

Exercices en classe sur la manipulation de fichier (module 4)

Solutions

- 1) Créer un fichier qui contiendra les services commençant par la lettre "R".

```
get-service | where-object {$_.Name.startswith("R")} | out-file processR.txt
```

ou

```
get-service R* > processR.txt
```

- 2)

- a) Créer un fichier nommé "process_powershell.txt" qui contiendra le résultat de get-process.

```
get-process | out-file process_powershell.txt
```

ou

```
get-process > process_powershell.txt
```

- b) Faire afficher ensuite toutes les lignes de ce fichier seulement si cette ligne contient le mot Powershell.

```
gc process.txt | where-object{$_ -like "*Powershell"}
```

- 3) Soit le fichier nommé "produits.txt" et qui a le format suivant:

Nom_Produit:Code_Barre: Prix

Tomates:1133322477:0.59

Bananes:2233411223:0.71

Pain:9809211123:3.89

Bieres:1990009112:11.99

- a) Comment puis-je faire afficher uniquement la 3e ligne de ce fichier ?

```
$Fichier = get-content produits.txt
```

```
$Fichier[2]
```

- b) Comment puis-je connaître le nombre de ligne (qui est en fait le nombre d'item) que ce fichier comporte ?

```
$Fichier.Count
```

- c) Comment doit-on écrire l'appel à la méthode "split" pour pouvoir récupérer les différents éléments d'une ligne dans un tableau ?

```
$Ligne = $Fichier[0].split(":")
```

- d) En utilisant un foreach tel que vu en classe, faites afficher "vrai" si le mot Bananes est présent dans le fichier ou "faux" dans le cas contraire.

```
$Resultat = $false
foreach ($Ligne in gc produits.txt)
{
    if ( $Ligne.split(":") -like "*Bananes*")
    {
        $Resultat = $true
    }
}
if ($Resultat -eq $true)
{
    write-host "Bananes a été trouvé"
}
else
{
    write-host "Bananes non trouvé"
}
```

- e) En utilisant les foreach, faites afficher uniquement la première colonne de ce fichier.

```
foreach ($Ligne in gc produits.txt)
{
    $Ligne.split(":")[0]
}
```

- 4) Créez un fichier qui pourra être lu par un navigateur Web et qui contiendra les processus qui utilise plus de 1Mo de mémoire. La page devra avoir le titre "Process eating memory". Le fichier devra faire afficher le nom du processus, le temps CPU et la quantité de mémoire utilisée en ordre décroissant. On devra pouvoir visualiser le tout avec Internet Explorer ou Firefox.

```
Get-Process | where-object {$_.WS -gt 10MB } | sort-object -Descending WS | ConvertTo-Html
WS,Name -title "Process eating memory" > process.html
```