#! /bin/bash

Echo “Hola, Cual es tu usuario”

Read -p ‘usuarios:’ user

Read -sp ‘contraseña’ password # s = silent p=password

Echo “gusto saludarte ” $(user)

echo "ingresa tus notas"

read nota1 nota2 nota3

echo "tu primera nota es "$(nota1)

echo "tu segunda nota es "$(nota2)

echo "tu tercera nota es "$(nota3)

exit 0

Ejercicio de funciones

#!/bin/bash

print\_test(){

        echo "esta es una funcion de prueba"

}

print\_test

echo "hola"

print\_test

exit 0

ejercicio de funciones saludar al parametrol

print\_test(){

        echo "esta es una funcion de prueba"

}

print\_test2(){

echo "hola "$1

}

echo "hola"

print\_test2 'juan'

exit 0

una función retorna un valor

#!/bin/bash

print\_test(){

        echo "esta es una funcion de prueba"

}

print\_test2{

print\_test2(){

echo "hola "$1

return 1

}

echo "hola"

print\_test2 'juan'

echo "la funcion test2 retorna el valor" $?

exit 0

ojo todas las variables por difinicion son globales

-hay maneras de usar variables locales solo existen en la función en la que se definieron

-siempre es recomendado utilizar variables locales

Uso de variables locales

locales(){

        local var1='local 1'

        echo "VALORES DE LAS VARIABLES" ${var1} " " ${var2}

        echo var1="cambio de 1 a 2"

        echo var2="cambio de 2 a 3"

}

var1="global 1"

var2='global 2'

echo ${var1} " " ${var2}

locales

echo ${var1} " " ${var2}

exit 0

función para escanear ip

#!/bin/bash

function\_scanner(){

        read -p "ingrese el rando a escanear" rango

        for i in ${seq 1 254}

        do

                existe=`ping -w 3 -c 1 $rango.$1 | grep 'time='`

                if [ "$existe" = "" ]

                then

                echo $rango.$i " esta disponible"

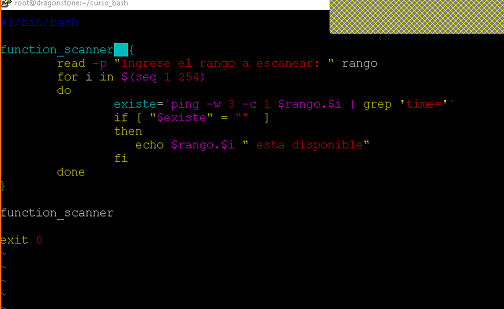
                fi

        done

}

function\_scanner

exit 0



\#!/bin/bash

function\_scanner(){

read -p "ingrese el rando a escanear " rango

for i in $(seq 1 254)

do

existe=`ping -w 3 -c 1 $rango.$i | grep 'time='`

if [ "$existe" = "" ]

then

echo $rango.$i " esta disponible" >>lista.csv

else

echo $rango.$i "no esta disponible" >>lista.csv

fi

done

}

>lista.csv

function\_scanner

exit 0

Ejercicio en clase

 #!/bin/bash

# desplegar todos los archivos con un tamaño mayora w mb, con su nombre, tamaño>

buscador(){

find / type f -size 2M -exec ls -lh {}\; >> lista.csv

}

>lista.csv

buscador

exit 0

#! /bin/bash

archivodesalida="lista.csv"

archivodesalida2="lista.txt"

buscador\_de\_archivos(){

$( find / -type f -size +2M -exec du -sh {} +2 > /dev/null > archivodesalida2 )

echo "Archivo,Tamaño,Ubicacion" > $archivodesalida

#

#while IFS= read -r line; do

# size=$(echo $line | awk '{print $1}')

# path=$(echo $line | awk '{print $2}')

# name=$(echo $line | awk -F "/"' {print $NF}')

# echo "${name},${size),${path}" >> $archivodesalida

#done < $archivodesalida2

}

echo " \_\_^\_\_ \_\_^\_\_"

echo "( \_\_\_ )------------------------------( \_\_\_ )"

echo " | / | Busqueda de archivo de 2 megas | \ |"

echo " |\_\_\_| |\_\_\_|"

echo "(\_\_\_\_\_)------------------------------(\_\_\_\_\_)"

echo "Deseas buscar archivos de 2 megas?"

