



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



SISTEMAS OPERATIVOS

PROFESOR: CORTES GALICIA JORGE

CICLO ESCOLAR: 2024-2

**PRACTICA 4 – PROGRAMACIÓN BAJO EL INTÉRPRETE DE
COMANDOS (SHELL) EN LINUX Y WINDOWS**

16 DE ABRIL DE 2024

INTRODUCCION

La programación en el intérprete de comandos, conocido como shell, representa una herramienta fundamental para interactuar con un sistema operativo a través de una interfaz de línea de comandos. A diferencia de las interfaces gráficas de usuario, que requieren la interacción con ventanas y menús, el shell permite a los usuarios ejecutar comandos directamente desde una terminal, lo que proporciona una forma rápida y eficiente de realizar diversas tareas del sistema.

Los shells están disponibles en la mayoría de los sistemas operativos, incluidos Linux y Windows, y proporcionan a los usuarios un control directo sobre el sistema. En el entorno Linux, el shell más popular y ampliamente utilizado es Bash (Bourne-Again SHell). Bash es una evolución del shell original de Unix, conocido como Bourne shell, y ofrece un intérprete de comandos completo con una amplia gama de herramientas para programar y automatizar tareas del sistema. Estas herramientas incluyen la ejecución de comandos, la redirección de entrada y salida, la manipulación de variables de entorno, la creación de scripts y la integración con otros programas.

En el caso de Windows, si bien el shell predeterminado es el Command Prompt, que utiliza un lenguaje de comando similar al de Bash pero con diferentes sintaxis y herramientas, Microsoft también ofrece PowerShell, un shell más potente y flexible. PowerShell permite la automatización de tareas del sistema mediante scripts, brindando a los usuarios la capacidad de administrar y controlar sus sistemas de manera eficiente y efectiva.

La programación en el intérprete de comandos resulta especialmente útil para automatizar tareas repetitivas o para llevar a cabo operaciones avanzadas del sistema que no son fáciles de realizar a través de una interfaz gráfica de usuario. Además, los scripts de shell son portátiles entre diferentes sistemas operativos, lo que facilita la creación de scripts una vez y su ejecución en múltiples plataformas, lo cual es una ventaja significativa para los administradores de sistemas y desarrolladores.

SECCIÓN LINUX

1.- A través de la ayuda en línea que proporciona Linux, investigue para qué es utilizado el comando `bash`, mencione sus características principales, dé un resumen de su funcionalidad

Bash, como intérprete de comandos de Unix, destaca como uno de los shells más utilizados en los entornos Unix y Linux. Sus características principales son:

- Amplio soporte para una extensa gama de comandos y utilidades presentes en Unix y Linux.
- Personalización flexible del prompt y la configuración del shell, adaptándose así a las necesidades individuales del usuario.
- Facilita la ejecución de comandos en secuencia o en conjunto mediante tuberías (pipes) y redirecciones.
- Capacidades de programación de scripts, permitiendo el uso de variables, bucles, condicionales y otros elementos esenciales.

El funcionamiento de Bash es directo: tras abrir una terminal o consola de comandos, el usuario introduce sus comandos en el prompt de Bash, que luego son ejecutados por el intérprete de comandos en el sistema operativo. Este poderoso shell habilita al usuario para ejecutar programas, editar archivos, administrar procesos y llevar a cabo tareas de administración de sistemas con eficacia. Además, la capacidad de Bash para ejecutar scripts automatizados resulta invaluable para administradores de sistemas y desarrolladores.

2.- Revise en la ayuda en línea de Linux los siguientes comandos: `date`, `tar`, `bzip2`, `bunzip2`. Dé un resumen de sus características y funcionalidad.

Date

Características:

- Permite visualizar la fecha y hora actuales en una variedad de formatos.
- Facilita la configuración de la fecha y hora del sistema a un momento específico.
- Ofrece soporte para la conversión de fechas y horas entre distintos formatos.

Funcionamiento:

El funcionamiento básico del comando `date` es sencillo, al ejecutarlo sin argumentos, muestra la fecha y hora actual en el formato por defecto del sistema. No obstante, si se requiere visualizar la fecha y hora en un formato particular, es posible utilizar argumentos adicionales para lograrlo.

Tar

Características:

- Permite la creación de archivos de archivo, ya sea comprimidos o sin comprimir.
- Facilita la adición de archivos a un archivo de archivo existente o la extracción de archivos de un archivo de archivo preexistente.
- Permite la compresión y descompresión de archivos utilizando una variedad de algoritmos de compresión.
- Puede ser utilizado para realizar copias de seguridad y restauraciones de datos de manera efectiva.

Funcionamiento:

El funcionamiento básico del comando *tar* para la creación de un archivo de archivo implica el uso del argumento *cvf*, seguido del nombre del archivo que se desea crear y los nombres de los archivos o directorios que se desean incluir en dicho archivo.

Bzip2

Características y funcionamiento:

- Compresión: Para comprimir un archivo, simplemente se utiliza el comando *bzip2* seguido del nombre del archivo. Por ejemplo: *bzip2 archivo*. Esto creará un archivo comprimido con extensión ".bz2" en el mismo directorio que el archivo original.
- Descompresión: Para descomprimir un archivo comprimido con *bzip2*, se emplea el comando *bzip2 -d archivo.bz2*. Esto restaurará el archivo original en el mismo directorio que el archivo comprimido.
- Compresión múltiple: *bzip2* permite la compresión de múltiples archivos simultáneamente. Para ello, se lista el nombre de los archivos que se desean comprimir después del comando *bzip2*. Por ejemplo: *bzip2 archivo1 archivo2 archivo3*. Esto generará archivos comprimidos separados para cada archivo de entrada.
- Opciones de nivel de compresión: Se pueden ajustar las opciones de compresión para controlar el nivel de compresión. El nivel de compresión predeterminado es 6, pero se puede aumentar o disminuir según las necesidades. Por ejemplo, *bzip2 -9 archivo* aumentará el nivel de compresión, resultando en un archivo más pequeño pero con un proceso de compresión más lento.
- Verificación de integridad: *bzip2* también ofrece la capacidad de verificar la integridad de los archivos comprimidos. Utilizando el comando *bzip2 -t archivo.bz2*, se puede verificar si el archivo comprimido está dañado o no.

