

Comenzado el	lunes, 11 de octubre de 2021, 20:11
Estado	Finalizado
Finalizado en	lunes, 11 de octubre de 2021, 20:37
Tiempo empleado	25 minutos 22 segundos
Puntos	12,92/23,00
Calificación	5,62 de 10,00 (56%)

Pregunta 1

Correcta

Se puntúa 1,00 sobre 1,00

Asocie los siguientes operadores de bash con sus significados o comandos equivalentes.

[expresión]	Equivale a: test expresión	✓
echo \$((expresión))	Equivale a: expr expresión	✓
echo cadena >> archivo	Escribe "cadena" al final de "archivo"	✓
echo cadena > archivo	Sobreescribe "archivo" con el string "cadena"	✓
echo cadena grep archivo	Busca el string "archivo" dentro del string "cadena"	✓
\$(./archivo)	Equivale a: `./archivo`	✓
cadena=`./archivo`	Guarda lo que imprime "archivo" al ejecutarse adentro de la variable "cadena"	✓

La respuesta correcta es: [expresión] → Equivale a: test expresión, echo \$((expresión)) → Equivale a: expr expresión, echo cadena >> archivo → Escribe "cadena" al final de "archivo", echo cadena > archivo → Sobreescribe "archivo" con el string "cadena", echo cadena | grep archivo → Busca el string "archivo" dentro del string "cadena", \$(./archivo) → Equivale a: `./archivo`, cadena=`./archivo` → Guarda lo que imprime "archivo" al ejecutarse adentro de la variable "cadena"

Pregunta 2

Correcta

Se puntúa 1,00 sobre 1,00

¿Con que simbolo se declara un comentario?

- Seleccione una:
- ☐ a. ?
 - ☒ b. # ✓
 - ☐ c. %
 - ☐ d. ;
 - ☐ e. //

La respuesta correcta es: #

Pregunta **3**

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuales de las siguientes sintaxis de funciones es correcta?

Seleccione una:

- ☐ a.
funtion mayor() {
if [\$1 > \$2]; then echo \$1; else echo \$2; fi
}
- ☐ b.
mayor(a,b) {
if [\$a > \$b]; then echo \$a; else echo \$b; fi
}
- ☒ c.
mayor() {
if [\$1 > \$2]; then echo \$1; else echo \$2; fi
} ✓

La respuesta correcta es:

```
mayor() {  
if [ $1 > $2 ]; then echo $1; else echo $2; fi  
}
```

Pregunta **4**

Incorrecta

Se puntúa 0,00 sobre 1,00

¿Cuales de las siguientes sintaxis del comando for son correctas?

Seleccione una o más de una:

- ☐ a.
for i in `seq 1 100`
do
echo "\$i"
done
- ☒ b.
for \$i in `seq 1 100`
do
echo "\$i"
done ✗
- ☐ c.
for i in `seq 1 100` do
echo "\$i"
done
- ☒ d.
for i in `seq 1 100`; do echo "\$i"; done ✓
- ☒ e.
for i in \$(seq 1 100); do echo "\$i"; done ✓

Las respuestas correctas son:

```
for i in `seq 1 100`  
do  
echo "$i"  
done,  
for i in `seq 1 100`; do echo "$i"; done,  
for i in $(seq 1 100); do echo "$i"; done
```

Pregunta **5**

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuales de las siguientes sintaxis del comando if son correctas?

Seleccione una o más de una:

- ☐ a. if test -d /home/pepe then echo "es un directorio" fi
- ☒ b. if test -r /home/pepe; then echo "Tengo permisos de lectura"; fi ✓
- ☐ c.
if ["\$nombre" == "Maria"]
then echo "Es igual"
fi
- ☒ d.
if ["\$nombre" == "Maria"]
then
echo "Es igual"
fi ✓

Las respuestas correctas son: if test -r /home/pepe; then echo "Tengo permisos de lectura"; fi,
if ["\$nombre" == "Maria"]
then
echo "Es igual"
fi

Pregunta **6**

Incorrecta

Se puntúa 0,00 sobre 1,00

¿Cuales de los siguientes usos de arreglos son correctos?

