Universidad de Buenos Aires
Facultad de Ingeniería

Seguridad en Redes
TP – Shell Scripting en Unix/Linux

1. Objetivo

El objetivo de este trabajo práctico es introducir al alumno en la programación de scripts básicos en SO Unix/Linux y brindar los conocimientos para comprender scripts ya realizados, tomando como referencia los elementos del SO estudiados en TPs anteriores.

2. Variables

2.1. Imprima el valor de la variable var1

```
#echo $var1
```

2.2. Asigne un valor numérico a la variable var1, un valor alfanumérico a la variable var2.

```
# var1=1
# var2=”abcdefghijklmnopqrstuvwxyz”
# echo $var1
# echo $var2
```

2.3. - Verifique la longitud de las variables

```
# echo ${#var1}
# echo ${#var2}
# echo `expr  length $var2` (Notar que lo que esta entre “`” se ejecuta primero)
```

2.4. - Extraiga las ultimas 10 letras de la variable var2 (3 maneras distintas). Extraiga las letras entre la c y la f.

```
Opcion 1
# echo ${var2:16}

Opcion 2
# longitud_varaible2=$(echo ${#var2})
# echo ${var2:`expr $longitud_variable2 – 10`}

Opcion 3
# echo ${var2:`expr ${#var2} – 10`}
# echo ${var2:2:4}
```

3. Test – if then else

3.1. Ver la salida de los siguientes commandos

```
# [ -e /etc/passwd ]
# [ -x /etc/passwd ]
# [ -f /etc/hosts ]
# [ -d /etc ]
# [ -L /crypto ]
```

3.2. Escriba un script que verifique la existencia de un archivo (pasado por parámetro) y en caso que exista haga un backup y en caso contrario imprima un error.

```
#!/bin/bash
echo $1
If [ -e $1 ]
then
  cp $1 ${1}.bkp && echo “Archivo resguardado”
else
  echo “el archivo no existe”
fi
```

4. For, while, case, select

4.1. Ejecutar el script que se muestra a continuación.

```
for planet in Mercury Venus Earth Mars Jupiter Saturn Uranus Neptune
do
  echo $planet
done
```

Cómo muestra la salida?

```
for planet in "Mercury Venus Earth Mars Jupiter Saturn Uranus Neptune"
do
  echo $planet
done
```

Cómo muestra la salida ahora? Que diferencias hay con la anterior?

```
for planetoide in $planet
do
  echo $planetoide
done
```

Cómo muestra la salida?

```
for planetoide in "$planet"
do
  echo $planetoide
done
```

Y ahora?

4.2. Ingresar una línea en blanco luego de cada línea del archivo pasado por parámetro previo chequeo de que exista y pueda ser leído.

```
#!/bin/bash
if [ $# -ne 1 ]; then
  echo “USO: $0 archivo”
  exit 2
fi
if [ ¡ -f $1 ] ; then
```

```
  echo “No existe el archivo”
  exit 3
fi
while read linea
do
  echo $linea
  echo
done < $1
```

4.3. Ejecutar el siguiente script y analizar la salida

```
#!/bin/bash
# Se asocia cada planeta con su distancia al centro egocentrico.
for planet in "Mercury 36" "Venus 67" "Earth 93"  "Mars 142" "Jupiter 483"
do
  A=$(echo $planet | cut -d" " -f1)
  B=$(echo $planet | cut -d" " -f2)
  echo "$A tiene  $B de distancia"
done
```

¿Qué hace “A=$(echo $planet | cut -d" " -f1)”?.

Hacer “man cut” y ver qué son los parámetros –d y –f.

¿Por qué los pares “Planeta distancia” van entre ‘ “ ’? Que pasa si se sacan las ‘ “ ‘?

4.4. El siguiente script pide que ingrese algo por teclado hasta que se ingrese “end”

```
#!/bin/bash
while [ "$var1" != "end" ]
do
  echo "Ingrese una linea"
  read var1
  echo "Ingresó: $var1"
  echo
done
echo Termino
```

4.5. El siguiente script pide que ingrese un caracter por teclado y dice que tipo de caracter es. Para salir se debe presionar “F”

```
while [ "$Keypress" != "F" ]
do
  echo; echo "entre una letra seguido de enter (F PARA SALIR)."
  read Keypress
  case "$Keypress" in
    [[:lower:]]   ) echo "Minuscula";;
    [[:upper:]]   ) echo "Mayuscula";;
    [0-9]         ) echo "Numerico";;
    *             ) echo "Error en el ingreso del caracter";;
  esac
done
echo Termino
```