Bunzip2


Características:

- bunzip2 es una herramienta de línea de comandos que se utiliza para descomprimir archivos que han sido comprimidos utilizando bzip2.
- Este comando es similar al comando gunzip, que se utiliza para descomprimir archivos que han sido comprimidos utilizando el algoritmo de compresión gzip.
- bunzip2 es capaz de descomprimir archivos que han sido comprimidos con bzip2 y bzip, pero no es capaz de descomprimir archivos que han sido comprimidos con otros algoritmos de compresión, como gzip o zip.
- Este comando es capaz de manejar archivos grandes y comprimirlos de manera eficiente.
- bunzip2 no elimina el archivo original después de descomprimirlo, por lo que el archivo comprimido y el archivo descomprimido estarán presentes en el sistema después de la operación.

Funcionamiento:

El funcionamiento básico del comando *bunzip2* es simple: se proporciona el nombre del archivo comprimido como argumento al comando. Por ejemplo, para descomprimir un archivo llamado *archivo.tar.bz2*, simplemente se ejecuta *bunzip2 archivo.tar.bz2*. Esto restaurará el archivo original *archivo.tar* en el sistema.

3.- Capture y ejecute el siguiente script. Observe su funcionamiento y explique.



```
1  #!/bin/sh
2  #
3  clear
4  #
5  ls -l
6  #
7  rm -f ./prueba
8  #
9  ls -l
10 #
11 echo "----- Creando archivo: prueba -----"
12 #
13 echo "Hola a todos" >> prueba
14 echo "Esto es una prueba" >> prueba
15 echo "De un script" >> prueba
16 #
17 ls -l ./prueba
18 #
19 echo "----- Script terminado -----"
20 more ./prueba
```

Para ejecutar el script cambie los permisos de su archivo para permitir ejecución a nivel dueño. Ejecute como un programa normal en C.

```
total 4
-rwxrwxr-- 1 vboxuser vboxuser 316 abr 14 22:14 Prueba.sh
total 4
-rwxrwxr-- 1 vboxuser vboxuser 316 abr 14 22:14 Prueba.sh
----- Creando archivo: prueba -----
-rw-rw-r-- 1 vboxuser vboxuser 45 abr 15 23:03 ./prueba
----- Script terminado -----
Hola a todos
Esto es una prueba
De un script
vboxuser@UBUNTU:~/Escritorio/PRACTICA 4$
```

Este script de shell primero limpia la pantalla, muestra el contenido del directorio actual, luego borra un archivo llamado "prueba", crea uno nuevo con ese mismo nombre y escribe algunas líneas de texto en él. Después muestra información detallada sobre el nuevo archivo y su contenido. Finalmente, muestra un mensaje de que el script ha terminado y muestra el contenido del archivo "prueba" en la terminal.

4.- Capture y ejecute el siguiente script. Sustituya las rutas utilizadas en el script por las rutas que usted utilice. Observe su funcionamiento y explique.

```
1 #!/bin/sh
2 #
3 clear
4 #
5 echo "----- Obteniendo fecha -----"
6 fecha=$(date +%Y%m%d%H%M)
7 #
8 echo "----- Empacando -----"
9 tar -cvf ./respaldo$fecha.tar /home/vboxuser/Escritorio/directorio1 /home/vboxuser/Escritorio/directorio2 /home/vboxuser/Escritorio/directorio3
10 #
11 echo "----- Comprimiendo -----"
12 bzip2 ./respaldo$fecha.tar
13 #
14 echo "----- Enviando a otro directorio -----"
15 cp ./respaldo$fecha.tar.bz2 /home/vboxuser/Escritorio/respaldos
16 #
17 echo "----- Descomprimiendo -----"
18 bunzip2 /home/vboxuser/Escritorio/respaldos/respaldo$fecha.tar.bz2
19 #
20 echo "----- Desempacando -----"
21 tar -xvf /home/vboxuser/Escritorio/respaldos/respaldo$fecha.tar
22 #
23 echo "----- Limpiando -----"
24 rm -f ./respaldo$fecha.tar.bz2
25 rm -f ./respaldo$fecha.tar
26 #
27 echo "----- Terminado -----"
```

```
----- Obteniendo fecha -----
----- Empacando -----
tar: Eliminando la '/' inicial de los nombres
/home/vboxuser/Escritorio/directorio1/
/home/vboxuser/Escritorio/directorio1/directorio_1.txt
tar: Eliminando la '/' inicial de los objetivos de los enlaces
/home/vboxuser/Escritorio/directorio2/
/home/vboxuser/Escritorio/directorio2/directorio_2.txt
/home/vboxuser/Escritorio/directorio3/
/home/vboxuser/Escritorio/directorio3/directorio_3.txt
----- Comprimiendo -----
----- Enviando a otro directorio -----
----- Descomprimiendo -----
Desempacando
/home/vboxuser/Escritorio/directorio1/
/home/vboxuser/Escritorio/directorio1/directorio_1.txt
/home/vboxuser/Escritorio/directorio2/
/home/vboxuser/Escritorio/directorio2/directorio_2.txt
/home/vboxuser/Escritorio/directorio3/
/home/vboxuser/Escritorio/directorio3/directorio_3.txt
----- Limpiando -----
----- Terminado -----
vboxuser@UBUNTU:~/Escritorio/PRACTICA4$
```

```
vboxuser@UBUNTU:~/Escritorio/PRACTICA4/home/vboxuser/Escritorio$ ls -R
.:
directorio1 directorio2 directorio3
./directorio1:
directorio_1.txt
./directorio2:
directorio_2.txt
./directorio3:
directorio_3.txt
vboxuser@UBUNTU:~/Escritorio/PRACTICA4/home/vboxuser/Escritorio$
```



respaldo20
240415235
1.tar

Este script de shell automatiza el proceso de respaldo de varios directorios. Primero, obtiene la fecha y hora actual y crea un archivo tar que contiene los directorios especificados. Luego, comprime este archivo usando bzip2 y lo copia a otro directorio. Después, descomprime el archivo en el directorio de respaldo y elimina los archivos temporales generados durante el proceso. Finalmente, muestra un mensaje indicando que el proceso ha terminado.

