

## **Practica 4**

# **Instituto Politecnico Nacional Escuela Superior de Computo**

## **Sistemas operativos**

## **Programacion bajo el interprete de comandos (shell)**

Ethan Jezreel Lopez Torres  
Gonzaga Martínez José Alberto  
Sebastian Absalon Cortes

## **Marco teorico**

### **Seccion de linux**

#### **Bash scripting**

Bash (Bourne-Again Shell) es un intérprete de comandos de Unix. Este lee los comandos del shell e interactúa con el sistema operativo para ejecutarlos.

Para entender completamente el scripting shell bash, necesitas conocer dos conceptos: shell y scripting.

Shell es un macroprocesador que utiliza comandos para interactuar con el sistema operativo. Esto significa que puede recuperar, procesar y almacenar información en un ordenador.

Por su parte, el scripting es el proceso de compilar comandos del shell en un nuevo archivo utilizando un editor de texto.

Cuando escribes bash en un editor de texto, estás compilando comandos bash o funciones bash, los cuales son un conjunto de comandos que pueden ser llamados numerosas veces tan solo usando el nombre de la función. El texto se guarda entonces como un archivo de script bash ejecutable con la extensión .sh.

### Comandos de bash

#### date

DATE(1)	User	Commands
DATE(1)		
NAME		
	date	- print or set the system date and time
SYNOPSIS		
	date	[OPTION]... [+FORMAT]
	date	[-u --utc --universal] [MMDDhhmm[[CC]YY][.ss]]
DESCRIPTION		
	Display	the current time in the given FORMAT, or set the system date.
	Mandatory	arguments to long options are mandatory for short options too.

#### tar

TAR(1)	GNU	TAR	Manual
TAR(1)			
NAME			
	tar	- an archiving utility	
SYNOPSIS			
	Traditional usage		
	tar	{A c d r t u x}[GnSkUwOmpsMBiajJzZhPlRvwo] [ARG...]	
NOTE			
	This	manpage	is a short description of GNU tar. For a detailed discussion, including ex-
		amples	and usage recommendations, refer to the GNU Tar Manual available in texinfo format.
	If	the	info reader and the tar documentation are properly installed on your system, the
		command	
		info	tar

should give you access to the complete manual.

You can also view the manual using the info mode in emacs(1), or find it in various for-

mats online at

<http://www.gnu.org/software/tar/manual>

If any discrepancies occur between this manpage and the GNU Tar Manual, the later shall be

considered the authoritative source.

#### DESCRIPTION

GNU tar is an archiving program designed to store multiple files in a single file (an ar-

chive), and to manipulate such archives. The archive can be either a regular file or a

device (e.g. a tape drive, hence the name of the program, which stands for tape archiver),

which can be located either on the local or on a remote machine.

## bzip2

bzip2(1)

General Commands Manual

bzip2(1)

#### NAME

bzip2, bunzip2 - a block-sorting file compressor, v1.0.8

bzcat - decompresses files to stdout

bzip2recover - recovers data from damaged bzip2 files

#### SYNOPSIS

bzip2 [ -cdfkqstvvzVL123456789 ] [ filenames ... ]

bzip2 [ -h|--help ]

bunzip2 [ -fkvsVL ] [ filenames ... ]

bunzip2 [ -h|--help ]

bzcat [ -s ] [ filenames ... ]

bzcat [ -h|--help ]

bzip2recover filename

#### DESCRIPTION

bzip2 compresses files using the Burrows-Wheeler block sorting text compression algorithm,

and Huffman coding. Compression is generally considerably better than that achieved by

more conventional LZ77/LZ78-based compressors, and approaches the performance of the PPM

family of statistical compressors.

The command-line options are deliberately very similar to those of GNU gzip,

but they are  
not identical.

bzip2 expects a list of file names to accompany the command-line flags. Each file is replaced by a compressed version of itself, with the name "original\_name.bz2". Each compressed file has the same modification date, permissions, and, when possible, ownership as the corresponding original, so that these properties can be correctly restored at decompression time. File name handling is naive in the sense that there is no mechanism for preserving original file names, permissions, ownerships or dates in filesystems which lack these concepts, or have serious file name length restrictions, such as MS-DOS.

## bunzip2

bzip2(1)  
bzip2(1)

General Commands Manual

### NAME

bzip2, bunzip2 - a block-sorting file compressor, v1.0.8  
bzipcat - decompresses files to stdout  
bzip2recover - recovers data from damaged bzip2 files

### SYNOPSIS

bzip2 [ -cdfkqstvvzVL123456789 ] [ filenames ... ]  
bzip2 [ -h|--help ]  
bunzip2 [ -fkvsVL ] [ filenames ... ]  
bunzip2 [ -h|--help ]  
bzipcat [ -s ] [ filenames ... ]  
bzipcat [ -h|--help ]  
bzip2recover filename

### DESCRIPTION

bzip2 compresses files using the Burrows-Wheeler block sorting text compression algorithm, and Huffman coding. Compression is generally considerably better than that achieved by more conventional LZ77/LZ78-based compressors, and approaches the performance of the PPM family of statistical compressors.

The command-line options are deliberately very similar to those of GNU gzip, but they are not identical.

bzip2 expects a list of file names to accompany the command-line flags.

