

TP 3 Commandes DOS

CONTEXTE

Le dépannage d'un ordinateur Windows nécessite de connaître les commandes Dos. Ce petit TP va vous faire manipuler les commandes les plus connues. Pour obtenir de l'aide sur une commande "commande/?".

GESTION DES FICHIERS

Tapez un texte quelconque en utilisant la commande *edit* et sauvegardez ce fichier sous le nom *essai.dat*.

Vérifiez (dir) que ce fichier est bien dans votre répertoire personnel.

Visualisez (more) le contenu d'*essai.dat*.

```
C:\Users\Nathan>dir desktop
Le volume dans le lecteur C s'appelle Windows
Le numéro de série du volume est AE8C-6CA5

Répertoire de C:\Users\Nathan\desktop

12/10/2021 09:11 <DIR> .
12/10/2021 09:11 <DIR> ..
16/09/2021 11:02 <DIR> BackupUSB
12/10/2021 09:09          32 essai.dat.txt
15/09/2021 08:15    11 134 336 Oracle_VM_VirtualBox_Extension_Pack-6.1.26.vbox-extpack
30/09/2021 09:38 <DIR> Programes_Powershell
30/09/2021 06:16 <DIR> Programes_Python
16/09/2021 10:58      1 181 752 rufus-3.15.exe
29/09/2021 11:45 <DIR> TP_NMAP
12/10/2021 09:04      26 694 TP3_DOS.odt
02/03/2021 17:17      2 877 227 008 ubuntu-20.04.2.0-desktop-amd64.iso
5 fichier(s)      2 889 569 822 octets
6 Rép(s)  830 260 899 840 octets libres

C:\Users\Nathan>more essai.dat
Impossible d'accéder au fichier C:\Users\Nathan\essai.dat

C:\Users\Nathan>more essai.dat.txt
Impossible d'accéder au fichier C:\Users\Nathan\essai.dat.txt

C:\Users\Nathan>cd desktop

C:\Users\Nathan\Desktop>more essai.dat.txt
texte plus ou moins quelconque

C:\Users\Nathan\Desktop>dir desktop
Le volume dans le lecteur C s'appelle Windows
Le numéro de série du volume est AE8C-6CA5

Répertoire de C:\Users\Nathan\Desktop

Fichier introuvable

C:\Users\Nathan\Desktop>more essai.dat.txt
texte plus ou moins quelconque
```

Changez le nom (rename) d'*essai.dat* en *essai1.dat*.

```
C:\Users\Nathan\Desktop>rename essai.dat.txt essai1.dat.txt

C:\Users\Nathan\Desktop>more essai1.dat.txt
texte plus ou moins quelconque

C:\Users\Nathan\Desktop>
```

Copiez (copy) `essai1.dat` vers `essai2.dat`.

```
C:\Users\Nathan\Desktop>copy essai1.dat.txt essai2.dat
1 fichier(s) copié(s).
```

Visualisez (dir) tous les fichiers « .dat » de votre répertoire et uniquement ceux-ci.

```
C:\Users\Nathan\Desktop>dir *.dat
Le volume dans le lecteur C s'appelle Windows
Le numéro de série du volume est AE8C-6CA5

Répertoire de C:\Users\Nathan\Desktop

12/10/2021  09:09                32 essai1.dat
12/10/2021  09:09                32 essai2.dat
               2 fichier(s)                64 octets
               0 Rép(s)   830 248 214 528 octets libres
```

Copiez (copy) les fichiers `essai1.dat` et `essai2.dat` dans un fichier global `texte.dat`.

```
C:\Users\Nathan\Desktop>copy essai1.dat+essai2.dat texte.dat
essai1.dat
essai2.dat
        1 fichier(s) copié(s).

C:\Users\Nathan\Desktop>more texte.dat
texte plus ou moins quelconque
texte plus ou moins quelconque
→

C:\Users\Nathan\Desktop>
```

