

UT 04. Lenguajes de Scripts: actividades Bash (II)

1.- Busca información para el manejo de cadenas de caracteres en Bash (pasar a mayúsculas, minúsculas, separar, concatenar, etc.).

Mayúsculas y minúsculas.

```
edu@US-ST:~/scripts$ echo "HOLA" | tr [[:upper:]] [[:lower:]]
hola
edu@US-ST:~/scripts$ echo "hola" | tr [[:lower:]] [[:upper:]]
HOLA
```

Separar cadenas con los comandos cut o tr.

```
edu@US-ST:~/scripts$ cadena="juan;pedro"
edu@US-ST:~/scripts$ echo $cadena | cut -f1 -d";"
```

Juntar cadenas.

```
edu@US-ST:~/scripts$ cadena1="Hola "
edu@US-ST:~/scripts$ cadena2="mundo"
edu@US-ST:~/scripts$ cadena_junta="$cadena1 $cadena2"
edu@US-ST:~/scripts$ echo $cadena_junta
Hola mundo
```

2.- Hacer un script que verifique si el directorio /sbin está en la variable PATH. Debe mostrar en pantalla: "/sbin está en la variable PATH" ó "/sbin no está en la variable PATH".



```
esta_path.sh
1  #!/bin/bash
2
3  directorio="/sbin"
4  esta=0
5
6  for i in $(echo $PATH | tr ":" "\n")
7  do
8      if [ $directorio = $i ]
9      then
10         esta=1
11     fi
12 done
13
14 if [ $esta -eq 1 ]
15 then
16     echo "$directorio está en la variable PATH."
17 else
18     echo "$directorio no está en la variable PATH."
19 fi
```

```
edu@US-ST:~/scripts$ esta_path.sh
/sbin está en la variable PATH.
```

UT 04. Lenguajes de Scripts: actividades Bash (II)

Ahora, generalizar este ejercicio para verificar cualquier directorio introducido en la línea de comandos (uno o varios argumentos).

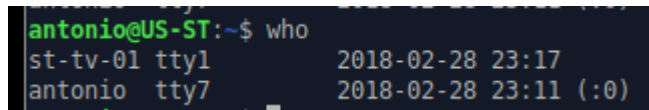
```
edu@US-ST:~$ esta_path.sh /sbin /home
/sbin si está en la variable PATH.
/home no está en la variable PATH.
```

3.- Crear un script que diga qué usuarios, **de los que le pasamos como argumentos en la línea de comandos**, están o no conectados al sistema.

Controlar posibles errores.

Ayuda

El comando **who** muestra los usuarios conectados. Por ejemplo, en la imagen que se muestra a continuación hay dos usuarios conectados.



```
antonio@US-ST:~$ who
st-tv-01 tty1      2018-02-28 23:17
antonio  tty7      2018-02-28 23:11 (:0)
```

Observación

Me conecto con distintos usuarios abriendo consolas: Control + Alt + F1, F2, F3, etc.

```
edu@US-ST:~$ conectados.sh edu pepe alvaro
edu está conectado.
pepe no está conectado.
alvaro no está conectado.
```

4.- Tenemos almacenado en un **array** las **empresas** en las que hacen prácticas los alumnos del IES Gregorio Prieto junto a sus **teléfonos y direcciones**.

Realizar un script que, cuando se ejecute, solicite al usuario el nombre de la empresa y le muestre su teléfono y dirección. Si no se encuentra la empresa, se debe mostrar un mensaje indicándolo. El programa se debe repetir hasta que el usuario decida terminar.

De momento tenemos este array que debes incluir en tu script.

```
fct=(“TECNOBIT” 123132333 “C/POLÍGONO 34” “SAVIA” 212122111 “C/SEIS DE JUNIO 23” “UNED” 111222000 “C/SEIS DE JUNIO 12”)
```

Por ejemplo, si el usuario introduce SAVIA, le mostramos en pantalla lo siguiente:

El teléfono de SAVIA es 212122111 y su dirección es C/SEIS DE JUNIO 23

Por ejemplo, si el usuario introduce FRIMANCHA, le mostramos en pantalla lo siguiente:

Lo sentimos, la empresa FRIMANCHA no la tenemos almacenada.

