

## TP2 : Langage de commande Windows

### Fichiers de commande

## 1 Rappels

---

### Script

Les scripts sont des programmes qui permettent d'automatiser certaines tâches répétitives telles que l'administration ou les sauvegardes de fichiers. Ce sont des fichiers de texte avec l'extension `.bat` ou `.cmd`. Ils contiennent les mêmes commandes que celles exécutées en ligne de commande, plus éventuellement des commandes de branchement à des étiquettes (`goto`), et des appels à d'autres scripts (`call`). Un script contient une commande par ligne. Il peut être paramétré.

### Exécuter un script Batch

Pour exécuter un script Batch, il suffit de taper le nom de ce fichier dans la fenêtre de commandes. L'interprète exécute les commandes de manière séquentielle, dans l'ordre où elles apparaissent dans le fichier. Il n'est pas nécessaire de taper l'extension `.bat` du fichier.

### La variable PATH

Pour pouvoir exécuter un programme, le système d'exploitation doit pouvoir retrouver le fichier correspondant à cette commande. C'est la variable d'environnement `PATH` qui contient la liste des répertoires que l'interprète explore pour trouver un programme à exécuter, s'il n'est pas dans le répertoire courant. Vos scripts seront donc placés dans le répertoire `P:\TP_SYS\BATCH` (cf. TP n°1).

## 2 Avant propos

---

### 2.1 Objectifs

- Comprendre l'utilité et les principes de base de la programmation de scripts sous Windows
- Utiliser les principales fonctionnalités du langage de script : création de fichiers de commandes, passage de paramètres, gestion de fichiers, itérations, conditionnelles, analyses de chaînes, redirections, gestion des erreurs, etc.

### 2.2 Avant de programmer

- Vérifiez que le répertoire `P:\TP_SYS\BATCH` existe et qu'il apparaît dans la variable `PATH` (voir TP1).
- Conseil : utilisez l'éditeur de texte SciTE ou Notepad++ pour écrire vos scripts et évitez d'utiliser les caractères accentués qui sont codés différemment.
- Pensez à utiliser votre cours et l'aide en ligne pour la syntaxe des commandes.

## 3 Scripts à écrire pendant la séance de TD

---

### 3.1 Tester l'existence de fichiers et/ou répertoires (fin du TP1)

- Créer un fichier de commandes **testFic.bat** qui reprend l'exemple du cours (ci-contre) sur les expressions conditionnelles.
- Affichez l'aide de la commande. Testez cette commande pour un fichier et un répertoire existants/non existants.

```
@echo off
if "%1"==" " goto usage
if "%1"=="/*" goto usage
if exist %1\NUL (
    echo le repertoire %1 existe !
) else (
    if exist %1 (
        echo le fichier %1 est présent
    ) else echo %1 est absent
)
goto fin
:usage
echo usage : testFic nomfichier
:fin
```

- Commentez le code :

➤ Que représente le `%1` ?

- Que fait `if exist %1\NUL` ? Ce test ne fonctionne pas sur toutes les versions de Windows. Fonctionne-t-il ici ? Il existe des alternatives utilisant des commandes de traitement des répertoires et le test de la condition `ERROLEVEL` (cf. encadré ci-dessous). Modifiez votre script en utilisant la commande `dir /ad`

**ERRORLEVEL** : Toutes les commandes renvoient une valeur de retour pour indiquer leur succès (0) ou leur échec ( $\geq 1$ ). Cette valeur est accessible par une condition de la forme `ERRORLEVEL n` qui signifie « le code renvoyé par la commande précédente est  $\geq n$  ». Pour tester si une commande s'est bien déroulée, on fait le test suivant :

```
IF ERRORLEVEL 1 goto erreur
```

Cette instruction a pour effet de se brancher à l'étiquette `erreur` si la commande

- Modifiez le script **testFic.bat** de façon à ce qu'il permette de tester l'existence de plusieurs fichiers ou répertoires (nombre indéfini de paramètres) en utilisant la commande `SHIFT` et une itération (voir l'exemple donné en cours pour afficher les paramètres d'un script).

### 3.2 Sauvegarder des fichiers et des répertoires

Les scripts Batch peuvent être utiles pour la réalisation de sauvegardes automatiques.

- Ecrire la commande **creerRep.bat** qui crée un répertoire du nom passé en paramètre dans le répertoire courant.
  - Si un répertoire de même nom existe déjà, un message est affiché
  - La commande peut-elle échouer ? Dans quelles conditions ?
  - En utilisant `ERROLEVEL` et la redirection de l'erreur standard, modifier votre commande de sorte qu'un message d'erreur compréhensible s'affiche en cas de problème.

Rappel : le langage de commande Windows utilise les symboles de redirection identiques à ceux d'UNIX :  
`>` et `>>` permettent la redirection des messages vers un fichier  
`2>` et `2>>` permet la redirection des messages d'erreur vers un fichier

- Ecrire la commande **sauver.bat** qui copie des fichiers passés en paramètres dans le répertoire sauvegarde du répertoire courant (à créer s'il n'existe pas).
  - Afficher un message d'erreur si le nom correspond à un répertoire ou si un fichier est absent ;
  - Ecrire un script **dateDuJour.bat** qui affiche :

```
Nous sommes le Xeme jour du Yeme mois de l'annee Z.
```

Pour extraire une sous-chaine d'une variable, on utilise la syntaxe suivante :  
`%nomvar:~d,lg%`  
`nomvar` est une variable d'environnement  
`d` est l'indice de début de la sous-chaine (0 est l'indice du premier caractère)  
`lg` est la longueur de la sous-chaine  
 exemple : si la variable `nomvar=tralala`, alors `%nomvar:~1,3%` désigne la chaine "ral"

- Ajouter l'option `/D:j-m-a` à la commande **sauver.bat** qui permet de spécifier une date. Si cette option est utilisée, seuls seront sauvegardés les fichiers dont la date de dernière modification est postérieure à la date donnée en paramètres. Pour la copie des fichiers, on consultera attentivement l'aide de la commande `xcopy`.