Each file is re-placed by a compressed version of itself, with the name "original\_name.bz2". Each compressed file has the same modification date, permissions, and, when possible, ownership as the corresponding original, so that these properties can be correctly restored at decompression time. File name handling is naive in the sense that there is no mechanism for preserving original file names, permissions, ownerships or dates in filesystems which lack these concepts, or have serious file name length restrictions, such as MS-DOS.

## Programa 1

```
#!/bin/sh
#
clear
#
ls -l
#
rm -f ./prueba
#
ls -l
#
echo "----- Creando archivo: prueba -----"
#
echo "Hola a todos" >> prueba
echo "Esto es una prueba" >> prueba
echo "De un script" >> prueba
#
ls -l ./prueba
#
echo "----- Script terminado -----"
more ./prueba
```

### Detalles de código

- Este código inicia especificando el shell que ejecutará nuestro código, que en este caso se utiliza "sh"
- A continuación el código ejecuta el comando clear, el cual permite hacer una limpieza de la consola sobre la que se está trabajando.
- A continuación se ejecuta el comando `ls - l`

**A continuación tenemos la ejecución del programa.**

```
total 36
-rw-rw-r-- 1 suzu suzu 710 nov 7 08:08 practica4.md
-rw-rw-r-- 1 suzu suzu 21021 nov 7 07:47 Practica4.pdf
-rw-rw-r-- 1 suzu suzu 347 nov 7 08:11 prog1.sh
-rw-rw-r-- 1 suzu suzu 45 nov 7 08:11 prueba
total 32
-rw-rw-r-- 1 suzu suzu 710 nov 7 08:08 practica4.md
-rw-rw-r-- 1 suzu suzu 21021 nov 7 07:47 Practica4.pdf
-rw-rw-r-- 1 suzu suzu 347 nov 7 08:11 prog1.sh
----- Creando archivo: prueba -----
-rw-rw-r-- 1 suzu suzu 45 nov 7 08:11 ./prueba
----- Script terminado -----
Hola a todos
Esto es una prueba
De un script
suzu@cozzy:~/Documents/operatingSystems/practicas/practica4$
```

## Programa 2

```
#!/bin/sh
#
clear
#
echo "----- Obteniendo fecha -----"
fecha=`date +%Y%m%d%H%M`
#
echo "----- Empacando -----"
tar -cvf ./respaldo$fecha.tar /home/suzu/Documents/AVL
/home/suzu/Documents/networks /home/suzu/Documents/scripts
#
echo "----- Comprimiendo -----"
bzip2 ./respaldo$fecha.tar
#
echo "----- Enviando a otro directorio -----"
cp ./respaldo$fecha.tar.bz2 /home/suzu/respaldos
#
echo "----- Descomprimiendo -----"
bunzip2 /home/suzu/respaldos/respaldo$fecha.tar.bz2
#
echo "----- Desempacando -----"
tar -xvf /home/suzu/respaldos/respaldo$fecha.tar
#
echo "----- Limpiando -----"
rm -f ./respaldo$fecha.tar.bz2
rm -f ./respaldo$fechar.tar
#
echo "----- Terminado -----"
```

A continuacion podemos ver la ejecucion del programa anterior.

```

/home/suzu/Documents/networks/ClienteDNS.o
/home/suzu/Documents/scripts/
/home/suzu/Documents/scripts/goog-mail.py
/home/suzu/Documents/scripts/complex.cs
----- Comprimiendo -----
----- Enviando a otro directorio -----
----- Descomprimiendo -----
bunzip2: Can't open input file /home/suzu/respaldos/respaldo202311111029.tar.bz2
: Not a directory.
----- Desempacando -----
tar: /home/suzu/respaldos/respaldo202311111029.tar: Cannot open: Not a directory
tar: Error is not recoverable: exiting now
----- Limpiando -----
----- Terminado -----
suzu@cozzy:~/Documents/operatingSystems/practicas/practica4$

```

### Programa 3

```

#!/bin/sh
clear
echo -n "Introduce dos valores nmericos separados por espacios"
read var1 var2
echo "Los valores son: "
echo var1=$var1
echo var2=$var2
#
if [ $var1 -lt $var2 ]
then
echo "$var1 menor que $var2"
while [ $var1 -lt $var2 ]
do
echo contador=$var1
var1=`expr $var1 + 1`
done
else
echo "$var1 mayor o igual que $var2"
case $var2 in
10) echo "Fui un 10";;
20) echo "Fui un 20";;
Otro) echo "Otro valor";;
esac
fi

```

A continuacion vemos la ejecucion del rpograma.

```

-n "Introduce dos valores númericos separados por espacios"
142 157
"Los valores son: "
var1=142
var2=157
"142 menor que 157"
contador=142
contador=143
contador=144
contador=145
contador=146
contador=147
contador=148
contador=149
contador=150
contador=151
contador=152
contador=153
contador=154
contador=155
contador=156
suzu@cozzy:~/Documents/operatingSystems/practicas/practica4$

```

## Programa 4

```

#!/bin/sh
clear
Op="Opcion: "
Select i in Directorio DirectorioExtendido Salir
do case $i in
    Directorio) ls;;
    DirectorioExtendido) ls -l;;
    Salir) break;;
    esac
done
#
for i in hola como estan todos
do
    echo -n $i
done
@echo off
rem
cls
rem
dir .\prueba
rem
del .\prueba
rem
dir .\prueba
rem
echo "----- Creando archivo: prueba -----"
rem
echo "Hola a todos" >> prueba
echo "Esto es una prueba" >> prueba
echo "De un batch" >> prueba
rem