UT 04. Lenguajes de Scripts: actividades Bash (II)

```
edu@US-ST:~/scripts$ empresas_fct.sh
```

Introduzca el nombre de la empresa: SAVIA

El teléfono de SAVIA es 212122111 y su dirección es C/SEIS_DE_JUNIO_23.

Pulse una tecla para continuar...

Pulse "s" para salir del programa.

Introduzca el nombre de la empresa: FONTANEROS

Lo sentimos, la empresa FONTANEROS no la tenemos almacenada.

Pulse una tecla para continuar...

Pulse "s" para salir del programa.

5.- Nos han pedido que ayudemos a los usuarios a solucionarse posibles problemas que puedan tener en el trabajo diario con su equipo, es decir, queremos poner a su disposición una aplicación a modo de **soporte técnico básico en línea.**

El script, `/usr/local/bin/tunombre_st.sh` (755) debe mostrar en pantalla el siguiente menú:

```
"Problemas con:"  
"1.- INTERNET"  
"2.- DATOS"  
"3.- LENTO"  
"4.- OTROS"  
"5.- SALIR"
```

→ Si el usuario pulsa 1, 2 ó 3, nuestro script le debe ir sugiriendo posibles soluciones (las tenemos almacenadas en `/usr/local/bin/soluciones` - 644 -) a los problemas planteados.

Contenido actual de `/usr/local/bin/soluciones` y que debes copiar así, tal como te lo muestro.

```
1:INTERNET:COMPRUEBA_QUE_ESTÉ_ENCENDIDO/CONECTADO_EL_ADAPTADOR  
1:INTERNET:UTILIZA_OTRO_NAVIGADOR  
2:DATOS:REVISA_SI_TIENES_CAPS_LOCK_ACTIVADO_E_INTRODUCE_DE_NUEVO_USUARIO/CLAVE  
2:DATOS:COMPRUEBA_QUE_ACCEDES_A_LAS_CARPETAS_ADECUADAS  
3:LENTO:ELIMINA_ARCHIVOS_INNECESARIOS  
3:LENTO:DESFRAGMENTA_EL_DISCO
```

Si alguna sugerencia de las que hay en este fichero les ha solucionado el problema, el programa termina y si no, se le dice que se toma nota y **además se añade una línea al fichero `./logs/no_solucionado`** indicando que ninguna de las sugerencias actuales soluciona el problema (el directorio `./logs` es oculto ya que empieza por `.` y debe tener permisos 777).

→ Si el usuario pulsa 4, es decir, OTROS, simplemente le pedimos que introduzca el problema y **lo añadimos al fichero `./logs/nuevos_problemas`**.

→ Si el usuario pulsa 5, terminamos.

→ Cualquier otra opción elegida hará que el programa se repita de nuevo indicándole que la opción es incorrecta.

UT 04. Lenguajes de Scripts: actividades Bash (II)

Vamos a ver el funcionamiento del programa para que lo entiendas mejor.

a) Por ejemplo, pulsamos 1

```
Problemas con:
1.- INTERNET
2.- DATOS
3.- LENTO
4.- OTROS
5.- SALIR
Elija opción...1
```

¿Qué responde nuestro programa? Pues todo lo relacionado con la opción 1 que está en contenido de /usr/local/bin/soluciones.

La primera vez le dice lo siguiente:

```
COMPRUEBA QUE ESTÉ ENCENDIDO/CONECTADO EL ADAPTADOR
Se ha solucionado (s/n):
```

Si al usuario haciendo lo que se le sugiere ha solucionado el problema le dirá que s y terminaremos y si le dice que n, le ofrecerá otra solución.