5.- Capture y ejecute el siguiente script. Observe su funcionamiento.



```
1  #!/bin/sh
2  clear
3  echo -n "Introduce dos valores numéricos separados por espacios"
4  read var1 var2
5  echo "Los valores son: "
6  echo var1=$var1
7  echo var2=$var2
8  #
9  if [ $var1 -lt $var2 ]
10 then
11     echo "$var1 menor que $var2"
12     while [ $var1 -lt $var2 ]
13     do
14         echo contador=$var1
15         var1=`expr $var1 + 1`
16     done
17 else
18     echo "$var1 mayor o igual que $var2"
19     case $var2 in
20         10) echo "Fui un 10";;
21         20) echo "Fui un 20";;
22         Otro) echo "Otro valor";;
23     esac
24 fi
```

```
-n "Introduce dos valores numéricos separados por espacios"
10 15
"Los valores son: "
var1=10
var2=15
"10 menor que 15"
contador=10
contador=11
contador=12
contador=13
contador=14
vboxuser@UBUNTU:~/Escritorio/PRACTICA4$
```

```
-n "Introduce dos valores numéricos separados por espacios"
15 10
"Los valores son: "
var1=15
var2=10
"15 mayor o igual que 10"
"Fui un 10"
vboxuser@UBUNTU:~/Escritorio/PRACTICA4$
```

Este script de shell solicita al usuario dos valores numéricos, luego los muestra y realiza diferentes acciones según la relación entre estos valores. Si el primer valor es menor que el segundo, el script muestra cada número en un rango ascendente hasta el segundo valor. Si el primer valor es mayor o igual que el segundo, el script imprime un mensaje específico dependiendo del valor del segundo número: "Fui un 10" si es 10, "Fui un 20" si es 20, o "Otro valor" si es diferente.

6. Capture y ejecute el siguiente script. Observe su funcionamiento.



```
1  #!/bin/sh
2  clear
3  echo "Opcion: "
4  select i in Directorio DirectorioExtendido Salir
5  do
6      case $i in
7          Directorio) ls;;
8          DirectorioExtendido) ls -l;;
9          Salir) break;;
10     esac
11 done
12 #
13 for i in hola como estan todos
14 do
15     echo -n "$i "
16 done
```

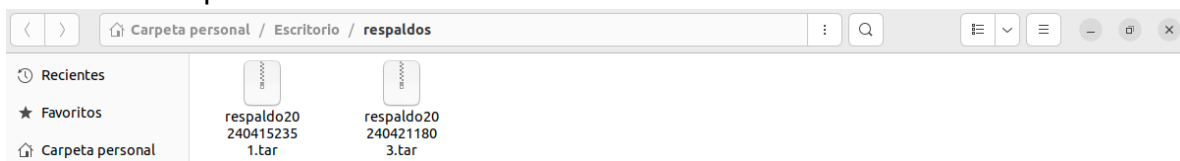
```
Opcion:
1) Directorio
2) DirectorioExtendido
3) Salir
#? 1
Comprimir.sh      Directorio.sh  Eliminar.sh      home              Operaciones.sh  Prueba.sh          script_generado.sh
Descomprimir.sh  Eliminar      Generador.sh     InfoDirectorio.sh prueba          respaldo202404142337.tar.bz2
#? 2
total 52
-rwxrwxr-- 1 vboxuser vboxuser 914 abr 14 22:53 Comprimir.sh
-rwxrwxr-- 1 vboxuser vboxuser 895 abr 14 23:40 Descomprimir.sh
-rwxrwxr-- 1 vboxuser vboxuser 266 abr 16 00:09 Directorio.sh
drwxrwxr-x 2 vboxuser vboxuser 4096 abr 15 00:53 Eliminar
-rwxrwxr-x 1 vboxuser vboxuser 467 abr 15 01:02 Eliminar.sh
-rwxrwxr-- 1 vboxuser vboxuser 664 abr 15 00:01 Generador.sh
drwxrwxr-x 3 vboxuser vboxuser 4096 abr 14 23:37 home
-rwxrwxr-- 1 vboxuser vboxuser 715 abr 15 00:44 InfoDirectorio.sh
-rwxrwxr-- 1 vboxuser vboxuser 594 abr 14 23:07 Operaciones.sh
-rw-rw-r-- 1 vboxuser vboxuser 45 abr 15 23:03 prueba
-rwxrwxr-- 1 vboxuser vboxuser 316 abr 14 22:14 Prueba.sh
-rw-rw-r-- 1 vboxuser vboxuser 279 abr 14 23:37 respaldo202404142337.tar.bz2
-rwxrwxr-x 1 vboxuser vboxuser 60 abr 15 00:04 script_generado.sh
#? 3
hola como estan todos vboxuser@UBUNTU:~/Escritorio/PRACTICA4$
```

Este script de shell proporciona un menú interactivo con tres opciones: "Directorio", "DirectorioExtendido" y "Salir". El usuario puede seleccionar una opción escribiendo su número correspondiente. Dependiendo de la opción seleccionada, el script muestra el contenido del directorio actual con el comando ls si se elige "Directorio", o muestra el contenido en formato largo con ls -l si se elige "DirectorioExtendido". Si se elige "Salir", el script termina. Después del menú, el script imprime las palabras "hola como estan todos" en una línea.