Seleccione una o más de una:

- ☐ a. arreglo = (3 4 5)
- ☒ b. echo \${arreglo[1]} ✓
- ☐ c. echo \$arreglo[2]
- ☒ d. echo \${#arreglo[*]} ✓
- ☒ e. arreglo=(3 4 5) ✓
- ☐ f. agregar un nuevo elemento: arreglo+=(5)
- ☒ g. agregar un nuevo elemento: arreglo+=5 ✗

Las respuestas correctas son: echo \${arreglo[1]}, echo \${#arreglo[*]}, arreglo=(3 4 5), agregar un nuevo elemento: arreglo+=(5)

Pregunta **7**

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuales de los siguientes usos de expr es correcto?

Seleccione una o más de una:

- ☒ a. expr 4 + 5 ✓
- ☐ b. expr 4 5 *
- ☐ c. expr - 6 5
- ☒ d. expr 5 != 5 ✓
- ☒ e. expr length "PEPE" ✓

Las respuestas correctas son: expr 4 + 5, expr 5 != 5, expr length "PEPE"

Pregunta **8**

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuales de los siguientes usos de variables son correctos?

Seleccione una o más de una:

- ☐ a. var NOMBRE="pepe"
- ☒ b. echo \$NOMBRE ✓
- ☐ c. APELLIDO = "sanchez"
- ☐ d. echo APELLIDO
- ☒ e. DIRECCION="56 nro 436" ✓
- ☒ f. echo \${DIRECCION} ✓

Las respuestas correctas son: echo \$NOMBRE, DIRECCION="56 nro 436", echo \${DIRECCION}

Pregunta **9**

Correcta

Se puntúa 1,00 sobre 1,00

¿Que comillas se utilizan para la hacer uso de la sustitucion de comandos?

Seleccione una:

- ☐ a. " "
- ☐ b. ' '
- ☒ c. `` ✓
- ☐ d. ^ ^

La respuesta correcta es: ``

Pregunta **10**

Incorrecta

Se puntúa -1,00 sobre 1,00

¿Que hace el comando "find / –name pepe 2> /dev/null" ?

Seleccione una:

- ☐ a. crea un archivo vacio en /dev/null
- ☒ b. envia la salida del comando al /dev/null ✗
- ☐ c. muestra los archivos llamados pepe, siempre y cuando tenga permisos de acceso a ellos

La respuesta correcta es: muestra los archivos llamados pepe, siempre y cuando tenga permisos de acceso a ellos

Pregunta **11**

Correcta

Se puntúa 1,00 sobre 1,00

¿Que hace el comando test -w /home/pepe/ ?

Seleccione una:

- ☐ a. Evalua si el archivo o directorio existe
- ☐ b. Evalua si el archivo o directorio existe y es un directorio
- ☐ c. Evalua si el archivo o directorio existe y se tiene el permiso de lectura
- ☒ d. Evalua si el archivo o directorio existe y se tiene el permiso de escritura ✓
- ☐ e. Evalua si el archivo o directorio existe y es un archivo regular

La respuesta correcta es: Evalua si el archivo o directorio existe y se tiene el permiso de escritura

Pregunta **12**

Correcta

Se puntúa 1,00 sobre 1,00

¿Que información tiene la variable \$#?

Seleccione una:

- ☐ a. Ninguna es solo un comentario
- ☐ b. La cantidad de arreglos utilizados en el script
- ☒ c. La cantidad de parametros que se enviaron ✓
- ☐ d. La cantidad de variables utilizadas en el script

La respuesta correcta es: La cantidad de parametros que se enviaron

Pregunta **13**

Incorrecta

Se puntúa -1,00 sobre 1,00

¿Que resultado tiene el siguiente comando "cat /etc/passwd | cut -f1 -d: | grep "^a" " ?

Seleccione una:

- ☐ a. Imprime los homes de los usuarios que tienen una letra a
- ☐ b. Imprime los nombre de usuario que contenga una letra a
- ☐ c. No imprime nada
- ☒ d. Imprime las password de los usuarios que contenga una letra a ✗
- ☐ e. Imprime los nombres de los usuarios que empiecen con la letra a

La respuesta correcta es: Imprime los nombres de los usuarios que empiecen con la letra a

Pregunta **14**

Correcta

Se puntúa 1,00 sobre 1,00

Dado el siguiente script, ¿Que afirmaciones son verdaderas acerca de su ejecucion?

```
#!/bin/bash
for i in {1..100}; do
  while true; do
    if ! (($i % 25)); then
      echo "$i es divisible por 25"
      continue 2
    elif [ $i -eq 53 ]; then
      break 2
    elif [ "$i % $i" ]; then
      break
    fi
  done
done
```