Le dice que sí:

```
Se ha solucionado (s/n): s
Nos alegramos de que se haya solucionado el problema
```

Le dice que no a la primera sugerencia y si a la segunda:

```
Problemas con:
1.- INTERNET
2.- DATOS
3.- LENTO
4.- OTROS
5.- SALIR
Elija opción...1
COMPRUEBA QUE ESTÉ ENCENDIDO/CONECTADO EL ADAPTADOR
Se ha solucionado (s/n): n
UTILIZA OTRO NAVEGADOR
Se ha solucionado (s/n): s
Nos alegramos de que se haya solucionado el problema
```

Le dice que no a la primera sugerencia y no a la segunda (sólo hay dos en este caso pero podría haber varias, 20 ó 30 o más).

```
Problemas con:
1.- INTERNET
2.- DATOS
3.- LENTO
4.- OTROS
5.- SALIR
Elija opción...1
COMPRUEBA QUE ESTÉ ENCENDIDO/CONECTADO EL ADAPTADOR
Se ha solucionado (s/n): n
UTILIZA OTRO NAVEGADOR
Se ha solucionado (s/n): n
Sentimos no poder haberle ayudado. Tomamos nota
```

En este caso, añadimos a **/.logs/no_solucionado** algo así como lo siguiente:

```
antonio@US-ST:/usr/local/bin$ cat /.logs/no_solucionado
La alternativas actuales no solucionan el problema INTERNET
```

Esto lo estudiará el administrador para añadir más sugerencias a las soluciones actuales que tenemos en /usr/local/bin/soluciones. Pero no hará nada de momento.

Este funcionamiento es igual si se elige 1, 2 ó 3.

UT 04. Lenguajes de Scripts: actividades Bash (II)

b) Si el usuario elige la opción 4, ¿qué hace nuestro script?

Simplemente decirle que nos introduzca el problema.

```
Problemas con:
1.- INTERNET
2.- DATOS
3.- LENTO
4.- OTROS
5.- SALIR
Elija opción...4
Introduzca su problema: No veo bien la pantalla
Intentaremos solucionar sus problemas con No veo bien la pantalla cuanto antes. Gracias
```

Además, tenemos que añadir a `./logs/nuevos_problemas` esta incidencia. Algo así como lo siguiente:

```
antonio@US-ST:/usr/local/bin$ cat ./logs/nuevos_problemas
No veo bien la pantalla
```

Esto lo estudiará el administrador para añadir soluciones a problemas no contemplados actualmente. Pero no hará nada de momento.

c) Si elige 5, el programa terminará.

CÓDIGO

```
#!/bin/bash
```

```
solucionado=0 # variable que cambiamos si se soluciona el problema
```

```
expresion='^[sSnN]${'$'} # expresión regular para comprobaciones
```

```
# Función pausa para hacer el programa más interactivo
```

```
function pausa {
```

```
    echo
```

```
    echo "Pulse una tecla para continuar..."
```

```
    read tecla
```

```
}
```

```
# Menú
```

```
while true
```

```
do
```

```
    clear;
```

```
    echo "SOPORTE TÉCNICO BÁSICO"
```

```
    echo
```

```
    echo "Problemas con:"
```

```
    echo "1.- INTERNET."
```

```
    echo "2.- DATOS."
```

```
    echo "3.- LENTO."
```

```
    echo "4.- OTROS."
```

```
    echo "5.- SALIR."
```

```
    echo
```

```
    read -p "Elija una opción: " eleccion
```

UT 04. Lenguajes de Scripts: actividades Bash (II)

case \$eleccion in

Opción 1 -----

1)

for i in \$(grep ^1 /usr/local/bin/soluciones)

do

sugerencia=\$(echo \$i | cut -f3 -d":" | tr " " "\n")

problema=\$(echo \$i | cut -f2 -d":")

echo

echo \$sugerencia

echo

read -p "¿Se ha solucionado? (s/n): " respuesta

if [[! \$respuesta =~ \$expresion]]

then

echo "Incorrecto, debe responder (s/n)."

echo "Vuelva a ejecutar el programa."

exit 1

fi

if [\$respuesta = s] || [\$respuesta = S]

then

solucionado=1

echo

echo "Nos alegramos que se haya solucionado el

problema!"

echo

exit 0

fi

done

if [\$solucionado -eq 0]

then

echo

echo "Sentimos no poder ayudarle. Tomamos nota."