```



```
dir .\prueba
rem
echo "----- Script terminado -----"
type .\prueba
```

## Programa 5

```
#!/bin/sh
#
clear
#
echo "===== Descomprimiendo archivos ====="
ls /home/suzu/respaldos

echo "===== Comprimiendo archivos ====="
tar -cvf ./respaldo.tar /home/suzu/Documents/AVL /home/suzu/Documents/networks
/home/suzu/Documents/scripts

echo "===== Comprimiendo archivo ====="
bzip2 ./respaldo.tar

echo "===== Copiando a la carpeta de respaldos ====="
cp ./respaldo.tar.bz2 /home/suzu/respaldos

echo "===== Descomprimiendo archivos ====="
bunzip2 /home/suzu/respaldos/respaldo.tar.bz2

echo "===== Desempaquetando archivos ====="
tar -xvf /home/suzu/respaldos/respaldo.tar
```

```
/home/suzu/Documents/scripts/goog-mail.py
/home/suzu/Documents/scripts/complex.cs
===== Comprimiendo archivo =====
bzip2: Output file ./respaldo.tar.bz2 already exists.
===== Copiando a la carpeta de respaldos =====
===== Descomprimiendo archivos =====
===== Desempaquetando archivos =====
/home/suzu/Documents/AVL/
```

```
===== Descomprimiendo archivos =====
===== Comprimiendo archivos =====
tar: Removing leading `/' from member names
/home/suzu/Documents/AVL/
/home/suzu/Documents/AVL/exe_media_bigplay.svg
tar: Removing leading `/' from hard link targets
/home/suzu/Documents/AVL/icon_present.gif
/home/suzu/Documents/AVL/AnimatedCircle.js
```

## programa 6

A continuacion tenemos un programa que aprovecha muchas de las funciones de bash para poder crear un programa capaz de crear otros programas, en este caso se aprovecha la fucion `echo` que escribe en el archivo out del sistema, pero en este caso direccionando su salida hacia un archivo, que

en este caso es nombrado script\$fecha.sh, el cual esta membretado con la fecha actual de creacion. Finalmente con la funcion de chmod le otorgamos permiso de ejecucion al programa.

```
#!/bin/sh
#
clear

fecha=`date +%Y%m%d%H%M`

echo "===== Creando directorio scripts ====="

mkdir scripts

echo "===== Escribiendo programa ====="

echo "#!/bin/sh
#
clear

fecha=`date +%Y%m%d%H%M`

echo "Escribiendo programa"

mkdir scripts

cd scripts

vim echo > script\$fecha.sh

chmod +x script\$fecha.sh

cd ..

ls /scripts" > ./scripts/script\$fecha.sh

echo "===== Cambiando permisos ====="

chmod +x ./scripts/script\$fecha.sh

ls ./scripts
```

A continuacion podemos ver la ejecucion de este programa.

```
===== Creando directorio scripts =====
===== Escribiendo programa =====
===== Cambiando permisos =====
'script$fecha.sh'
suzu@cozzy:~/Documents/operatingSystems/practicas/practica4$ _
```

## Programa 7

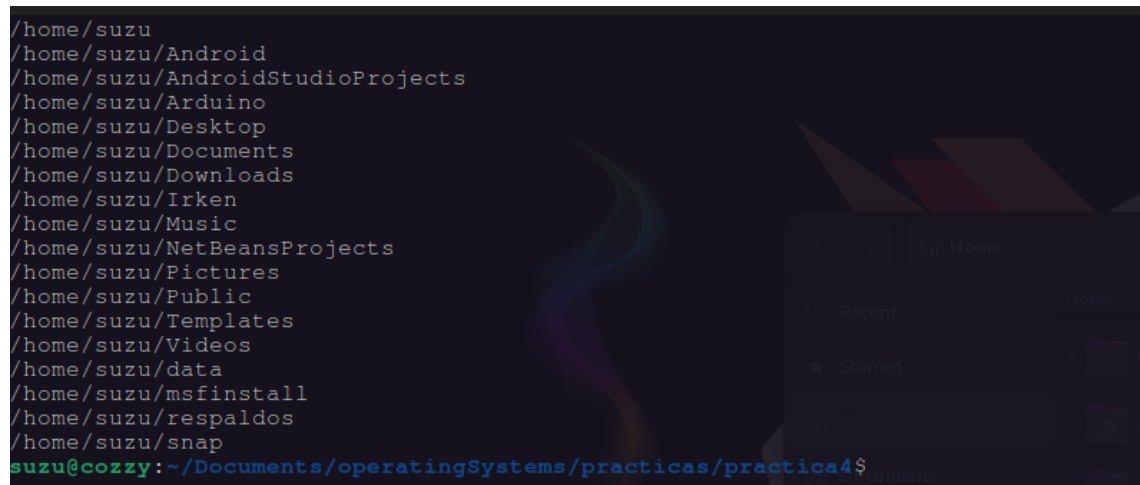
A continuacion tenemos un programa que nos permite leer ver una lista de todos los archivos y carpetas presentes en un directorio seleccionado pro el usuario.

```
#!/bin/sh
#
clear

read search_dir

for entry in "$search_dir"/*
do
    echo "$entry"
done
```