7.- Programe un script para que descomprima en sus rutas originales los directorios que se han compreso con el anterior script

```
1  #!/bin/bash
2  #
3  clear
4  #
5  echo "----- Obteniendo archivo más reciente -----"
6  #
7  directorio="/home/vboxuser/Escritorio/respaldos"
8  #
9  archivo=$(ls -t "$directorio"/*.tar | head -1)
10 #
11 if [ -z "$archivo" ]; then
12     echo "No se encontraron archivos tar en el directorio especificado."
13     exit 1
14 fi
15 #
16 echo "El archivo más reciente es: $archivo"
17 #
18 echo "----- Descomprimiendo -----"
19 #
20 ruta_destino="/home/vboxuser/Escritorio/PRACTICA4/Descromprimidos"
21 #
22 mkdir -p "$ruta_destino"
23 #
24 tar -xf "$archivo" -C "$ruta_destino"
25 #
26 echo "----- Terminado -----"
```

Archivos comprimidos:



Descompresión del ultimo archivo comprimido:



8.- Programe un script generador de scripts, es decir, que cree un archivo, que posteriormente permita editarlo y que finalmente cambie los permisos para ejecución del archivo, mandando a ejecutar el script desde el script original.

```
1  #!/bin/sh
2  #
3  clear
4  # Crear el archivo de script
5  echo "----- Creando archivo-----"
6  echo "#!/bin/sh" > script_generado.sh
7  echo "echo "Hola, mundo!"" >> script_generado.sh
8
9  # Dar permisos de edicion y ejecucion
10 echo "----- Dando permisos -----"
11 chmod +x script_generado.sh
12
13 # Abrir el archivo de script para edicion
14 echo "----- Abro Nano para editar -----"
15 nano script_generado.sh
16
17 # Dar permisos de ejecucion
18 echo "----- Permisos de ejecucion -----"
19 chmod +x script_generado.sh
20
21 # Ejecutar script generado
22 echo "----- Ejecutando el nuevo script -----"
23 ./script_generado.sh
```



```
----- Creando archivo-----
----- Dando permisos -----
----- Abro Nano para editar -----
----- Permisos de ejecucion -----
----- Ejecutando el nuevo script -----
Hola, mundo!
Este código se edita desde GNU nano 6.2
vboxuser@UBUNTU:~/Escritorio/PRACTICA4$
```

9.- Programe un script a través de las estructuras de control de flujo para que permita visualizar los archivos de un directorio que se envíe como argumento de entrada (no utilizar el comando ls). Además, el script deberá mostrar el número total de archivos en el directorio.



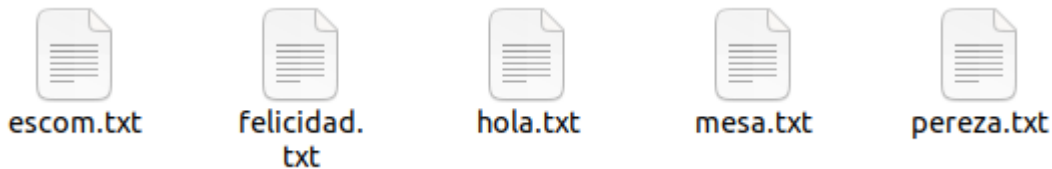
```
1  #!/bin/sh
2  #
3  clear
4
5  # Verificar que se haya ingresado un argumento de entrada
6  if [ $# -eq 0 ]
7      then
8          echo "Debe ingresar un directorio como argumento de entrada"
9          exit 1
10 fi
11
12 # Verificar que el argumento de entrada sea un directorio valido
13 if [ ! -d $1 ]
14     then
15         echo "$1 no es un directorio valido."
16         exit 1
17 fi
18
19 # Contar el numero de archivos en el directorio
20 num_archivos=$(find $1 -maxdepth 1 -type f | wc -l)
21
22 # Mostrar el numero total de archivos en el directorio
23 echo "El directorio $1 contiene $num_archivos archivos."
24
25 # Mostrar los archivos del directorio
26 echo "Archivos en $1:"
27 for archivo in $1/*; do
28     if [ -f "$archivo" ]; then
29         echo "${archivo##*/}"
30     fi
31 done
```

```
El directorio /home/vboxuser/Escritorio/PRACTICA4 contiene 11 archivos.
Archivos en /home/vboxuser/Escritorio/PRACTICA4:
Comprimir.sh
Descomprimir.sh
Directorio.sh
Eliminar.sh
Generador.sh
InfoDirectorio.sh
Operaciones.sh
Prueba.sh
prueba
respaldo202404142337.tar.bz2
script_generado.sh
vboxuser@UBUNTU:~/Escritorio/PRACTICA4$
```

10.- Programe un script que elimine todos los archivos de un directorio especificado desde el propio script, y cuyo segundo carácter sea la letra “e”.

```
1  #!/bin/bash
2
3  clear
4
5  # Ruta a evaluar
6  ruta_destino="/home/vboxuser/Escritorio/Eliminar"
7
8  # Eliminar los archivos del directorio que cumplan con el criterio
9  echo "Eliminando archivos con segundo carácter igual a 'e' en $ruta_destino..."
10 for archivo in $ruta_destino/*; do
11     nombre=${archivo##*/}
12     if [ -f "$archivo" ] && [ "${nombre:1:1}" = "e" ]; then
13         echo "Eliminando archivo ${archivo##*/}"
14         rm "$archivo"
15     fi
16 done
17
18 echo "Proceso finalizado"
```

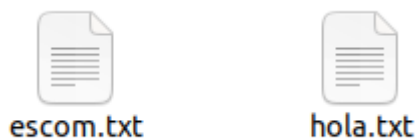
Archivos con distintos nombres:



El programa elimina los que tienen en la segunda letra una ‘e’:

```
Eliminando archivos con segundo carácter igual a 'e' en /home/vboxuser/Escritorio/Eliminar...
Eliminando archivo felicidad.txt
Eliminando archivo mesa.txt
Eliminando archivo pereza.txt
Proceso finalizado
vboxuser@UBUNTU:~/Escritorio/PRACTICA4$
```

El directorio solo queda con estos archivos:



SECCIÓN WINDOWS

1.- Investigue los siguientes comandos: echo, del, date, tar, rar. Dé un resumen de sus características y funcionalidad.

