



## 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 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.

## 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 pour écrire vos scripts et évitez d'utiliser les caractères accentués.
- 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.
- Commentez le code :

```
@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
```

- 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 ? Par quelle(s) alternative(s) utilisant les commandes Batch de traitement des répertoires et la condition `ERROLEVEL` (cf. encadré ci-dessous) remplacer ce test ?

**ERROLEVEL** : 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 `ERROLEVEL 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 binômes

- Pour chacune de vos fonctions, les 3 points suivants doivent être respectés :
  - Votre code doit être commenté et chaque fonction 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 [Benoit.Lemaire@upmf-grenoble.fr](mailto:Benoit.Lemaire@upmf-grenoble.fr) (Faire .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.

**4.1** – Écrire la commande **SigneZodiaque.bat** qui calcule dans la variable d'environnement **SIGNE** et affiche **le signe zodiacal correspondant à une date donnée**. La date est donnée sous la forme de 2 paramètres correspondant respectivement au Jour et au Mois.

Les signes du zodiaque correspondent respectivement aux dates suivantes :

Signe	Période dans l'année
Capricorne	du 22 Décembre au 20 Janvier
Verseau	du 21 Janvier au 19 Février
Poissons	du 20 Février au 20 Mars
Belier	du 21 Mars au 20 Avril
Taureau	du 21 Avril au 21 Mai
Gémeaux	du 22 Mai au 21 Juin
Cancer	du 22 juin au 22 juillet
Lion	du 23 juillet au 23 Aout
Vierge	du 24 Aout au 23 septembre
Balance	du 24 septembre au 23 octobre
Scorpion	du 24 octobre au 22 novembre
Sagittaire	du 23 Novembre au 21 Décembre

Par exemple, la commande suivante : `signezodiaque 14 4` a pour effet d'afficher et de placer dans la variable **SIGNE** la valeur **Belier** car le 14/04 correspond bien au signe du zodiacal du **Bélier**.

Pour calculer le signe zodiacal, on pourra définir des variables dont la valeur dépend du mois, et faire le calcul final à l'aide de ces variables.

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 que 20 il faut procéder de la manière suivante :

a) calculer la différence entre **A** et 20 dans une variable intermédiaire : `set /a d=%A%-20`

b) si le résultat est négatif, c'est-à-dire si le résultat commence par le caractère "-", cela signifie que  $A < 20$  ; dans le cas contraire, on a  $A \geq 20$

On utilisera cette méthode pour déterminer le signe zodiacal d'une date donnée (comparaison du jour de la date donnée avec le jour du mois déterminant le changement de signe zodiacal dans le mois).

La commande `SigneZodiaque.bat` vérifiera la présence des paramètres et affichera un message d'aide si un paramètre est manquant.

**4.2** – Écrire la commande **AfficherSigne.bat** qui affiche le signe zodiacal d'une personne donnée. Les paramètres de la commande sont le prénom, le nom et la date de naissance de la personne donnée sous la forme jour-mois-année.

Par exemple, la commande suivante : `affichersigne Jean Dupont 14-4-1983`  
a pour effet d'afficher le message suivant :

Jean Dupont est né(e) un 14/4 et a pour signe zodiacal Belier

Bien entendu, la commande `AfficherSigne.bat` pourra appeler la commande `SigneZodiaque.bat`

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

Nom:Prénom:date de naissance:adresse rue:ville:code postal

Exemple de fichier `personnes.txt` :

```
#Fichier de personnes
#Nom:Prenom:Date:adresse rue:Ville:Code postal
Galle:Martin:12-2-1989:rue des roses:Grenoble:38000
Honnete:Marie:14-9-1980:avenue du général Shi Paoli Vaopo:Paris:75002
Pleur:Jean:13-3-1974:rue des marronniers:Grenoble:38000
Prane:Dolly:19-11-1995:rue des escaliers:Grenoble:38100
Mieu:Jeff-Edmond:22-12-1992:rue des étangs:Grenoble:38000
```

En utilisant la commande `for`, et l'extraction de chaîne, écrire la commande `CalculerSignes.bat` qui affiche, pour chaque personne du fichier, son signe du zodiaque en utilisant la commande `AfficherSigne.bat`

Paramétrez la commande `CalculerSignes.bat` de sorte que le fichier contenant les informations sur des personnes soit un paramètre de la commande.

**4.4** – Écrire la commande `CreerSignes.bat` qui crée, à partir du fichier `personnes.txt` (premier paramètre de cette commande), le fichier `personnes-signes.txt` (second paramètre de la commande) qui contient des lignes comprenant pour chaque personne : le nom, le prénom, la date de naissance et le signe zodiacal.

Exemple de fichier `personnes-signes.txt` :

```
#Fichier personnes.txt avec signes zodiacaux
#nom:prenom:date naissance:signe zodiacal
Galle:Martin:12-2-1989:Verseau
Honnete:Marie:14-9-1980:Vierge
Pleur:Jean:13-3-1974:Poissons
Prane:Dolly:19-11-1995:Scorpion
Mieu:Jeff-Edmond:22-12-1992:Capricorne
```

**4.5** – Écrire la commande `CompterSignes.bat` qui prend en paramètre un fichier au format du fichier `personnes-signes.txt` et un signe de zodiaque, et affiche le nombre de personnes dont le signe est celui passé en second paramètre.

Par exemple, la commande `Comptersignes personnes-signes.txt Verseau`  
a pour effet d'afficher le message :

Il y a 1 personne(s) de signe Verseau

### 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.