4.6. El siguiente script imprime un menú y permite que se seleccione un item

```
PS3='Seleccione una materia: '
echo
select materia in "crypto" "Circuitos1" "Senales" "Trabajo Profesional"
"Chino"
do
  echo
  echo "La materia seleccionada es $materia."
  echo "mas suerte la proxima!"
  echo
  break
done
```

5. Funciones

5.1. Corra el siguiente script y analice la salida. Este script imprime mensajes usando funciones para tal fin.

```
#!/bin/bash
imprimeCartel ()
{
  echo "Cartel de Inicio del sistema."
  echo "siga los pasos."
}

imprimemensaje ()
{
  if [ $1 == "ERR" ]; then
      echo "ERROR: $@"
  else
      echo "El mensaje es: $@"
  fi
}

imprimeCartel
imprimemensaje "ERR" Mensaje importante
imprimemensaje "OK" Otro Mensaje importante
```

Qué es $1 dentro de la función imprimemensaje()?

Qué es $@ dentro de las función imprimemensaje()?

6. Expresiones Regulares

6.1. Ejecute el siguiente script

```
#!/bin/bash
A="Matematica Discreta es genial"
B="Pipeta es una bomba"
C="Escopeta es un peligro"
echo $A
```

```
echo $B
echo $C
read
echo $A | sed "s/.*ta/Crypto/g"
echo $B | sed "s/.*ta/Crypto/g"
echo $C | sed "s/.*ta/Crypto/g"
read
echo $A | sed "s/M.*ta/Crypto/g"
echo $B | sed "s/M.*ta/Crypto/g"
echo $C | sed "s/M.*ta/Crypto/g"
echo "ingresar una string"
read D
echo $D | sed "s/M.*ta/Crypto/g"
echo $A | sed "s/[MP].*ta/Crypto/g"
echo $B | sed "s/[MP].*ta/Crypto/g"
echo $C | sed "s/[MP].*ta/Crypto/g"
```

6.2. Ejecute el siguiente script

```
#!/bin/bash
A="Matematica Discreta es genial"
B="Pipeta es una bomba"
C="Escopeta es un peligro"
echo $A
echo $B
echo $C
echo “presione ENTER para continuar”
read
echo “Salida 1”
echo $A | sed "s/es//g"
echo “Salida 2”
echo $A | sed "s/es//g" | sed "s/genial/aburrida/g"
echo “Salida 3”
echo $A | sed "s/(M.*ta)/ vino \1/g"
echo “Salida 4”
echo $A | sed "s/\(M.*ta\).*/ vino \1/"
```

Qué sucede en cada salida?

7. Compilación de Código C

7.1. Escribir el programa C

```
#vi pepe.c
int main(void) {
  printf("hello world\n");
  exit(55);
}
```

7.2. Compilar el programa

```
gcc -o programa pepe.c
```

7.3. Correr el programa compilado

```
# ./programa
```

8. Ejercicios de scripting (opcional)

8.1. Ejercicio Nº1

Escribir un script que contenga:

- Una función que permita saber si un archivo existe (el archivo se pasa por parámetro).
- Una función que verifique la existencia de un usuario pasado por parámetro.
- Un menú con las siguientes opciones
    - "Verificar la existencia de un usuario" (al usuario lo lee de la consola) e imprime cual es su directorio HOME en caso que exista.
    - "Verificar la existencia de un archivo". Este punto debe desplegar un menú con el path de al menos 4 archivos
    - Escribir un programa que pida el ingreso de una lista de palabras hasta presionar la letra F. Luego verificar de esas palabras cuales empiezan con minúscula, cuales con mayúsculas y cuales con números.

Link para ayuda: http://www.tldp.org/LDP/abs/html/

8.2. Ejercicio Nº2

Escribir un script que permita efectuar las siguientes acciones

- Indicar la dirección IP de la interfaz de red (ej: eth0)
- Obtener una IP en forma automática mediante DHCP
- Configurar manualmente IP y máscara de red, solicitando estos datos al usuario
- Mostrar el contenido del archivo /etc/hosts
- Mostrar la tabla de ruteo
- Mostrar los puertos TCP/UDP activos
- Efectuar un ping a un destino indicado por el usuario