**Practica 3- Shell-Scripting.**

**Alumno: Cristian Ortiz**

13. Uso de las estructuras de control:

(a) Realizar un script que visualice por pantalla los números del 1 al 100 así como sus

cuadrados.

#!/bin/bash

#

for variable in {1..100}

do

resultado=$(expr $variable \\* 2)

echo $variable " su cuadrado es: " $resultado

done

echo "Bucle finalizado!"

**(b)** Crear un script que muestre 3 opciones al usuario: Listar, Donde Estoy y Quien Esta.

Según la opción elegida se le debe mostrar:

Listar: lista el contenido del directorio actual.

Donde Estoy: muestra el directorio donde me encuentro ubicado.

Quien Esta: muestra los usuarios conectados al sistema.

#!/bin/bash

################################

# Menú para comandos sencillos

################################

echo –e "\n --- MENU DE COMANDOS---\n"

#

echo " a. Lista del directorio de trabajo"

echo " b. Nombre del directorio de trabajo"

echo " c. Usuarios conectados"

echo

read -p "Introduzca una opción: " opcion

echo

case "$opcion" in

a) ls ;;

b) pwd ;;

c) whoami ;;

\*) echo Opción incorrecta ;;

esac

**(c)** Crear un script que reciba como parámetro el nombre de un archivo e informe si el

mismo existe o no, y en caso afirmativo indique si es un directorio o un archivo. En

caso de que no exista el archivo/directorio cree un directorio con el nombre recibido

como parámetro.

! /bin/bash

if [ -d $1 ];

then

echo “si, es un directorio”

else

if [ -f $1 ];

then

echo “si, es un archivo”

else

mkdir $1

echo “se creo un directorio con el nombre $1”

fi

fi

**14.** Renombrando Archivos: haga un script que renombre solo archivos de un directorio pasado como parametro agregandole una CADENA, contemplando las opciones:

“-a CADENA”: renombra el fichero concatenando CADENA al final del nombre del

archivo

“-b CADENA”: renombra el fichero concantenado CADENA al principio del nombre

del archivo

Ejemplo:

Si tengo los siguientes archivos: /tmp/a /tmp/b

Al ejecutar: ./renombra /tmp/ -a EJ

Obtendré como resultado: /tmp/aEJ /tmp/bEJ

Y si ejecuto: ./renombra /tmp/ -b EJ

El resultado será: /tmp/EJa /tmp/EJb

#! /bin/bash

if [ $# -ne 3 ] ;

then

echo "cantidad de parametros incorrecta"

exit 1

fi

if [ ! -d $1 ] ;

then

echo "no existe el directorio"

exit 2

fi

if [ -z $3 ];

then

echo "el tercer parametro es nulo"

exit 3

fi

case $2 in

"-a") for i in $( ls $1 );

do

mv $1/$i $1/$i$3

done ;;

"-b") for i in $( ls $1 );

do

mv $1/$i $1/$3$i

done ;;

\*)

echo "tiene que poner -b o -a"

exit 4

esac

15. Comando cut. El comando cut nos permite procesar la líneas de la entrada que reciba

(archivo, entrada estándar, resultado de otro comando, etc) y cortar columnas o campos,

siendo posible indicar cual es el delimitador de las mismas. Investigue los parámetros que

puede recibir este comando y cite ejemplos de uso.

**Nos permite sacar campos o simplificar ficheros mucho más complejos**

**Si hacemos “ cut -d “ “ -f 1 fichero.txt ”**

**Lo que hacemos es sacar el primer campo delimitado por un espacio**

**Si hacemos “ cut -d “ “ -f 1,3 fichero.txt “**

**Lo que hacemos es sacar el primer y el tercer campo delimitados por un espacio**

**Si hacemos “ cut -c 10 fichero.txt “**

**Lo que hacemos es sacar los primeros diez caracteres de cada línea del fichero.txt**

**Filtrado de textos**

Para empezar debemos conocer los comandos que nos permiten mostrar el contenido de un fichero. Los comandos más utilizados son: less y more. Por ejemplo, para mostrar el contenido del fichero /etc/passwd podemos ejecutar: less /etc/passwd Para filtrar la salida podemos hacerlo de las siguientes formas:

Mostrar las líneas que cumplen una determinada condición (grep)

Mostrar las n primeras líneas (head) o las n últimas líneas (tail).

**Mostrar una determinada columna (cut)**

Además, podemos ordenar la salida utilizando el comando sort.

A continuación vamos a ver cada uno de los comandos:

grep

El comando grep nos permite filtrar la salida para que se muestren las líneas que cumplen una determinada condición. La sintaxis es:

grep [expresión regular o palabra]

Por ejemplo, si queremos mostrar las lineas del fichero /etc/passwd que contienen la palabra root ejecutaremos:

less /etc/passwd | grep root

head y tail

El comando head muestra las primeras n líneas de un fichero mientras que el comando tail muestra las últimas n últimas líneas del fichero. Ambos comandos tienen la misma sintaxis:

head -n num\_lineas

tail -n num\_lineas

Por ejemplo, si queremos motrar las primeras 5 líneas del fichero /etc/passwd ejecutaremos:

less /etc/passwd | head -n 5

Y para ver las últimas 5 líneas ejecutaremos:

less /etc/passwd | tail -n 5

**cut**

**El comando cut permite obtener de una salida unas determinadas columnas de datos. Su sintaxis es:**

**cut –d “delimitador” -f filas**

**donde delimitador es el carácter que separa los datos entre filas y filas son los números de filas que queremos mostrar. Podemos mostrar una fila (p.e. ¬-f2) o varias filas (p.e. -f2,3). Por ejemplo, para obtener el nombre de todos los usuarios del sistema deberemos obtener la primera fila del fichero /etc/passwd cuyo delimitador es el carácter “:”. Por lo tanto, ejecutaremos:**

**less /etc/passwd | cut -d ":" –f1**

sort

El comando sort permite ordenar una salida de datos. Por ejemplo, siguiente el ejemplo anterior, podemos ordenar los nombres de usuario del equipo.

less /etc/passwd | cut -d ":" –f1 | sort

El caso de estar ordenando valores númericos deremos poner el operador –n. Por ejemplo, si queremos ordenar los identificadors de los usuarios del sistema deberemos ejecutar:

less /etc/passwd | cut -d ":" –f3 | sort -n