PRACTICA 5

5.1

Creamos las variables:

dias=365

dia=`date +%j`

echo "Faltan $(( ($dias - $dia) / 7 )) semanas hasta el fin de año"

echo “Faltan $[ ($dias - $dia) /7 ] semanas hasta el fin de año”

echo "Faltan `expr \( $dias - $dia \) / 7` semanas hasta finde año"

5.3:

jose@Jose-Ubuntu:~$ x=18

jose@Jose-Ubuntu:~$ echo $((x/=2))

9

jose@Jose-Ubuntu:~$ x=18

jose@Jose-Ubuntu:~$ echo x=$((x/2))

x=9

5.4:

jose@Jose-Ubuntu:~$ echo 6/5 | bc -l

1.20000000000000000000

jose@Jose-Ubuntu:~$ echo "6/5 | bc -l"

6/5 | bc -l

jose@Jose-Ubuntu:~$ echo '6/5 | bc -l'

6/5 | bc -l

jose@Jose-Ubuntu:~$ echo "6/5" | bc -l

1.20000000000000000000

jose@Jose-Ubuntu:~$ echo '6/5' | bc -l

1.20000000000000000000

5.5:

echo \(3-2\)/5 | bc -l

Si le ponemos comillas dobles ya nos cogeria el “texto” bien y por lo tanto no nos haria

falta escapar los parentesis, asique quedaria asi: echo "(3-2)/5" | bc -l y en el caso de las comillas

simples se quedaria igual que con las comillas dobles: echo '(3-2)/5' | bc -l

En el caso de poner las comillas tumbadas no se realizaria ya que no tenemos ninguna orden dentro

de las comillas tumbadas.

5.7:

jose@Jose-Ubuntu:~$ let w=4+5\*6 j=(4+5)\*6

jose@Jose-Ubuntu:~$ echo $w

34

jose@Jose-Ubuntu:~$ echo $j

54

5.8:

#!/bin/bash

if [ $# -eq 2 ]

then

echo "TENED EN CUENTA QUE 0 ES VERDADERO Y 1 ES FALSO."

[ $1 -gt $2 ]; echo "Si comparamos $1 mayor que $2 obtenemos $?"

[ $1 -lt $2 ]; echo "Si comparamos $1 menor que $2 obtenemos $?"

[ $1 -eq $2 ]; echo "Si comparamos $1 igual que $2 obtenemos $?"

fi

5.9:

#!/bin/bash

if [ $# -eq 1 ]; then

plano\_ejecucion=`test -x $1 && test -f $1 ; echo $?`

enlace=`test -h $1 ; echo $?`

echo $plano\_ejecucion ". Vamos a obtener 0 si el fichero es plano y tiene permisos de ejecucion."

echo $enlace ". Vamos a obtener 0 si el fichero es un enlace simbolico."

fi

5.10

test “hola” < “pepito el de los palotes” ;; → Nos va a devolver 0 porque se cumple

test 9 -lt 2 ;; → Nos va a devolver 1 porque no se cumple

5.11:

a) Primero comprueba que sesion5.pdf es archivo existente y es plano o regular, si eso se cumple, imprime por pantalla un mensaje que muestra que el archivo existe.

b) if test -f ./sesion5.pdf ; then

printf “El archivo ./sesion5.pdf existe\n”

elif test -d /bin ; then

pintf “El archivo ./sesion5.pdf no existe\n”

pintf “El directorio /bin si existe.\n”

else

printf “El directorio /bin no existe.\n”

fi

c)

#!/bin/bash

if test -f $1 ; then

printf "El archivo $1 existe\n"

elif test -d $2 ; then

printf "El archivo $1 no existe\n"

printf "El directorio $2 si existe.\n"

else

printf "El directorio $2 no existe.\n"

fi

5.12:

#!/bin/bash

if [ $# -eq 1 ]; then

if test -O $1 ; then

echo "Eres propietario del archivo o directorio."

else

echo "No eres propietario del archivo o directorio."

fi

if test -r $1 ; then

echo "Tienes permiso de lectura sobre el archivo o directorio."

else

echo "No tienes permisos de lectura sobre el archivo o directorio."

fi

fi

5.13:

#!/bin/bash

if [ "$1" = "-h" ]; then

echo "Intrucciones del guion:"

echo "Este guion va a calcular los dias que quedan para fin de año y va a mostrar un mensaje diciendo si el numero de dias es multiplo de 5 o no."

echo "\n"

echo "Si ejecutamos el nombre del fichero con la opcion -h accederemos a la ayuda."

elif [ $(( (365 - `date +%j`) % 5 )) -eq 0 ]; then

echo "El numero de dias que faltan para fin de año es multimplo de 5."

else

echo "El numero de dias que faltan para fin de año no es multimplo de 5."

Fi

5.15:

#!/bin/bash

if [ $# -eq 1 ]; then

if echo $1 | grep -E [a-z] >/dev/null; then

echo "El caracter introducido es una letra minuscula."

elif echo $1 | grep -E [A-Z] >/dev/null; then

echo "El caracter introducido es una letra mayusucula."

else

echo "El caracter introducido no es una unica letra."

fi

fi

5.16:

Como no especifica el orden pongo varias formas:

Al menos un digito y una e sin importar el orden: ls -R | grep -E '[0-9]' | grep -E 'e'

Al menos un digito y una e siendo éste el orden: ls -R | grep -E '[0-9]+.\*e'

Contengan una letra e y no lleven un 0 o un 1: ls -R | grep -E 'e' | grep -Ev '[01]'

5.17:

A un archivo llamado hola haríamos: cat hola | wc –l

Usando grep podemos hacerlo: cat hola | grep -Ec ''