#!/bin/bash

output\_file="output.txt"

output\_csv="output.csv"

$(find/-typef-size+2M -exec du -sh{}+2>/dev/null>$output\_file)

echo"Archivo,Tamaño,Ubicación">$output\_csv

while IFS read -r line;do

size=$(echo$line|awk'{print$1}')

path=$(echo$line|awk'{print$2}')

name=$(echo$line|awk -F"/"'{print$NF}')

echo"${name},${size},${path}">>$output\_csv

done$output\_file

#!/bin/bash

output\_file="output.txt"

output\_csv="output.csv"

$(find / -type f -size +2M -exec du -sh {} + 2> /dev/null > $output\_file)

echo "Archivo,Tamaño,Ubicación" > $output\_csv

while IFS= read -r line; do

size=$(echo $line | awk '{print$1}')

path=$(echo $line | awk '{print$2}')

name=$(echo $line | awk -F "/" '{print$NF}')

echo "${name},${size},${path}" >> $output\_csv

done < $output\_file

Comando Find

Tarea resuelta tipo 1

#!/bin/bash

archivosalida="lista.csv"

echo "nombre,tamaño,path" > $(archivosalida)

temp=$(find / type f -size  -2M

for archivo in $temp

do

  echo $(archivo)

  nombre=$basepath $(archivo)

  tamaño=$(du -sh $(archivo))

  directorio=$(dirname $(archivo))

  echo ${nombre}","${tamaño}","${directorio} >> ${archivosalida}

done

exit 0

Método 2

#!/bin/bash

archivosalida="lista.csv"

echo "nombre,tamaño,path" > $(archivosalida)

temp=$(find / type f -size  -2M)

for archivo in $temp

do

  echo $(basename ${archivo})","$(du -sh ${tamaño})","$(dirname ${directorio}) >

done

exit 0

Ejercicio en bash con el awk 2

#!/bin/bash

awk 'BEGIN {i=1; while (i < 6) {print i; ++i}}'

exit 0

método 2 utilizar c++ en bash o programas de java

#!/bin/bash

awk 'BEGIN {i=1; while (i < 6) {print i; ++i}}'

awk 'BEGIN {sum=0;

        for(i=0;i<20;++i)

        { sum+=i;

          if(sum>50)

                break;

         else

          print "Suma=",sum}}'

exit 0

#!/bin/bash

awk 'BEGIN {i=1; while (i < 6) {print i; ++i}}'

awk 'BEGIN {sum=0;

        for(i=0;i<20;++i)

        { sum+=i;

          if(sum>50)

                break;

         else

          print "Suma=",sum}}'

awk 'BEGIN{

        for(i=1;i<=20;++i)

        { if (i%2==0)

                print i;

         else

            continue}}'

exit 0

numeros primos

#!/bin/bash

awk 'BEGIN {suma=0;

        print "Ingrese el primer numero"; getline n1; print "Ingrese el segundo>

for(i=1;i<=(n1/2);++i){

        if(n1%i==0)

         suma+=i;

}

        if(suma==2)

          suma=0;

        for(i=1;i<n2;++i){

           if(n2%i==0)

                suma=suma+i;

           }

if(suma==n1){

        echo "Son Amigos"

else

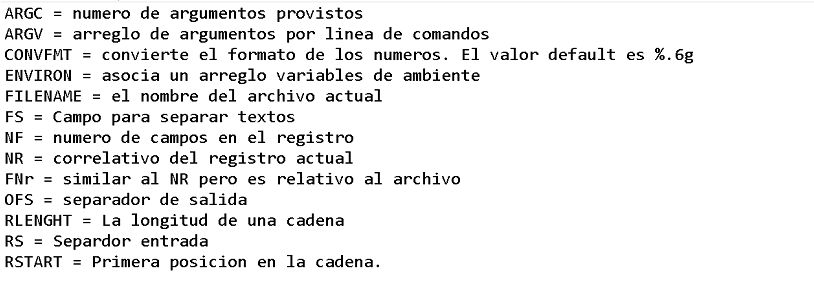
        echo "No son amigos"

else

echo  "no son amigos"

'

parametros awk



+= sumar

-= restar

\*= multiplicar

/= dividir