A continuacion tenemos la ejecucion del programa.



```
/home/suzu
/home/suzu/Android
/home/suzu/AndroidStudioProjects
/home/suzu/Arduino
/home/suzu/Desktop
/home/suzu/Documents
/home/suzu/Downloads
/home/suzu/Irken
/home/suzu/Music
/home/suzu/NetBeansProjects
/home/suzu/Pictures
/home/suzu/Public
/home/suzu/Templates
/home/suzu/Videos
/home/suzu/data
/home/suzu/msfinstall
/home/suzu/respaldos
/home/suzu/snap
suzu@cozzy:~/Documents/operatingSystems/practicas/practica4$
```

## Programa 7

A continuaicon tenemos el codigo de un programa que elimina de un directorio seleccionado por el usuairo todos los archivos que cumplan con un patron definido.

```
#!/bin/sh
#
clear

read search_dir

for entry in "$search_dir"/*
do
    filter = ${entry:0:1}
    echo $filter
    if [[ $filter = "e" ]]
    then
        rm -r $entry
    fi
done
```

```
fi
done
```

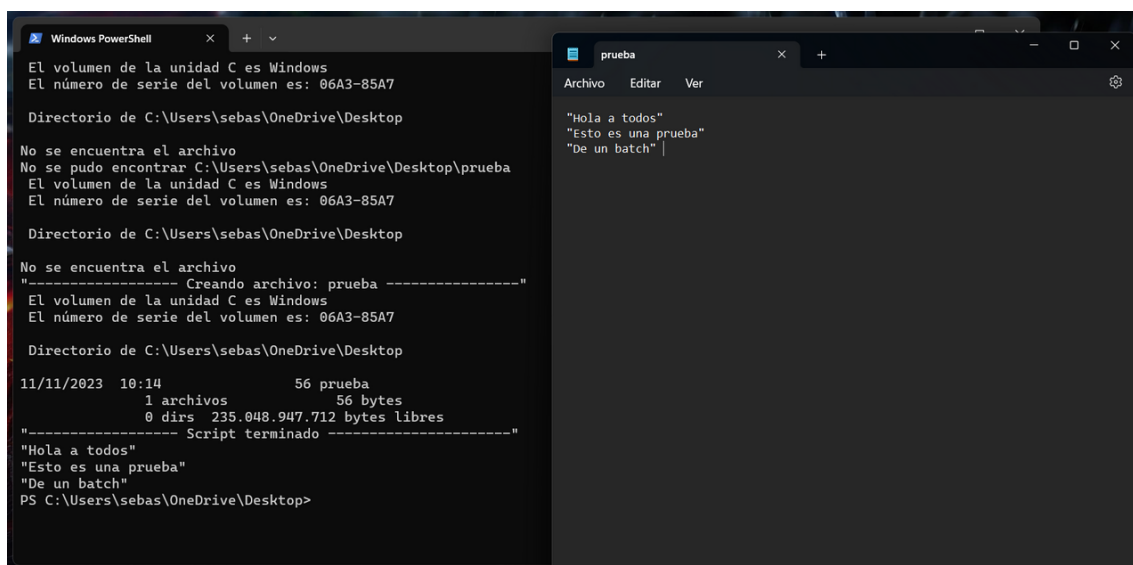
## Seccion de Windows

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

Comandos	Características
Echo	Muestra mensajes o activa o desactiva la característica de eco de comandos. Si se usa sin parámetros, echo muestra la configuración de eco actual.
Del	Elimina uno o varios archivos. Este comando realiza las mismas acciones que el comando erase. El comando del también se puede ejecutar desde la Consola de recuperación de Windows mediante parámetros diferentes.
Date	Muestra o establece la fecha del sistema. Si se usa sin parámetros, date muestra la configuración de fecha actual del sistema y le pide que introduzca una nueva fecha.
Tar	El comando tar archiva y recupera archivos en y a partir de un solo archivo denominado tarfile. Un tarfile suele ser una cinta, pero puede tratarse de un archivo cualquiera.
Rar	RAR es un formato de compresión propietario, perteneciente a la empresa RarLabs. Esto sirve para comprimir y descomprimir archivos en formato *.rar

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

```
@echo off
rem
cls
rem
dir .\prueba
rem
del .\prueba
rem
dir .\prueba
rem
echo "----- Creando archivo: prueba -----"
rem
echo "Hola a todos" >> prueba
echo "Esto es una prueba" >> prueba
echo "De un batch" >> prueba
rem
dir .\prueba
rem
echo "----- Script terminado -----"
type .\prueba
```



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.

```
@echo off
cls
echo "----- Captura fecha -----"
for /F "tokens=1,2,3 delims=/ " %%V in ('date /t') do set dia=%%V%%W%%X
set tiempo=%time%
for /F "tokens=1,2,3 delims=:." %%V in ("%tiempo%") do set tiempo=%%V%%W%%X
set fecha=%dia%%tiempo%
echo "----- Empacando -----"
tar -cvf .\respaldo%fecha%.tar .\directorio1 .\directorio2 .\directorio3
echo "----- Comprimiendo -----"
rar a -c- .\respaldo%fecha%.rar .\respaldo%fecha%.tar
echo "----- Enviando a otro directorio -----"
copy .\respaldo%fecha%.rar .\respaldos\
echo "----- Descomprimiendo -----"
rar e -y .\respaldos\respaldo%fecha%.rar .\respaldos
echo "----- Desempacando -----"
tar -xvf .\respaldos\respaldo%fecha%.tar -C respaldos\
echo "----- Limpiando -----"
del .\respaldo%fecha%.rar
del .\respaldo%fecha%.tar
echo "----- Terminado -----"
```