echo: Este comando se utiliza para mostrar mensajes en la consola o para activar y desactivar la característica de eco de comandos. Cuando se ejecuta sin parámetros, *echo* muestra la configuración actual de la característica de eco. El eco de comandos es la capacidad del sistema para mostrar en pantalla los comandos que se ingresan antes de su ejecución.

del: *del* es un comando que se emplea para eliminar uno o varios archivos del sistema. Esencialmente, realiza la misma función que el comando *erase*. Al ejecutar *del*, se elimina de manera permanente el archivo especificado, sin enviarlo a la papelera de reciclaje o a una carpeta de recuperación.

date: Este comando se utiliza para mostrar la fecha actual del sistema o para establecer una nueva fecha. Cuando se ejecuta sin parámetros, *date* simplemente muestra la fecha actual del sistema y solicita al usuario que ingrese una nueva fecha si desea modificarla. Es una herramienta fundamental para administrar y mantener la precisión temporal del sistema.

tar: *tar* es un comando utilizado para archivar y recuperar archivos en y desde un único archivo, conocido como "tarfile". Aunque su nombre se deriva de "tape archive" (archivo de cinta), el *tarfile* puede ser cualquier tipo de archivo en el sistema. Permite la compresión y la agrupación de múltiples archivos y directorios en un solo archivo, facilitando su transporte o respaldo.

rar: Este comando se encarga de comprimir o descomprimir archivos directamente desde la consola de comandos. La herramienta *rar* es especialmente útil para la compresión de archivos grandes o la creación de archivos comprimidos con múltiples archivos y directorios. Su funcionalidad versátil lo convierte en una herramienta valiosa para la gestión de archivos en entornos de línea de comandos.

2. Capture y ejecute el siguiente batch. Guarde el archivo con extensión `.bat`. Observe su funcionamiento y explique.

```
1  @echo off
2  rem
3  cls
4  rem
5  dir .\prueba
6  rem
7  del .\prueba
8  rem
9  dir .\prueba
10 rem
11 echo "----- Creando archivo: prueba -----"
12 rem
13 echo "Hola a todos" >> prueba
14 echo "Esto es una prueba" >> prueba
15 echo "De un batch" >> prueba
16 rem
17 dir .\prueba
18 rem
19 echo "----- Script terminado -----"
20 type .\prueba
```

Para ejecutar escriba desde la línea de comando `>nombre_archivo.bat`.

```
El volumen de la unidad C es Windows
El número de serie del volumen es: 5E36-58DE

Directorio de C:\Users\jarro\OneDrive\Escritorio\PRACTICA4
15/04/2024  07:44 p. m.           56 prueba
               1 archivos           56 bytes
               0 dirs  45,952,851,968 bytes libres
El volumen de la unidad C es Windows
El número de serie del volumen es: 5E36-58DE

Directorio de C:\Users\jarro\OneDrive\Escritorio\PRACTICA4
No se encuentra el archivo
"----- Creando archivo: prueba -----"
El volumen de la unidad C es Windows
El número de serie del volumen es: 5E36-58DE

Directorio de C:\Users\jarro\OneDrive\Escritorio\PRACTICA4
16/04/2024  12:14 a. m.           56 prueba
               1 archivos           56 bytes
               0 dirs  45,952,851,968 bytes libres
"----- Script terminado -----"
"Hola a todos"
"Esto es una prueba"
"De un batch"

C:\Users\jarro\OneDrive\Escritorio\PRACTICA4>|
```

Este script de lotes para Windows inicia limpiando la pantalla y luego busca un archivo llamado "prueba" en el directorio actual. Si encuentra este archivo, lo elimina y muestra el contenido actualizado del directorio. Luego, imprime un mensaje indicando que está creando el archivo "prueba" y agrega tres líneas de texto al mismo. Después, muestra la información actualizada del archivo y finaliza el script mostrando su contenido completo en la terminal.