Copiez (copy) le fichier `texte.dat` vers `texte1.dat`.

```
C:\Users\Nathan\Desktop>copy texte.dat texte1.dat
1 fichier(s) copié(s).
```

Détruisez (del) en une seule fois tous les fichiers dont le nom commence par "texte".

```
C:\Users\Nathan\Desktop>del /A "texte*"
```

Détruisez (del) tous les fichiers "essai?.dat".

```
C:\Users\Nathan\Desktop>del /A "essai*.dat"
```

GESTION DES RÉPERTOIRES

Créez (mkdir) le répertoire `Perso` et à l'intérieur les dossiers `Langage`, `Logiciels` et `Autre`.

```
C:\Users\Nathan\Desktop>cd C:\Users\Nathan
C:\Users\Nathan>mkdir Perso
C:\Users\Nathan>cd Perso
C:\Users\Nathan\Perso>mkdir Langage Logiciels Autre
```

Visualisez (tree) la hiérarchie créée.

```
C:\Users\Nathan\Perso>tree
Structure du dossier pour le volume Windows
Le numéro de série du volume est AE8C-6CA5
C:.
├── Autre
├── Langage
└── Logiciels
```

Créez (edit) un fichier `essai.dat` dans le répertoire `Autre` tout en restant dans le répertoire `Perso`.

```
C:\Users\Nathan\Perso>notepad C:\Users\Nathan\Perso\Autre\essai.dat
```

Allez (`cd`) dans le répertoire `Autre`.

Le répertoire de travail étant `Autre`, copiez ce fichier `essai.dat` sous le répertoire `Langage`.

```
C:\Users\Nathan\Perso>cd Autre

C:\Users\Nathan\Perso\Autre>copy essai.dat C:\Users\Nathan\Perso\Langage
1 fichier(s) copié(s).
```

Détruisez (`rmdir`) le répertoire `Perso` et tout ce qu'il contient en une seule commande.

```
C:\Users\Nathan\Perso\Autre>cd C:\Users\Nathan
C:\Users\Nathan>rmdir /S C:\Users\Nathan\Perso
C:\Users\Nathan\Perso, êtes-vous sûr (O/N) ? o
```

(/Q ne demande pas de confirmation)

LES REDIRECTIONS

Par défaut, MS-DOS lit l'entrée d'une commande saisie 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.

">" redirige la sortie d'une commande vers un fichier ou vers un périphérique. Exemples :

dir c: > listerep.txt

tree c: > 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*

LES FILTRES

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

- `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.

Exemples d'utilisation :

On veut afficher l'arborescence des répertoires page par page :

tree c: > arbre.txt

more < arbre.txt

On veut afficher uniquement les fichiers d'un répertoire :

```
dir c: > liste.txt
find /v Windows < liste.txt > listefic.txt
more < listefic.txt
```

On veut connaître tous les fichiers du disque dur et les voir par ordre alphabétique.

```
dir c: > listefic.txt
find /v Windows < listefic.txt > pasrep.txt
sort < pasrep.txt > fictri.txt
more < fictri.txt
```

LA REDIRECTION

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

```
tree c: | more
dir c: | find /v Windows | more
find /v Windows < listefic.txt | sort | more
```

EXERCICES

Afficher écran par écran et par ordre alphabétique tous les fichiers du disque dur créés en 2019.

```
C:\>dir /s/n/o:n/t:c|find "/2019" > C:\Users\Nathan\fichier_dir.txt
```

Idem mais les afficher triés par heure (2 solutions : option de la commande dir ou de sort).

```
C:\Windows>dir /s/o:d/t |find "/2019" > C:\Users\Nathan\Documents\fichier_dir2.txt
```

Cette commande génère un bug du au « /s », mais nous n'y avons pas trouver d'explications.