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

```
@echo off
set /p Var1="Introduce valor del primer numero: "
set /p Var2="Introduce valor del segundo numero: "
if %Var1% LSS %Var2% echo %Var1% es menor que %Var2% & goto :seguir_if
```

```

echo %Var1% es mayor o igual que %Var2%
if %Var1% EQU 10 echo Fui un 10 & goto :salir
if %Var1% EQU 20 echo Fui un 20 & goto :salir
echo Fui otro valor
goto :salir
:seguir_if
for /L %%V IN (%Var1%,1,%Var2%) do (call :imprime_valor %%V)
goto :salir
:imprime_valor
echo contador=%1
goto :eof
:salir
pause

```

5. -Capture y ejecute el siguiente

```

@echo off
echo Opciones:
echo Directorio
echo DirectorioExtendido
echo Salir
set /p Op="Opcion: "
for %%V IN (Directorio DirectorioExtendido Salir) do (call :opcion_elegida %%V)
goto :salir

:opcion_elegida
if %Op%==Directorio dir & goto :salir
if %Op%==DirectorioExtendido echo "Similar a dir" & goto :salir
if %Op%==Salir goto :salir
echo "Opcion no encontrada"
goto :salir
:salir
for %%V in (Hola como estan todos) do echo %%V
pause
exit(0)

```

```
Símbolo del sistema - ejempl...
C:\Users\alber\Desktop>ejemplo.bat
Opciones:
Directorio
DirectorioExtendido
Salir
Opcion: Directorio
El volumen de la unidad C es OS
El número de serie del volumen es: 42F9-D8C0

Directorio de C:\Users\alber\Desktop

06/11/2023  10:45 a. m.  <DIR>      .
03/11/2023  07:10 p. m.  <DIR>      ..
07/11/2023  08:26 a. m.           468 ejemplo.bat
03/11/2023  10:25 a. m.       1,417 Grammarly.lnk
03/11/2023  07:29 p. m.       219 Left 4 Dead 2.url
17/10/2023  07:45 a. m.       1,546 nasm.lnk
31/07/2023  07:46 p. m.       2,233 Notion.lnk
31/10/2023  06:38 p. m.  <DIR>      ORDENAR CARPETAS Y ARCHIVOS
06/11/2023  10:45 a. m.           56 prueba
14/08/2023  06:09 p. m.       222 Wallpaper Engine.url
10/10/2022  03:34 p. m.       1,933 Zoom.lnk
            8 archivos      8,094 bytes
            3 dirs  42,148,716,544 bytes libres

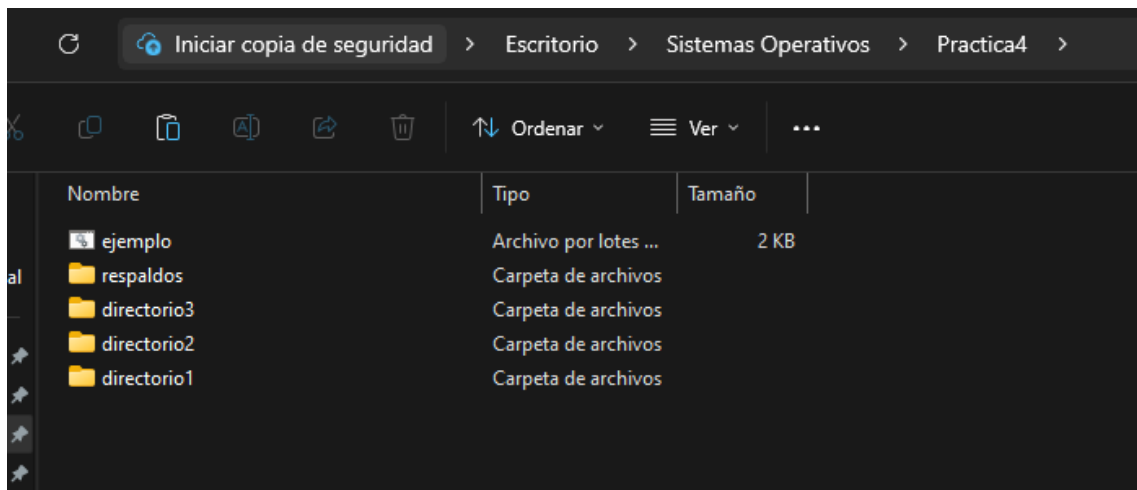
Hola
como
están
todos
Presione una tecla para continuar . . .
```

```
Símbolo del sistema - ejempl...
C:\Users\alber\Desktop>ejemplo.bat
Opciones:
Directorio
DirectorioExtendido
Salir
Opcion: DirectorioExtendido
"Similar a dir"
Hola
como
están
todos
Presione una tecla para continuar . . .
```

