHIERS DE COMMANDES AVEC PARAMETRES

A) DÉFINITION : LES PARAMÈTRES

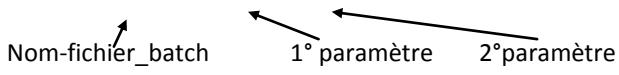
Les commandes incluses dans un fichier de commandes peuvent être paramétrées i.e. que les objets de ces commandes sont des paramètres formels.

Les paramètres réels seront fournis lors du lancement du fichier batch :

A:\> nom-fichier-batch param1 param2

Exemple :

A:\> **Affichage rouge noir**



Lors de l'appel du fichier, les paramètres réels seront substitués aux paramètres formels à mesure de l'exécution des différentes commandes.

Les paramètres formels s'écrivent **%n** , où n est un chiffre de 0 à 9 représentant la position du paramètre réel dans la liste des paramètres réels fournie lors du lancement du fichier de commande.

%0 représente donc le nom du fichier de commande lui-même.
%1 représente la valeur du 1° paramètre.
.....
%n représente la valeur du nième paramètre.

Exemple : Dans le fichier affichage.bat, la ligne de commande est **echo %1** :

résultat : **rouge**

Exemple : Construire un fichier de commande concaténant 2 fichiers existant pour en créer un 3° (Voir copy).

La commande dans le fichier CONCAT.BAT s'écrit :

```
COPY   %1.C + %2.C  %3.C
```

Les fichiers considérés sont supposés avoir l'extension .C
CONCAT est le nom du fichier de commande . L'appel peut se faire de la façon suivante (Les fichiers ver-1.c et suppl.c existent) :

```
CONCAT    VER-1    SUPPL     SUITE
  (%0)      (%1)     (%2)      (%3)
```

Il y a, à l'exécution, substitution des paramètres réels (VER-1...) et la commande finalement exécutée sera :

```
COPY    VER-1.C+SUPPL.C    SUITE.C
```

B) QUELQUES INSTRUCTIONS SPÉCIFIQUES AUX FICHIERS DE COMMANDES.

L'instruction IF

L'instruction conditionnelle se présente sous la forme
IF [NOT] <condition> <commande>
Tester l'égalité : **==** IF « %1 » == « STOP » GOTO FIN
Tester l'existence : **EXIST** IF NOT EXIST a:\langage\langc\%1.C GOTO ERROR

L'instruction FOR

C'est une boucle d'itération qui se présente sous la forme:
FOR %%<caractères> IN (<liste>) DO <commande> %%<caractères>

<caractères> valeur alphabétique = var muette

Exemple Créer un sous-répertoire "trinome" contenant 3 sous-répertoires portant le nom de chaque participant du trinome.

Ecrire CREAT-CAT.BAT sous un éditeur de texte quelconque.
A:\> EDIT CREA-CAT.BAT

```
mkdir %1
chdir %1
FOR %%+ IN (%2 %3 %4) DO mkdir %%+
```

A:\> **CREA-CAT trinome pascal pierre jacques** /* Appel */

L'instruction SHIFT

Cette instruction permet d'utiliser plus de 9 paramètres réels par simple décalage.

Après chaque instruction SHIFT le paramètre réel 0 disparaît, le paramètre réel 1 devient 0 ... le paramètre 10 qui ne pouvait pas être atteint devient 9 donc disponible et ainsi de suite.

Exemple : afficher le contenu des répertoires données en paramètre sans connaître au préalable leur nombre.

A:\> EDIT CAT.BAT

```
ECHO OFF
:cat-suiv
IF « %1 » == « STOP » GOTO FIN
DIR /P %1
SHIFT
PAUSE
GOTO cat-suiv
:Fin
Echo "listage des répertoires terminé"
ECHO ON
```

APPEL : A:\> **CAT \langage \langage\langC \logiciel\word**

C) EXERCICE

Ecrire une procédure permettant d'afficher le contenu d'un fichier du repertoire courant. Le contenu s'affiche page à page et le nom du fichier est entré en paramètre.

```
@echo off
if exist %1 goto ficexist
echo Aucun fichier répondant au critère saisi : %1
goto fin
: ficexist
echo Le fichier existe
type %1 | more
:fin
echo fin de procédure
```