

TP2 - Shell Windows: boucles "for" et manipulation de chaînes

1 Rappels TP1

Q1.1 - Chercher dans le cours la commande permettant d'afficher la liste des variables d'environnement :

Q1.2 - Exécuter cette commande, et chercher le nom de la variable d'environnement donnant le nom de la machine :

Q1.3 - Ecrire un script **T:\M1105\tp2_0.bat** qui prend en argument une chaîne de caractères et qui affiche OUI si cette chaîne correspond au nom de la machine et NON dans le cas contraire, en affichant ce nom (voir cours pour le test de chaînes).

Q1.4 - Ecrire un script **T:\M1105\tp2_1.bat** qui va créer une archive **T:\M1105.zip** de tout le contenu de votre dossier **T:\M1105**. Vous utiliserez le programme "7-Zip" dans sa version en ligne de commande **7z.exe**. Tapez **7z** dans le dossier **C:\Program Files (x86)\7-Zip** pour avoir la syntaxe d'utilisation.

Ce script devra fonctionner quel que soit le dossier courant (utilisation d'un chemin absolu pour le programme 7z).

2 Boucles for

Les boucles "for" sont disponibles sous plusieurs formes selon ce que l'on souhaite faire.

Énumération de fichiers et de dossiers La forme initiale permet d'énumérer des fichiers donnés à travers une spécification de fichier, en utilisant des caractères génériques. Essayez par exemple :

```
FOR %a IN ( \* ) DO ECHO %a
```

Ceci va énumérer tous les fichiers de la racine du disque courant et afficher leur nom la partie entre parenthèses est la **spécification de fichier**).

Q2.1 - Donner la commande **for** (en interactif) permettant d'afficher tous les fichiers du dossier **c:\windows\system32** qui commencent par **win** et qui sont exécutables, et ce sans afficher la commande echo elle-même :

En mode non-interactif (script) et de façon à alléger l'écriture, il sera préférable de faire appel à un sous-programme dans le corps de la boucle. Par ailleurs, dans ce mode, il faudra doubler le caractère '%'.
 Par exemple, pour procéder à un traitement plus complexe que le simple affichage avec la commande donnée ci-dessus, on écrira ceci :

```
for %%a in ( \* ) do call :mon_sp %%a
echo fini !
goto :eof

:mon_sp
set arg1=%1
... ici des trucs...
goto :eof
```

L'option /D passée à la commande FOR permet d'énumérer les dossiers au lieu des fichiers.

Q2.2 - Donner la commande (en interactif) permettant d'afficher tous les dossiers contenus dans **c:\windows** qui commencent par **P** :

Q2.3 - Ecrire un script **T:\M1105\tp2_2.bat** qui va compter le nombre de fichiers ayant une certaine extension de fichier dans le dossier **c:\Windows** et qui afficher le résultat sous la forme :

```
le nbe de fichiers XXX dans c:\Windows est: 123
```

L'extension recherchée sera donnée en argument du script. Par exemple **tp2_2 txt**. Il faudra vérifier que l'argument est **non vide** et appartient à l'une de celle-ci :DLL, INI ou TXT. Il devra afficher une erreur dans les autres cas.

Boucles numériques La deuxième variante de la commande "for" permet de générer des valeurs numériques entières :

```
FOR /L %a IN ([début], [pas], [fin]) DO [commande]
```

en remplaçant [...] par une valeur entière. Par exemple :

```
for /L %a in (1,1,10) do @echo %a
```

va afficher les nombres de 1 à 10

Q2.4 - Ecrire un script qui affiche les nombres de 2 en 2 en partant de la valeur donnée en 1^{er} argument du script et en allant jusqu'à la valeur donnée en 2^e argument. Donner la ligne du "for" :

3 Manipulation de chaînes de caractères

L'interpréteur autorise la manipulation de chaînes de façon à construire des chaînes, ou extraire des sous-chaînes.

Pour la concaténation, l'interpréteur le fait automatiquement. Par exemple, le script suivant affichera **AB=123456**

```
set A=123
set B=456
echo AB=%A%%B%
```

On peut aussi stocker cette concaténation dans une variable, avec la commande **SET var=%A%%B%**

Pour l'extraction de chaînes, ceci se fait avec la commande **SET**, qui peut assigner à une chaîne une partie d'une autre chaîne.

Syntaxe pour créer une variable B à partir d'une partie d'une variable A :

```
SET B=%A:~[debut],[longueur]%
```

Par exemple, en considérant une variable A contenant la chaîne **ABCDEF**, on va donner à la variable B la valeur **CDE** avec la commande :

```
SET B=%A:~2,3%
```

L'argument [longueur] peut être absent.

Q3.1 - Avec la valeur de A ci-dessus, quelle valeur aura B avec la commande :

```
SET B=%A:~2%
```

Q3.2 - Qu'en déduisez-vous? Quelle est la longueur utilisée si elle n'est pas précisée :

Q3.3 - Ecrire un script donnant l'affichage suivant, en utilisant la variable d'environnement **USERNAME** contenant votre identifiant :

```
lastname starts with: XXXXX
first name starts with: XXX
```

4 Exercice de synthèse

Q4.1 - Ecrire un script de nom **T:\M1105\TP2_3.bat** permettant de créer 10 dossiers d'utilisateurs génériques, qui seront nommés **user_00** à **user_09**. Il seront créés (pour l'instant) dans le dossier **T:\M1105\users**. Chacun d'eux devra contenir 3 dossiers : **docs**, **factures**, **commandes**.

Q4.2 - Donner la commande permettant de les effacer, quel que soit le dossier courant :

Q4.3 - Modifier le script pour qu'il prenne en argument le nombre d'utilisateurs à créer. Vérifier que l'utilisateur donne bien un argument. Sinon, afficher un message d'erreur et quitter.

Q4.4 - Modifier le script pour que le chemin racine où ces dossiers sont créés soit donné en deuxième argument. Vérifier que l'utilisateur donne bien deux arguments, et vérifier que ce 2^e argument existe.

Note : on peut tester l'existence d'un fichier ou d'un répertoire avec

```
IF [NOT] EXIST filename command
```

ou

```
IF [NOT] EXIST filename (command) ELSE (command)
```