Seleccione una o más de una:

- ☐ a. Es un bucle infinito
- ☐ b. Imprime todos los valores del 1 al 100 divisibles por 25
- ☒ c. Cuando el valor de i llega al 53, el script termina. ✓
- ☒ d. Imprime que los valores 25 y 50 son divisibles por 25 ✓

Las respuestas correctas son: Cuando el valor de i llega al 53, el script termina., Imprime que los valores 25 y 50 son divisibles por 25

Pregunta **15**

Correcta

Se puntúa 1,00 sobre 1,00

¿Cuáles de las siguientes son funciones correctamente definidas?

Seleccione una o más de una:

☐

a.

```
function x(1, 2){
    echo $1 | grep $2
}
```

☒

b.

```
function x{
    echo $1 | grep $2
}
```

☐

c.

```
x(1, 2){
    echo $1 | grep $2
}
```

☐

d.

```
function x(1, 2); do
    echo $1 | grep $2
done
```

☒

e.

```
x(){
    echo $1 | grep $2
}
```



Las respuestas correctas son:

```
function x{
    echo $1 | grep $2
}
```

```
x(){
    echo $1 | grep $2
}
```

Pregunta **16**

Parcialmente
correcta

Se puntúa 0,25
sobre 1,00

¿Cuales de las siguientes equivalencias son ciertas?

Seleccione una o más de una:

☐

a.

```
cat archivo | wc -c
# equivale a:
wc -c archivo | cut -d' ' -f1
```

☒

b.

```
find -name archivo
# equivale a:
find . -name archivo
```

☐

c.

```
grep ac archivo
# equivale a:
cat archivo | grep ac
```

☐

d.

```
find -name archivo
# equivale a:
echo archivo | find -name
```

☐

e.

```
echo hola | cat > salida
# equivale a:
echo hola > salida
```

☐

f.

```
echo hola | cat > salida
# equivale a:
cat hola > salida
```

Las respuestas correctas son:

```
cat archivo | wc -c
# equivale a:
wc -c archivo | cut -d' ' -f1
```

```
find -name archivo
# equivale a:
find . -name archivo
```

```
grep ac archivo
# equivale a:
cat archivo | grep ac
```

```
echo hola | cat > salida
# equivale a:
echo hola > salida
```

Pregunta **17**

Incorrecta

Se puntúa 0,00 sobre 1,00

Qué opciones son verdaderas respecto a la secuencia de comandos:

```
(test -f archivo && grep menta archivo && echo Z) || echo Q
```

Seleccione una o más de una:

- ☐ a. Siempre imprime "Z"
- ☒ b. Si "archivo" existe y contiene el string "menta" imprime "Z", sino imprime "Q" ✓
- ☐ c. Siempre imprime "Q"
- ☐ d. Sin los paréntesis el resultado sería el mismo
- ☒ e. Sin los paréntesis el resultado sería distinto ✗
- ☐ f. Si "archivo" existe y contiene el string "menta" imprime "Q", sino imprime "Z"

Las respuestas correctas son: Si "archivo" existe y contiene el string "menta" imprime "Z", sino imprime "Q", Sin los paréntesis el resultado sería el mismo

Pregunta **18**

Correcta

Se puntúa 1,00 sobre 1,00

Supongamos que tenemos un archivo llamado "archivo" con el siguiente contenido:

```
eth2 Link encap:Ethernet HWaddr 00:e0:7d:b4:1c:38
inet addr:192.168.1.112 Bcast:255.255.255.255 Mask:255.255.255.0
inet6 addr: fe80::2e0:7dff:feb4:1c38/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:115588267 errors:0 dropped:0 overruns:0 frame:0
TX packets:35601221 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:3240134640 (3.2 GB) TX bytes:2042178372 (2.0 GB)
Interrupt:20 Base address:0xce00
```

que resultado se obtiene al ejecutar:

```
cat archivo | grep inet | cut -d: -f2 | cut -d" " -f1
```

Seleccione una:

- ☐ a. Ningun resultado
- ☐ b. fe80::2e0:7dff:feb4:1c38/64
- ☐ c. 255.255.255.255
- ☒ d. 192.168.1.112 ✓
- ☐ e. 00:e0:7d:b4:1c:38

La respuesta correcta es: 192.168.1.112

Pregunta **19**

Parcialmente
correcta

Se puntúa 0,67
sobre 1,00

Se desea hacer un script que imprima la lista de argumentos que recibe. ¿Cuales de las siguientes implementaciones son correctas?.