## 4 Travail à rendre : lecture de fichiers, analyses de chaînes, calculs

### Travail à faire individuellement ou en binôme

- Pour chacune de vos commandes (scripts), les 3 points suivants doivent être respectés :
  - Votre code doit être commenté et chaque commande doit avoir un en-tête précisant son rôle et ses paramètres.
  - L'utilisateur doit pouvoir obtenir l'aide des fonctions en tapant : `nomfonction /?`
  - Si l'utilisateur appelle la fonction avec de mauvais paramètres, un message d'erreur et d'aide approprié doit s'afficher.
- A envoyer par mail à [Jean-Michel.Adam@upmf-grenoble.fr](mailto:Jean-Michel.Adam@upmf-grenoble.fr) **au plus tard le 13 novembre 2011 à 23h59**  
(renommer tous les fichiers .bat en .bat.txt puis faire un .zip du répertoire contenant tous les scripts : clic droit sur le répertoire -> créer une archive .zip).
- **Vous devez recevoir un mail de confirmation.** Si vous ne recevez rien : ce n'est pas bon, renvoyez le tp ou demandez confirmation

Les questions suivantes concernent la gestion d'une épreuve de lancé de Javelot. Durant cette compétition, chaque athlète a eu 3 essais.

**3.1** – Écrire la commande **Maximum.bat** qui calcule dans la variable d'environnement MAX et affiche le maximum d'une liste d'entiers passée en paramètres. Le nombre de paramètres n'est pas fixé.

Par exemple, la commande suivante : `maximum 1 2 3 4 3 2 1`

a pour effet d'afficher et de placer dans la variable MAX la valeur 4 car il s'agit du maximum des paramètres.

Dans le langage de commande, on ne peut pas directement tester si un nombre est supérieur à un autre. Pour le faire, il faut utiliser la commande `set` avec l'option `/a` pour faire des calculs. Mais cette commande ne permet pas de manipuler des expressions logiques. Pour savoir si un nombre A est plus grand qu'un nombre B, il faut procéder de la manière suivante :

- a) calculer la différence entre A et B dans une variable intermédiaire en utilisant la commande `set /a`.
- b) si le résultat est négatif, c'est à dire si le résultat commence par le caractère "-", cela signifie que  $A < B$ , dans le cas contraire  $A \geq B$ .

La commande **Maximum.bat** vérifiera la présence d'au moins un paramètre et affichera un message d'aide dans le cas contraire.

**3.2** – Écrire la commande **Categorie.bat** qui calcule dans la variable d'environnement CAT et affiche la catégorie d'âge d'un athlète.

Par exemple, la commande suivante : `categorie 1982`

a pour effet d'afficher et de placer dans la variable CAT la valeur Senior.

En athlétisme, les catégories d'âge de la saison 2011/2012 sont données par le tableau suivant :

Catégorie	Année de naissance
Vétérans	1971 et avant
Séniors	1972 à 1988
Espoirs	1989 à 1991
Juniors	1992 à 1993
Cadets	1994 à 1995
Minimes	1996 à 1997
Benjamins	1998 à 1999
Poussins	2000 à 2001
École d'Athlétisme	2002 et après

**3.3** – Écrire la commande **AfficherRésultat.bat** qui affiche la catégorie et le meilleur lancé d'un athlète donné.

Par exemple, la commande suivante : `AfficherResultat Martin Galle 12-2-1989 94 87 93`

a pour effet d'afficher « Le meilleur lancé du junior Martin Galle est de 94 m ».

La commande `AfficherResultat` doit comporter 6 paramètres : le prénom et le nom de l'athlète, sa date de naissance (au format `jj-mm-aaaa`) et le meilleur de ces 3 essais.

La commande `AfficherResultat` pourra appeler les commandes `Maximum` et `Categorie`.

**3.4** – On dispose d'un fichier de texte appelé `personnes.txt` et contenant une ligne pour chaque athlète avec le format suivant :

Nom:Prenom:date de naissance:essai1:essai2:essai3

Exemple de fichier `personnes.txt` :

```
#Fichiers de personnes
#nom:prenom:dateNaissance:essai1:essai2:essai3
Galle:Martin:12-2-1989:94:87:93
Honnete:Marie:14-9-1980:82:91:93
Pleur:Jean:13-3-1974:93:79:72
Prane:Dolly:19-11-1995:71:67:79
Mieu:Jeff-Edmond:22-12-1992:89:95:91
```

En utilisant la commande `for`, et l'extraction de chaîne, écrire la commande **CalculerResultats.bat** qui affiche, pour chaque athlète du fichier, sa catégorie d'âge et son meilleur lancé en utilisant la commande `AfficherResultat.bat`.

Paramétrez la commande `CalculerResultats.bat` de sorte que le fichier contenant les informations soit un paramètre de la commande.

### Conseils :

1) pensez à :

- Commenter votre code,
- Tester vos scripts séparément,
- Gérer les erreurs et les aides en ligne.

2) Si un script ne fonctionne pas, mettez en commentaire la ligne `@echo off` du script. Vous aurez alors la trace des commandes interprétées et trouverez plus facilement l'origine d'une erreur.