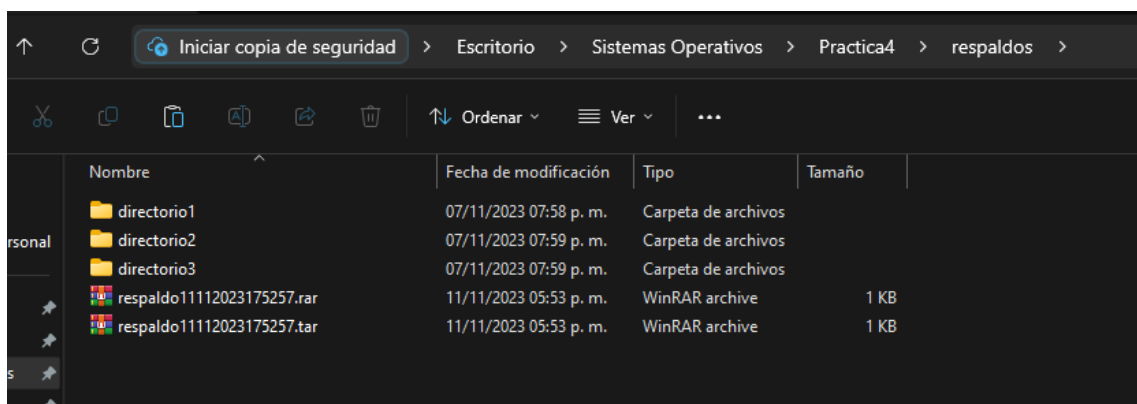
6.- Programe un script para que descomprima en sus rutas originales los directorios que se han compreso con el anterior script.

```
@echo off
cls
set rar_path=C:\Program Files\WinRAR\WinRAR.exe
echo "----- Captura fecha -----"
for /F "tokens=1,2,3 delims=/ " %%V in ('date /t') do set dia=%%V%%W%%X
set tiempo=%time%
for /F "tokens=1,2,3 delims=:." %%V in ("%tiempo%") do set tiempo=%%V%%W%%X
set fecha=%dia%%tiempo%
echo "----- Empacando -----"
tar -cvf .\respaldo%fecha%.rar .\directorio1 .\directorio2 .\directorio3
echo "----- Comprimiendo -----"
"%rar_path%" a -c- .\respaldo%fecha%.rar .\respaldo%fecha%.tar
echo "----- Enviando a otro directorio -----"
copy .\respaldo%fecha%.rar .\respaldos\
echo "----- Descomprimiendo -----"
"%rar_path%" e -y .\respaldos\respaldo%fecha%.rar .\respaldos
echo "----- Desempacando -----"
tar -xvf .\respaldos\respaldo%fecha%.tar -C respaldos\
echo "----- Limpiando -----"
del .\respaldo%fecha%.rar
```

```
del .\respaldo%fecha%.tar
echo "----- Terminado -----"
```



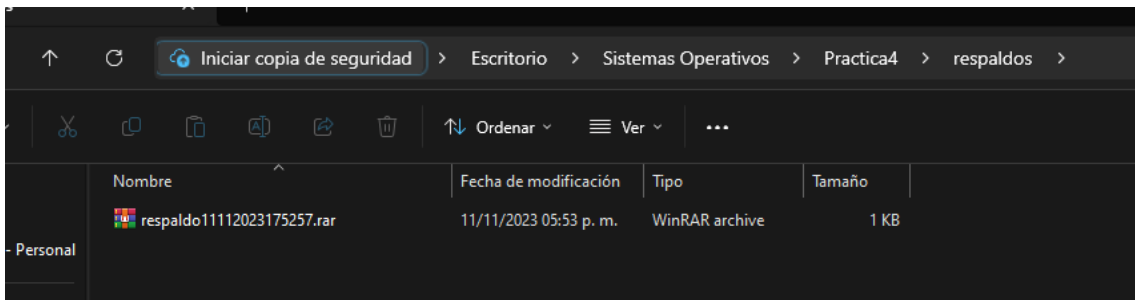
```
----- Captura Fecha -----
----- Empacando -----
a ./directorio1
a ./directorio2
a ./directorio3
----- Comprimiendo -----
----- Enviando a otro directorio -----
1 archivo(s) copiado(s).
----- Descomprimiendo -----
----- Desempacando -----
x ./directorio1/
x ./directorio2/
x ./directorio3/
----- Limpiando -----
----- Terminado -----
```



```
@echo off
cls
set rar_path=C:\Program Files\WinRAR\WinRAR.exe
echo "----- Enviando al directorio original -----"
copy .\respaldos\respaldo.rar .\
echo "----- Descomprimiendo -----"
```



```
"%rar_path%" e -y .\respaldo.tar -C .\
echo "----- Limpiando -----"
del .\respaldo.rar
del .\respaldo.tar
echo "----- Terminado -----"
```

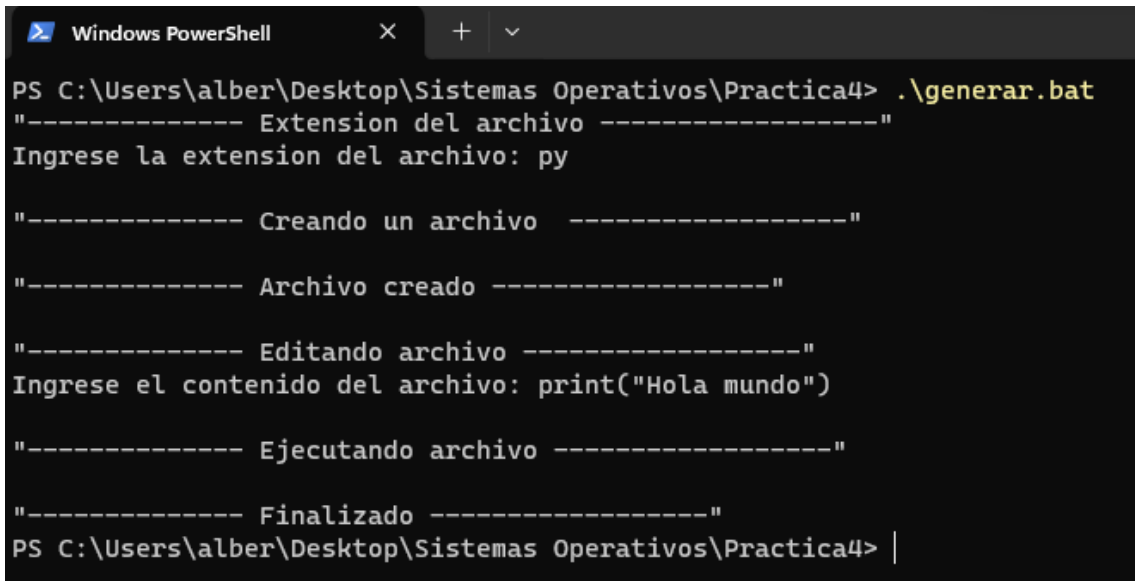


```
"----- Enviando al directorio original -----"
1 archivo(s) copiado(s).
"----- Descomprimiendo -----"
"----- Desempacando -----"
x ./directorio1/
x ./directorio2/
x ./directorio3/
"----- Limpiando -----"
"----- Terminado -----"
```