3. Capture y ejecute el siguiente batch. Sustituya las rutas utilizadas en el batch por las rutas que usted utilice. Deberá tener instalado el programa Winrar para utilizar el comando rar en el script. Observe su funcionamiento y explique.

```
1 @echo off
2 cls
3 echo "----- Captura fecha -----"
4 for /F "tokens=1,2,3 delims=" %X in ('date /t') do set dia=%X/%X/%X
5 set tiempo=%time%
6 for /F "tokens=1,2,3 delims=" %X in ("%time%") do set tiempo=%X/%X/%X
7 set fecha=%dia%%tiempo%
8 echo "----- Empacando -----"
9 tar -cvf ".\respaldo%fecha%.tar" "C:\Users\jarro\OneDrive\Escritorio\directorio1" "C:\Users\jarro\OneDrive\Escritorio\directorio2" "C:\Users\jarro\OneDrive\Escritorio\directorio3"
10 echo "----- Comprimiendo -----"
11 "C:\Program Files\WinRAR\rar.exe" a -c- ".\respaldo%fecha%.rar" ".\respaldo%fecha%.tar"
12 echo "----- Enviando a otro directorio -----"
13 copy ".\respaldo%fecha%.rar" "C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldos"
14 echo "----- Descomprimiendo -----"
15 "C:\Program Files\WinRAR\rar.exe" e -y "C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldos\respaldo%fecha%.rar" "C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldos"
16 echo "----- Desempaquetando -----"
17 tar -xvf "C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldos\respaldo%fecha%.tar" -C "C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldos"
18 echo "----- Limpiando -----"
19 del ".\respaldo%fecha%.rar"
20 del ".\respaldo%fecha%.tar"
21 echo "----- Terminado -----"
```

```
*----- Captura fecha -----*
*----- Empacando -----*
tar: Removing leading drive letter from member names
a Users\jarro\OneDrive\Escritorio\directorio1
a Users\jarro\OneDrive\Escritorio\directorio1\directorio1.txt
a Users\jarro\OneDrive\Escritorio\directorio2
a Users\jarro\OneDrive\Escritorio\directorio2\directorio2.txt
a Users\jarro\OneDrive\Escritorio\directorio3
a Users\jarro\OneDrive\Escritorio\directorio3\directorio3.txt
*----- Comprimiendo -----*
RAR 6.24 x64 Copyright (c) 1993-2023 Alexander Roshal 3 Oct 2023
Versión de evaluación Escribe 'rar -?' para ayuda
Copia de evaluación. Regístrate, por favor.
Creando el archivo .\respaldo16042024 02052 .rar
Añadiendo .\respaldo16042024 02052 .tar Aceptar
Listo
*----- Enviando a otro directorio -----*
1 archivo(s) copiado(s).
*----- Descomprimiendo -----*
RAR 6.24 x64 Copyright (c) 1993-2023 Alexander Roshal 3 Oct 2023
Versión de evaluación Escribe 'rar -?' para ayuda
Extracting from C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldos\respaldo16042024 02052 .rar
Extrayendo C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldos\respaldo16042024 02052 .tar Aceptar
Todo correcto
*----- Desempaquetando -----*
x Users\jarro\OneDrive\Escritorio\directorio1\
x Users\jarro\OneDrive\Escritorio\directorio1\directorio1.txt
x Users\jarro\OneDrive\Escritorio\directorio2\
x Users\jarro\OneDrive\Escritorio\directorio2\directorio2.txt
x Users\jarro\OneDrive\Escritorio\directorio3\
x Users\jarro\OneDrive\Escritorio\directorio3\directorio3.txt
*----- Limpiando -----*
*----- Terminado -----*
C:\Users\jarro\OneDrive\Escritorio\PRACTICA4>
```

```
Directorio de C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldos
16/04/2024 12:20 a. m. <DIR> .
16/04/2024 12:20 a. m. <DIR> .
202 respaldo16042024 02052 .rar
16/04/2024 12:20 a. m. <DIR> .
4,096 respaldo16042024 02052 .tar
16/04/2024 12:20 a. m. <DIR> .
2 archivos 4,378 bytes

Directorio de C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldos\Users
16/04/2024 12:20 a. m. <DIR> .
16/04/2024 12:20 a. m. <DIR> .
jarro
0 archivos 0 bytes

Directorio de C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldos\Users\jarro
16/04/2024 12:20 a. m. <DIR> .
16/04/2024 12:20 a. m. <DIR> .
OneDrive
0 archivos 0 bytes

Directorio de C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldos\Users\jarro\OneDrive
16/04/2024 12:20 a. m. <DIR> .
16/04/2024 12:20 a. m. <DIR> .
Escritorio
0 archivos 0 bytes

Directorio de C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldos\Users\jarro\OneDrive\Escritorio
16/04/2024 12:20 a. m. <DIR> .
16/04/2024 12:20 a. m. <DIR> .
directorio1
15/04/2024 08:06 p. m. <DIR> .
directorio2
15/04/2024 08:06 p. m. <DIR> .
directorio3
0 archivos 0 bytes

Directorio de C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldos\Users\jarro\OneDrive\Escritorio\directorio1
15/04/2024 08:06 p. m. <DIR> .
16/04/2024 12:20 a. m. <DIR> .
0 directorio1.txt
15/04/2024 08:06 p. m. <DIR> .
1 archivos 0 bytes

Directorio de C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldos\Users\jarro\OneDrive\Escritorio\directorio2
15/04/2024 08:06 p. m. <DIR> .
16/04/2024 12:20 a. m. <DIR> .
0 directorio2.txt
15/04/2024 08:06 p. m. <DIR> .
1 archivos 0 bytes

Directorio de C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldos\Users\jarro\OneDrive\Escritorio\directorio3
15/04/2024 08:06 p. m. <DIR> .
16/04/2024 12:20 a. m. <DIR> .
0 directorio3.txt
15/04/2024 08:06 p. m. <DIR> .
1 archivos 0 bytes

Total de archivos en la lista:
1 archivos 4,378 bytes
23 dirs 40,897,621 bytes libres
```

Este script de lote (batch) para Windows automatiza el proceso de respaldo de varios directorios. Primero, captura la fecha y la hora actual y las utiliza para generar un nombre único para el respaldo. Luego, utiliza el comando tar para empaquetar los directorios especificados en un archivo tar. A continuación, utiliza el programa WinRAR para comprimir el archivo tar en formato RAR. Posteriormente, copia el archivo comprimido a otro directorio. Después, descomprime el archivo RAR en el directorio de respaldo. Finalmente, desempaqueta el archivo tar dentro del directorio de respaldo, elimina los archivos temporales generados durante el proceso y muestra un mensaje indicando que el proceso ha terminado.

4.- Capture y ejecute el siguiente script. Observe su funcionamiento.



```
1 @echo off
2 set /p Var1="Introduce valor del primer numero: "
3 set /p Var2="Introduce valor del segundo numero: "
4 if %Var1% LSS %Var2% echo %Var1% es menor que %Var2% & goto :seguir_if
5 echo %Var1% es mayor o igual que %Var2%
6 if %Var1% EQU 10 echo Fui un 10 & goto :salir
7 if %Var1% EQU 20 echo Fui un 20 & goto :salir
8 echo Fui otro valor
9 goto :salir
10 :seguir_if
11 for /L %%V IN (%Var1%,1,%Var2%) do (call :imprime_valor %%V)
12 goto :salir
13
14 :imprime_valor
15 echo contador=%1
16 goto :eof
17 :salir
18 pause
```

```
C:\Users\jarro\OneDrive\Escritorio\PRACTICA4>Operaciones.bat
Introduce valor del primer numero: 15
Introduce valor del segundo numero: 20
15 es menor que 20
contador=15
contador=16
contador=17
contador=18
contador=19
contador=20
Presione una tecla para continuar . . .
```

```
C:\Users\jarro\OneDrive\Escritorio\PRACTICA4>Operaciones.bat
Introduce valor del primer numero: 20
Introduce valor del segundo numero: 15
20 es mayor o igual que 15
Fui un 20
Presione una tecla para continuar . . .
```

Este script de lotes para Windows solicita al usuario que ingrese dos números. Luego, compara los números ingresados y ejecuta diferentes acciones según las comparaciones. Si el primer número es menor que el segundo, muestra todos los números en el rango entre ellos. Si el primer número es mayor o igual que el segundo, comprueba si el primer número es igual a 10 o 20, imprimiendo un mensaje correspondiente en cada caso. Si el primer número no es ni 10 ni 20, imprime un mensaje indicando que fue otro valor. Después de ejecutar las acciones correspondientes, el script se detiene y espera a que el usuario presione una tecla para salir.

