

## Práctica 5a Comandos Básicos W10 CMD y PowerShell.

Realiza las siguientes operaciones básicas utilizando cmd, para ello utiliza la ayuda proporcionada por el comando help. Realiza las correspondientes capturas, e incluyelas aquí:

1.- Accede a la carpeta escritorio del usuario con el que has iniciado la sesión.

```
C:\Users\manuc>cd Desktop  
C:\Users\manuc\Desktop>
```

2.- Crea tres carpetas: dir1, dir2, dir3.

```
C:\Users\manuc\Desktop>mkdir dir1  
C:\Users\manuc\Desktop>mkdir dir2  
C:\Users\manuc\Desktop>mkdir dir3  
C:\Users\manuc\Desktop>
```

3.- Crea un fichero en dir1 llamado arc-dir1.txt, que contenga las palabras:. Realiza una copia de éste en dir2.

```
C:\Users\manuc\Desktop\dir1>copy con arc-dir1.txt  
Texto de prueba para el primer fichero.  
^Z          1 archivo(s) copiado(s).  
C:\Users\manuc\Desktop\dir1>
```

```
C:\Users\manuc\Desktop\dir1>cd ..  
C:\Users\manuc\Desktop>copy C:\Users\manuc\Desktop\dir1\arc-dir1.txt C:\Users\manuc\Desktop\dir2  
1 archivo(s) copiado(s).  
C:\Users\manuc\Desktop>
```

4.- Cambia el nombre de arc-dir1 en el directorio dir2 por arc-dir2.txt.

```
C:\Users\manuc\Desktop\dir2>rename arc-dir1.txt arc-dir2.txt
```

5.- Mueve arc-dir1.txt al directorio dir3 con el nombre arc-dir3.txt.

```
C:\Users\manuc\Desktop\dir1>move arc-dir1.txt C:\Users\manuc\Desktop\dir3\arc-dir3.txt
Se han movido      1 archivos.
```

6.- Elimina el fichero arc-dir2.txt.

```
C:\Users\manuc\Desktop\dir2>del arc-dir2.txt
```

7.- Elimina todas las carpetas creadas.

```
C:\Users\manuc\Desktop>rmdir dir1 dir2 dir3
El directorio no está vacío.
```

```
C:\Users\manuc\Desktop>cd dir3
C:\Users\manuc\Desktop\dir3>del *
C:\Users\manuc\Desktop\dir3\*, ¿Está seguro (S/N)? s
C:\Users\manuc\Desktop\dir3>cd ..
C:\Users\manuc\Desktop>rmdir dir3
```

Abre Powershell y ejecuta las siguientes órdenes, y realizando las capturas correspondientes:

8.- Cambia al directorio “Desktop” del usuario con el que has accedido al sistema.

9.- Crea una carpeta llamada mundo. En ella crea 2 nuevas carpetas llamadas Europa y África.

10.- Crea en la carpeta Europa un fichero llamado alimentos.txt y escribe en su interior: Fruta, leche y carne. Crea otro fichero vacío llamado noHayDinero.txt

11.- Cambia el nombre del fichero alimentos.txt por el nuevo nombre alimentos2.txt y duplica este fichero en la misma carpeta llamando a la copia alimentos3.txt

12.- Mueve los ficheros alimentos2.txt y alimentos3.txt a la carpeta África. Elimina el fichero noHayDinero.txt

Cmdlet ó command-let en PowerShell

Son comandos de una sola función que sirven para manipular objetos en PowerShell. Cuando trabajamos con PowerShell en realidad estamos trabajando con objetos .NET. Se diferencia de los cmd en su sintaxis que es Verbo-Nombre: su formato es un verbo (por ejemplo: get) y un nombre (por ejemplo: help) separados por un guión. Así, un cmdlet de PowerShell es “get-help” que es un comando que nos permite obtener ayuda. Por ejemplo, para obtener ayuda sobre “dir”, escribimos en la consola “get-help dir”. Los verbos utilizados son:

get -> obtener datos

set -> establecer o modificar datos

format -> aplicar formato a los datos

out -> enviar la salida a un destino

Los básicos cmdlet más básicos son:

get-help -> obtener información de ayuda

get-command -> nos muestra por pantalla la lista de comandos:

get-service -> obtener información sobre los servicios y su estado

get-process -> obtener información sobre los procesos

13. Mostrar la fecha y hora del sistema: Con “get-date” el shell nos dará información sobre la fecha y hora:

Si queremos limpiar la pantalla, escribimos “cls”.

14. Usar símbolos comodines en nuestras sintaxis. Por ejemplo, para obtener todos los ficheros ejecutables del sistema escribimos “get-command \*.exe” en la consola y obtendremos una pantalla de 3 columnas que nos indica el tipo de fichero, el nombre y su localización:

15. Personalizar el formato de salida de datos. El cmdlet “format” nos permite aplicar formato a la salida de datos. Por ejemplo, hemos visto anteriormente que el formato de salida por defecto del cmdlet “get-service” es una tabla de 3 columnas para estado, denominación y descripción.

Pues bien, si queremos que en vez de una tabla se muestre una lista de los datos, podemos escribir:

get-service | format-list

La salida de datos es una lista y no una tabla. Además, se puede observar que para darle formato a la salida hemos necesitado el llamado “operador de canalización”, el símbolo “|”, para que la salida del comando de la izquierda se convierta en la entrada del comando escrito a la derecha.

Los cmdlets de format son: format-table, format-list, format-wide y format-custom.

16. Variantes que tenemos con el cmdlet “out”: Sabemos ya que el cmdlet “out” permite redirigir la salida de datos. Cuando utilicemos algunas de sus opciones, necesitaremos aplicar el “operador de canalización”, con el símbolo “|”, para que la acción tenga su efecto en los datos de entrada.

Tenemos cuatro modalidades con el cmdlet “out”, que son:

out-host -> dirige los datos de salida a la pantalla

out-printer -> dirige los datos de salida a la impresora

out-null -> para descartar datos innecesarios producto de una ejecución

out-file -> para almacenar en un fichero los datos de salida

Veamos un ejemplo con “out-host”. Supongamos, que necesitamos dirigir los datos obtenidos sobre los comandos de una manera paginada, entonces podemos aplicar la siguiente sintaxis:

get-command | out-host -paging

Con esto logramos que por pantalla veamos la lista de todos los comandos y de una manera paginada:

17. Obtener la lista de todos los comandos de PowerShell: Con “get-help \*” podemos obtener la lista completa. Usamos el símbolo “\*” (comodín) para seleccionar a todos.