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

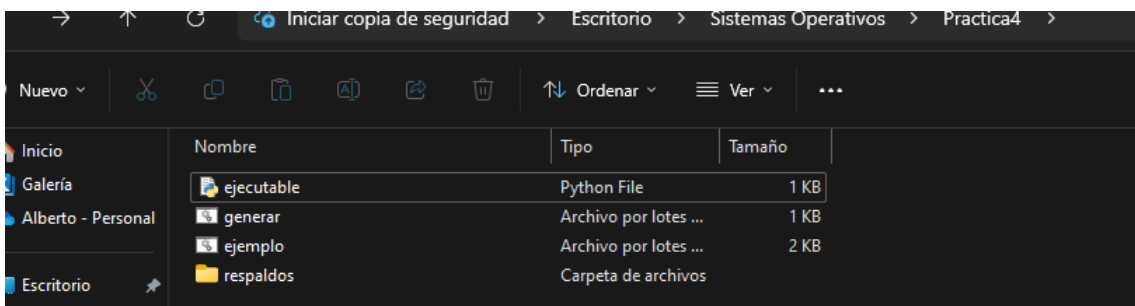
```
@echo off
echo "----- Extension del archivo -----"
set/p extension = "Ingrese la extension del archivo: "
set mensaje = Creando un archivo %extension%
echo.
timeout /t 1 /nobreak > NUL
echo "----- %mensaje% -----"
echo .>ejecutable.%extension%
echo.
timeout /t 1 /nobreak > NUL
echo "----- Archivo creado -----"
echo.
timeout /t 1 /nobreak >NUL
echo "----- Editando archivo -----"
set /p edicion= "Ingrese el contenido del archivo: "
echo %edicion%>ejecutable.%extension%
echo.
timeout /t 1 /nobreak >NUL
echo "----- Ejecutando archivo -----"
start ejecutable .%extension%
```

```
echo.  
echo "----- Finalizado -----"
```



A screenshot of a Windows PowerShell terminal window. The title bar shows 'Windows PowerShell' with standard window controls. The command prompt shows the user is in the directory 'C:\Users\alber\Desktop\Sistemas Operativos\Practica4'. The user has run a batch file named 'generar.bat'. The batch file's output is displayed in the terminal, showing a series of prompts and messages: 'Extension del archivo', 'Ingrese la extension del archivo: py', 'Creando un archivo', 'Archivo creado', 'Editando archivo', 'Ingrese el contenido del archivo: print("Hola mundo")', 'Ejecutando archivo', and 'Finalizado'. The terminal ends with the prompt 'PS C:\Users\alber\Desktop\Sistemas Operativos\Practica4>'.

```
PS C:\Users\alber\Desktop\Sistemas Operativos\Practica4> .\generar.bat  
"----- Extension del archivo -----"  
Ingrese la extension del archivo: py  
  
"----- Creando un archivo -----"  
  
"----- Archivo creado -----"  
  
"----- Editando archivo -----"  
Ingrese el contenido del archivo: print("Hola mundo")  
  
"----- Ejecutando archivo -----"  
  
"----- Finalizado -----"  
PS C:\Users\alber\Desktop\Sistemas Operativos\Practica4> |
```



A screenshot of a Windows File Explorer window showing the contents of the 'Practica4' folder. The address bar shows the path: 'Inicio > Escritorio > Sistemas Operativos > Practica4'. The left sidebar shows the navigation pane with 'Inicio', 'Galería', 'Alberto - Personal', and 'Escritorio'. The main area displays a table of files and folders.