Idem mais les afficher triés par taille décroissante.

```
C:\Windows>dir /s/o:-s/t:c|find "/2019" > C:\Users\Nathan\Documents\fichier_dir3.txt
```

LES FICHIERS DE COMMANDES

Ces fichiers se composent de lignes de commande DOS qui sont lues et exécutées 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 frappes fastidieuses de commandes. Ces fichiers de traitement par lot sont reconnus par le DOS par leur extension .bat (abréviation de BATCH se traduisant par lot).

Exemple de fichier de commande :

```
c:\> type exemple.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'exécution de ces fichiers peut s'interrompre à tout moment par Ctrl-C.

QUELQUES INSTRUCTIONS SPÉCIFIQUES AUX FICHIERS .BAT

L'instruction ECHO

Par défaut, 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 mais affiche cette commande à l'écran
ECHO ON	Rétablit cette visualisation (le mode normal)
ECHO	sans argument, affiche 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..." Elle sert à donner à l'opérateur le temps de réfléchir, la possibilité de changer un disque, la possibilité d'interrompre l'exécution par Ctrl-C...

L'instruction GOTO

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

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.

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 /D:2 /T : 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
```

...

```
:Fin
```

...

Exercice : Écrire un fichier de commande permettant de faire plusieurs copies (si l'utilisateur le souhaite) sur des clés (d:) différentes du répertoire Mes Documents.

```
« « « -----
```

```
@echo off
```

```
echo Faites un choix parmi les options du menu
```

```
echo Choix 1 : copier sur le support externe D:
```

```
echo Choix 2 : Revenir à MS-DOS
```

```
CHOICE /C 12 /M "Faites votre choix : "
```

```
IF ERRORLEVEL 2 GOTO salam
```

```
IF ERRORLEVEL 1 GOTO copy
```

```
:copy
```

```
copy C:\Users\Nathan\Documents D:\Documents
```

```
GOTO menu
```

```
:menu
```

```
echo Faites un choix parmi les options du menu
```

```
echo Choix 1 : copier sur le support externe D:
```

```
echo Choix 2 : Revenir à MS-DOS
```

```
CHOICE /C 12 /M "Faites votre choix : "
```

```
IF ERRORLEVEL 2 GOTO salam
```

```
IF ERRORLEVEL 1 GOTO copy
```

```
:salam
```

```
PAUSE
```

```
echo "Appuyez sur une touche pour quitter..."
```

```
exit
```

```
« « « -----
```

LES PARAMÈTRES

Les commandes incluses dans un fichier de commandes peuvent être paramétrées, c'est-à-dire que les objets de ces commandes sont des paramètres formels. Les paramètres réels seront fournis lors du lancement du fichier batch : c:\> nom-fichier-batch param1 param2

Exemple : c:\> *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 "rouge".

%2 représente la valeur "noir".

Dans le fichier affichage.bat, la ligne de commande est echo %1 affichera "rouge".

Exercice : Construire un fichier de commande concaténant 2 fichiers existants pour en créer un 3e (Voir copy).

LES BOUCLES

L'instruction IF

L'instruction conditionnelle se présente sous la forme :

IF [NOT] <condition> <commande1> [ELSE <commande2>]

Les comparants utilisables sont :

- EQU - égal à
- NEQ - différent de
- LSS - inférieur à
- LEQ - inférieur ou égal à
- GTR - supérieur à
- GEQ - supérieur ou égal à

Ou bien on peut tester l'existence d'un fichier. Exemple : *if not exist test.jpg echo fichier manquant*

L'instruction FOR

C'est une boucle d'itération qui se présente sous la forme :

FOR %%<variable> IN (<liste>) DO <commande>

Exercice : créer un sous-répertoire "trinome" contenant 3 sous-répertoires portant le nom de chaque participant du trinôme entrés comme paramètres.

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, etc. Le paramètre 10 qui ne pouvait pas être atteint devient 9 donc disponible et ainsi de suite.

Exercice : Afficher le contenu des répertoires donnés en paramètre sans connaître leur nombre.