5.- Capture y ejecute el siguiente script. Observe su funcionamiento.



```
1 @echo off
2 echo Opciones:
3 echo Directorio
4 echo DirectorioExtendido
5 echo Salir
6 set /p Op="Opcion: "
7 set fin=0
8 for %%V IN (Directorio DirectorioExtendido Salir) do (call :opcion_elegida %%V, %Op%)
9 if %fin%==0 echo "Opcion no encontrada"
10 goto :salir
11 :opcion_elegida
12 if "%2"=="%1" if %1==Directorio dir & set fin=1 & goto :EOF else (
13 if "%2"=="%1" if %1==DirectorioExtendido echo "Similar a dir" & set fin=1 & goto :EOF else (
14 if "%2"=="%1" if %1==Salir set fin=1 & goto :EOF ))
15 goto :EOF
16 :salir
17 for %%V in (Hola como estan todos) do echo %%V
18 pause
```



```

C:\Users\jarro\OneDrive\Escritorio\PRACTICA4>Directorio.bat
Opciones:
Directorio
DirectorioExtendido
Salir
Opcion: Directorio
El volumen de la unidad C es Windows
El número de serie del volumen es: 5E36-58DE

Directorio de C:\Users\jarro\OneDrive\Escritorio\PRACTICA4

16/04/2024 12:20 a. m. <DIR> .
15/04/2024 10:59 p. m. <DIR> ..
15/04/2024 10:05 p. m. 1,058 Compresor.bat
15/04/2024 11:37 p. m. 1,395 Comprimir.bat
15/04/2024 08:39 p. m. 583 Directorio.bat
15/04/2024 09:00 p. m. 3 ejecutable
15/04/2024 09:39 p. m. <DIR> eliminar
15/04/2024 09:38 p. m. 276 Eliminar.bat
15/04/2024 09:00 p. m. 754 Generador.bat
15/04/2024 09:04 p. m. 414 InfoDirectorio.bat
15/04/2024 08:38 p. m. 521 Operaciones.bat
16/04/2024 12:14 a. m. 56 prueba
15/04/2024 07:43 p. m. 360 Prueba.bat
16/04/2024 12:20 a. m. <DIR> respaldos
10 archivos 5,420 bytes
4 dirs 45,940,453,376 bytes libres

Hola
como
están
todos
Presione una tecla para continuar . . .

C:\Users\jarro\OneDrive\Escritorio\PRACTICA4>Directorio.bat
Opciones:
Directorio
DirectorioExtendido
Salir
Opcion: DirectorioExtendido
"Similar a dir"
Hola
como
están
todos
Presione una tecla para continuar . . .

C:\Users\jarro\OneDrive\Escritorio\PRACTICA4>

```

Este script de lotes para Windows presenta un menú con tres opciones: "Directorio", "DirectorioExtendido" y "Salir". El usuario elige una opción ingresando su nombre, y el script ejecuta acciones correspondientes a esa opción. Si se elige "Directorio", muestra el contenido del directorio actual. Si se elige "DirectorioExtendido", imprime "Similar a dir". Si se elige "Salir", termina el script. Si ninguna opción coincide con la entrada del usuario, se muestra "Opción no encontrada". Luego, el script imprime las palabras "Hola", "como", "están" y "todos" en líneas separadas antes de pausar la ejecución y esperar a que el usuario presione una tecla para salir.

6.- Programe un script para que descomprima en sus rutas originales los directorios que se han compreso con el anterior script



```
1 @echo off
2 cls
3 echo "----- Obteniendo archivo más reciente -----"
4 set "directorio=C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldos"
5 for /f "delims=" %%a in ('dir /b /od /a-d "%directorio%\*.tar" 2^>nul') do set "archivo=%%~fa"
6 if "%archivo%"==" " (
7     echo No se encontraron archivos tar en el directorio especificado.
8     exit /b 1
9 )
10 echo El archivo más reciente es: %archivo%
11 echo "----- Descomprimiendo -----"
12 set "ruta_destino=C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\Descomprimidos"
13
14 if not exist "%ruta_destino%" (
15     mkdir "%ruta_destino%"
16 )
17 "%ProgramFiles%\WinRAR\WinRAR.exe" e -y "%archivo%" "%ruta_destino%"
18 echo "----- Terminado -----"
19
```

```
"----- Obteniendo archivo más reciente -----"
El archivo más reciente es: C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\respaldo21042024 182346.tar
"----- Descomprimiendo -----"
"----- Terminado -----"

C:\Users\jarro\OneDrive\Escritorio\PRACTICA4>
```

Escritorio > PRACTICA4 > Descomprimidos > Users > jarro > OneDrive > Escritorio >			
Ordenar Ver ...			
Nombre	Fecha de modificación	Tipo	Tamaño
directorio1	21/04/2024 06:56 p. m.	Carpeta de archivos	
directorio2	21/04/2024 06:56 p. m.	Carpeta de archivos	
directorio3	21/04/2024 06:56 p. m.	Carpeta de archivos	

```
C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\Descomprimidos\Users\jarro\OneDrive\Escritorio>dir /b /s /a
C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\Descomprimidos\Users\jarro\OneDrive\Escritorio\directorio1
C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\Descomprimidos\Users\jarro\OneDrive\Escritorio\directorio2
C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\Descomprimidos\Users\jarro\OneDrive\Escritorio\directorio3
C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\Descomprimidos\Users\jarro\OneDrive\Escritorio\directorio1\directorio1.txt
C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\Descomprimidos\Users\jarro\OneDrive\Escritorio\directorio2\directorio2.txt
C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\Descomprimidos\Users\jarro\OneDrive\Escritorio\directorio3\directorio3.txt
```