Nombre	Tipo	Tamaño
ejecutable	Python File	1 KB
generar	Archivo por lotes ...	1 KB
ejemplo	Archivo por lotes ...	2 KB
respaldos	Carpeta de archivos	

```
ejecutable.py X
C: > Users > alber > Desktop > Sistemas Operativos > Practica4 > executable.py
1 print("Hola mundo")
2

OUTPUT  DEBUG CONSOLE  PORTS  PROBLEMS  TERMINAL

PS C:\Users\alber> python -u "c:\Users\alber\Desktop\Sistemas Operativos\Practica4\ejecutable.py"
● Hola mundo
○ PS C:\Users\alber>
```

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.

```
@echo off
set /a num_archivo=0
if "%1"==" "(
    echo Debe ingresar un directorio como argumento de entrada.
    goto fin
)
if not exist %1(
    echo %1 no es un directorio válido.
    goto fin
)
echo Archivos en %1:
for %%f in (%1\*) do(
    if %%~xf NEQ "" (
        echo %%~nxf
        set /a num_archivos+=1
    )
)
echo El directorio %1 contiene %num_archivos% archivos.

:fin
```

```

PS C:\Users\alber\Desktop\Sistemas Operativos\Practica4> .\estructuras.bat C:\Users\alber\Desktop
Archivos en C:\Users\alber\Desktop:
6.1.png
6.2.png
6.3.png
6.4.png
6.png
7.png
Grammarly.lnk
Left 4 Dead 2.url
nasm.lnk
Notion.lnk
p5.1_batch.png
p5_batch.png
p6.png
prueba
Unidad 2 Práctica 4 Meza Araujo Jose Emmanuel.pdf
Wallpaper Engine.url
Zoom.lnk
El directorio C:\Users\alber\Desktop contiene 17 archivos.
PS C:\Users\alber\Desktop\Sistemas Operativos\Practica4> |

```

Para su ejecución se requiere con un argumento que será el directorio al que se evaluará (no cuenta carpetas ya que estos cuentan como directorios), con la estructura for vamos imprimiendo el nombre de cada archivo.

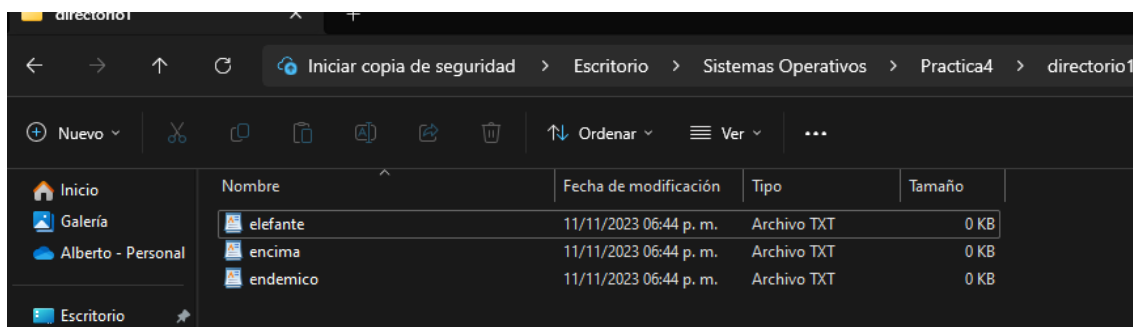
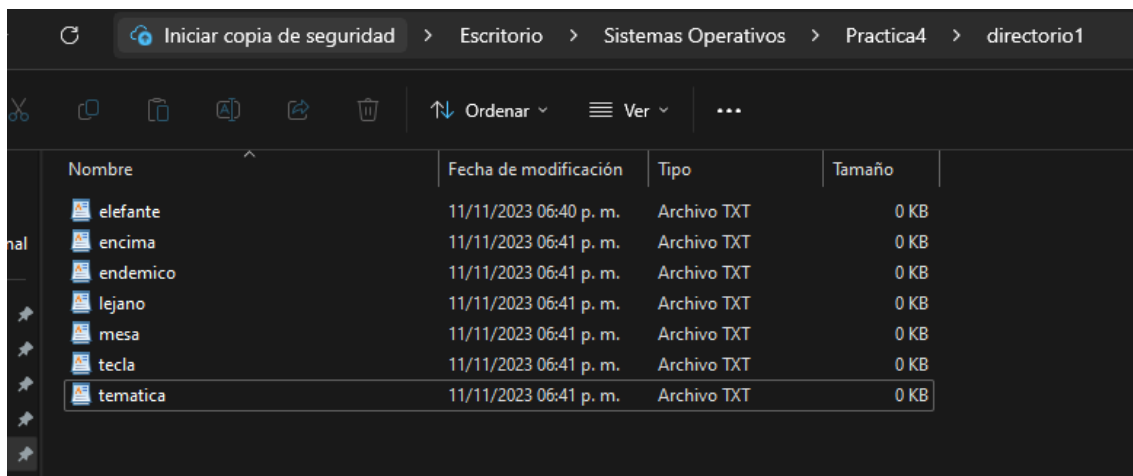
9.- Programe un script que elimine todos los archivos de un directorio especificado desde el propio script, y cuyo segundo carácter sea la letra “e”.

```

@echo off
setlocal enabledelayedexpansion
set /p directorio=Introduce la ruta del directorio:
for %%f in ("%directorio%\") do (
    set "filename=%%~nfx"
    if "!filename:~1,1!" == "e" (
        del "%%~f"
    )
)

echo Archivos eliminados correctamente
pause

```



## Conclusiones

En conclusion, la programación bajo el intérprete de comandos (shell) en sistemas Linux y Windows representa una valiosa técnica para interactuar con el sistema operativo a través de una interfaz de línea de comandos en lugar de una interfaz gráfica. Tanto el Bash en Linux como el PowerShell en Windows ofrecen herramientas avanzadas para la programación y la automatización de tareas del sistema.

Esta habilidad resulta fundamental para administradores de sistemas, desarrolladores de software y cualquier persona que busque optimizar la ejecución de tareas del sistema. Los scripts creados en el entorno de shell son portátiles entre diferentes sistemas operativos, lo que posibilita su escritura una sola vez y su ejecución en diversas plataformas.