echo "Las alternativas actuales no solucionaron el

problema \$problema." >> /.logs/no_solucionado

fi

pausa;

;;

Opción 2 -----

2)

for i in \$(grep ^2 /usr/local/bin/soluciones)

do

sugerencia=\$(echo \$i | cut -f3 -d":" | tr " " "\n")

problema=\$(echo \$i | cut -f2 -d":")

echo

echo \$sugerencia

echo

UT 04. Lenguajes de Scripts: actividades Bash (II)

```

        read -p "¿Se ha solucionado? (s/n): " respuesta
        if [[ ! $respuesta =~ $expresion ]]
        then
            echo "Incorrecto, debe responder (s/n)."
            echo "Vuelva a ejecutar el programa."
            exit 1
        fi
        if [ $respuesta = s ] || [ $respuesta = S ]
        then
            solucionado=1
            echo
            echo "Nos alegramos que se haya solucionado el
problema!"

            echo
            exit 0
        fi
    done
    if [ $solucionado -eq 0 ]
    then
        echo
        echo "Sentimos no poder ayudarle. Tomamos nota."
        echo "Las alternativas actuales no solucionaron el
problema $problema." >> /.logs/no_solucionado
    fi
    pausa;
    ;;

# Opción 3 -----
3)
for i in $(grep ^3 /usr/local/bin/soluciones)
do
    sugerencia=$(echo $i | cut -f3 -d":" | tr "_ " " ")
    problema=$(echo $i | cut -f2 -d":")
    echo
    echo $sugerencia
    echo
    read -p "¿Se ha solucionado? (s/n): " respuesta
    if [[ ! $respuesta =~ $expresion ]]
    then
        echo "Incorrecto, debe responder (s/n)."
        echo "Vuelva a ejecutar el programa."
        exit 1
    fi
    if [ $respuesta = s ] || [ $respuesta = S ]
    then
        solucionado=1
        echo
    fi
done
```

UT 04. Lenguajes de Scripts: actividades Bash (II)

```

                                echo "Nos alegramos que se haya solucionado el
problema!"
                                echo
                                exit 0
                                fi
                                done
                                if [ $solucionado -eq 0 ]
                                then
                                    echo
                                    echo "Sentimos no poder ayudarle. Tomamos nota."
                                    echo "Las alternativas actuales no solucionaron el
problema $problema." >> /.logs/no_solucionado
                                    fi
                                    pausa;
                                    ;;

                                4)
                                echo
                                read -p "Introduzca su problema: " problem
                                echo
                                echo "Intentaremos solucionar sus problemas con: $problem.

Gracias."

                                echo $problem >> /.logs/nuevos_problemas
                                pausa;
                                ;;

                                5)
                                echo
                                echo "Hasta luego!"
                                echo
                                exit 0
                                ;;

                                *) echo "Opción incorrecta."
                                pausa;
                                ;;

                                esac

                                done
```


FUNCIONAMIENTO

```
edu@US-ST:~$ edu_st.sh █
```

SOPORTE TÉCNICO BÁSICO

Problemas con:

- 1.- INTERNET.
- 2.- DATOS.
- 3.- LENTO.
- 4.- OTROS.
- 5.- SALIR.

Elija una opción: 1

COMPRUEBA QUE ESTÉ ENCENDIDO/CONECTADO EL ADAPTADOR

¿Se ha solucionado? (s/n): n

UTILIZA OTRO NAVEGADOR

¿Se ha solucionado? (s/n): s

Nos alegramos que se haya solucionado el problema!

```
edu@US-ST:~$ edu_st.sh █
```

SOPORTE TÉCNICO BÁSICO

Problemas con:

- 1.- INTERNET.
- 2.- DATOS.
- 3.- LENTO.
- 4.- OTROS.
- 5.- SALIR.

Elija una opción: 3

ELIMINA ARCHIVOS INNECESARIOS

¿Se ha solucionado? (s/n): n

DESFRACTA EL DISCO

¿Se ha solucionado? (s/n): n

Sentimos no poder ayudarle. Tomamos nota.

Pulse una tecla para continuar...