7.- Programe un script generador de scripts, es decir, que cree un archivo, que posteriormente permita editarlo y que finalmente cambie los permisos para ejecución del archivo, mandando a ejecutar el script desde el script original.



```
1 @echo off
2 echo "----- Extension del archivo -----"
3 set/p extension= "Ingrese la extension del archivo: "
4 set mensaje=Creando un archivo %extension%
5 echo.
6 timeout /t 1 /nobreak > nul
7 echo "----- %mensaje% -----"
8 echo .>ejecutable.%extension%
9 echo.
10 timeout /t 1 /nobreak > NUL
11 echo "----- Archivo creado -----"
12 echo.
13 timeout /t 1 /nobreak > NUL
14 echo "----- Editando archivo -----"
15 set/p edicion = "Ingrese el contenido del archivo: "
16 echo %edicion%>ejecutable.%extension%
17 echo.
18 timeout /t 1 /nobreak > NUL
19 echo "----- Ejecutando archivo -----"
20 start ejecutable.%extension%
21 echo.
22 echo "----- Finalizado -----"
23 pause>NUL
```

```
"----- Extension del archivo -----"
Ingrese la extension del archivo: .bat

"----- Creando un archivo .bat -----"

"----- Archivo creado -----"

"----- Editando archivo -----"
Ingrese el contenido del archivo: echo "Hola mundo!"

"----- Ejecutando archivo -----"


"----- Finalizado -----"
```

```
Microsoft Windows [Versión 10.0.22631.3447]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\jarro\OneDrive\Escritorio\PRACTICA4>ejecutable.bat
"Hola mundo!"

C:\Users\jarro\OneDrive\Escritorio\PRACTICA4>|
```

8.- Programe un script a través de las estructuras de control de flujo para que permita visualizar los archivos de un directorio que se envíe como argumento de entrada (no utilizar el comando ls). Además, el script deberá mostrar el número total de archivos en el directorio.



```
1  @echo off
2  set /a num_archivos=0
3
4  if "%1"==" " (
5      echo Debe ingresar un directorio como argumento de entrada.
6      goto fin
7  )
8
9  if not exist %1 (
10     echo %1 no es un directorio válido.
11     goto fin
12 )
13
14 echo Archivos en %1:
15 for %%f in (%1\*) do (
16     if %%~xf NEQ "" (
17         echo %%~nxf
18         set /a num_archivos+=1
19     )
20 )
21
22 echo El directorio %1 contiene %num_archivos% archivos.
23
24 :fin
25
```

```

Microsoft Windows [Versión 10.0.22631.3447]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\jarro\OneDrive\Escritorio\PRACTICA4>InfoDirectorio.bat C:\Users\jarro\OneDrive\Escritorio\PRACTICA4
Archivos en C:\Users\jarro\OneDrive\Escritorio\PRACTICA4:
Compresor.bat
Comprimir.bat
Directorio.bat
ejecutable.bat
Eliminar.bat
Generador.bat
InfoDirectorio.bat
Operaciones.bat
prueba
Prueba.bat
El directorio C:\Users\jarro\OneDrive\Escritorio\PRACTICA4 contiene 10 archivos.
C:\Users\jarro\OneDrive\Escritorio\PRACTICA4>






```

9.- Programe un script que elimine todos los archivos de un directorio especificado desde el propio script, y cuyo segundo carácter sea la letra “e”.

```

1  @echo off
2  setlocal enabledelayedexpansion
3
4  set /p directorio=Introduce la ruta del directorio:
5  for %%f in ("%directorio%\*") do (
6      set "filename=%%~nxf"
7      if "!filename:~1,1!" == "e" (
8          del "%%~f"
9      )
10 )
11
12 echo Archivos eliminados correctamente
13 pause
14

```



	escom.txt	15/04/2024 09:11 p. m.	Documento de tex...	0 KB
	felicidad.txt	16/04/2024 08:22 a. m.	Documento de tex...	0 KB
	hola.txt	15/04/2024 09:39 p. m.	Documento de tex...	0 KB
	mesa.txt	16/04/2024 08:22 a. m.	Documento de tex...	0 KB
	pereza.txt	16/04/2024 08:22 a. m.	Documento de tex...	0 KB

```

Microsoft Windows [Versión 10.0.22631.3447]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\jarro\OneDrive\Escritorio\PRACTICA4>Eliminar.bat
Introduce la ruta del directorio:C:\Users\jarro\OneDrive\Escritorio\PRACTICA4\eliminar
Archivos eliminados correctamente
Presione una tecla para continuar . . .

```

	escom.txt	15/04/2024 09:11 p. m.	Documento de tex...	0 KB
	hola.txt	15/04/2024 09:39 p. m.	Documento de tex...	0 KB

CONCLUSION

En conclusión, la programación bajo el intérprete de comandos (shell) en sistemas operativos como Linux y Windows representa una técnica sumamente valiosa para interactuar con el sistema operativo utilizando una interfaz de línea de comandos en lugar de una interfaz gráfica de usuario convencional. Los shells, como Bash en Linux y PowerShell en Windows, ofrecen una amplia gama de herramientas y funcionalidades avanzadas que permiten la programación y la automatización eficientes de tareas del sistema.

Dominar la programación en el intérprete de comandos es una habilidad esencial para administradores de sistemas, desarrolladores de software y cualquier persona que busque mejorar su eficiencia en la realización de tareas del sistema. Los scripts de shell son altamente portátiles entre diferentes sistemas operativos, lo que significa que pueden crearse una vez y ejecutarse en múltiples plataformas, lo que facilita la gestión de sistemas heterogéneos.

En definitiva, la programación en el intérprete de comandos es una herramienta poderosa y flexible que puede mejorar significativamente la productividad y la eficiencia en el trabajo con sistemas operativos Linux y Windows. Además de simplificar tareas cotidianas, la automatización a través de scripts de shell permite a los usuarios concentrarse en tareas más complejas y estratégicas, optimizando así los procesos y recursos del sistema de manera efectiva.