Seleccione una o más de una:

☐

a.

```
#!/bin/sh
for arg in argv; do
    echo $arg
done
```

☐

b.

```
#!/bin/sh
for arg in ${argv[*]}; do
    echo $arg
done
```

☐

c.

```
#!/bin/sh
echo $*
```

☐

d.

```
#!/bin/sh
echo $#
```

☒

e.

```
#!/bin/sh
for arg in $*; do
    echo $arg
done
```

☒

f.

```
#!/bin/sh
for arg in $@; do
    echo $arg
done
```



Las respuestas correctas son:

```
#!/bin/sh
echo $*
```

```
#!/bin/sh
for arg in $*; do
    echo $arg
done
```

```
#!/bin/sh
for arg in $@; do
    echo $arg
done
```

Pregunta **20**

Incorrecta

Se puntúa 0,00 sobre 1,00

Se desea hacer un script que imprima su primer argumento en pantalla, por ejemplo:

`$./mi_script.sh "Hola Mundo"`

debe imprimir "Hola Mundo" (sin las comillas).

¿Cuál de las siguientes es una implementación que cumple este objetivo?

Seleccione una o más de una:



a.

```
#!/bin/sh
echo argv[1]
```



b.

```
#!/bin/sh
echo $argv[1]
```



c.

```
#!/bin/sh
echo ${argv[1]}
```



d.

```
#!/bin/sh
echo $1
```



e.

```
#!/bin/sh
echo argv[0]
```



f.

```
#!/bin/sh
echo ${argv[0]}
```



g.

```
#!/bin/sh
echo $0
```



h.

```
#!/bin/sh
for i; do
echo $i
break
done
```

Las respuestas correctas son:

```
#!/bin/sh
echo $1
```

```
#!/bin/sh
for i; do
echo $i
break
done
```

Pregunta **21**

Correcta

Se puntúa 1,00 sobre 1,00

¿Qué hace la siguiente secuencia de comandos?

```
tar -czvf zip.tar.gz $(ls); tar -xvfz zip.tar.gz -C /tmp
```

Seleccione una:

- ☐ a. Genera un archivo empaquetado con el contenido del directorio actual.
- ☐ b. Falla, no se le da una lista de archivos
- ☐ c. Genera un archivo comprimido con el contenido del directorio actual y lo mueve al directorio /tmp
- ☒ d. Genera un archivo comprimido con el contenido del directorio actual y lo descomprime en /tmp ✓

La respuesta correcta es: Genera un archivo comprimido con el contenido del directorio actual y lo descomprime en /tmp

Pregunta **22**

Correcta

Se puntúa 1,00 sobre 1,00

¿Qué hace el siguiente script?

```
#!/bin/sh
a=$(wc -c xyz | cut -d' ' -f1)
expr $a / 1024 / 1024
```

Seleccione una:

- ☒ a. Imprime el tamaño en MiB del archivo "xyz" ✓
- ☐ b. Cuenta la cantidad de caracteres en el string "xyz" y la pasa a MiB (obviamente da cero)
- ☐ c. Comprime el archivo "xyz" a "xyz.wc" e imprime el tamaño en MiB del archivo comprimido.
- ☐ d. Falla ya que "wc" debe recibir los datos por la entrada estándar.
- ☐ e. Falla porque el valor calculado por "expr" no es almacenado en ninguna variable.

La respuesta correcta es: Imprime el tamaño en MiB del archivo "xyz"

Pregunta **23**

Sin contestar

Puntúa como 1,00

¿Qué imprime la siguiente secuencia de comandos? (Responder solamente el resultado de la secuencia de comandos, sin explicaciones).

```
echo 1zTrao dpaoqda | tr Taqz1rd irbuQea
```

Respuesta:



La respuesta correcta es: Quiero aprobar

◀ Comando diff, tar y arreglos en UNIX

Ir a...

Simulador ▶