Ejercicios de scripts de linux

Para todos los ejercicios se pide que estén comentados tanto al principio del script como con comentarios intermedios que aclaren el código. Además se pide que el código esté ordenado, sea claro y tenga indentación. El alumno subirá el código y lo colocará debajo del enunciado.

1º a) Realiza un script que pida por teclado una nota que esté comprendida entre 0 y 10 (si no lo está mostrar mensaje de error). Si la nota es mayor del 8, se le informará al usuario que el ejercicio está superado. Si la nota está entre 5 y 8 se le dirá al usuario que aunque ha aprobado no es suficiente para avanzar al siguiente nivel. Y si es menor que 5 se le informará que debe estudiar más los contenidos teóricos y volverse a examinar.

|  |
| --- |
| #!/bin/bash  # Programa: Realiza un script que pida por teclado una nota que esté comprendida entre 0 y 10 (si no lo está mostrar mensaje de error). Si la nota es mayor del 8, se le informará al usuario que el ejercicio está superado. Si la nota está entre 5 y 8 se le dirá al usuario que aunque ha aprobado no es suficiente para avanzar al siguiente nivel. Y si es menor que 5 se le informará que debe estudiar más los contenidos teóricos y volverse a examinar.  # Autor: Emmanuel Ramos Biado  # Fecha: 28-4-2022  # Limpia la pantalla  clear  numero=0  # Introduce el número  **read** -p "Introduce un número entre el 0 y el 10: " numero  # Si el número es menor que 0 o mayor que 10  **if** [ $numero -lt 0 ] || [ $numero -gt 10 ]  **then**  **echo** "Error"  # Si el número se encuentra entre el 5 y el 8  **elif** [ $numero -ge 5 ] && [ $numero -le 8 ]  **then**  **echo** "Aunque has aprobado, no es suficiente para avanzar de nivel."  # Si el número es menor que 5  **elif** [ $numero -lt 5 ]  **then**  **echo** "Debes estudiar más los contenidos teóricos y volverte a examinar"  # Si el número es mayor que 8  **elif** [ $numero -gt 8 ]  **then**  **echo** "El ejercicio está superado"  **fi** |

b) Se modifica la versión anterior para que se quede en un bucle preguntando la nota hasta que pase de nivel (con nota mayor de 8 y válida)

|  |
| --- |
| #!/bin/bash  # Programa: Se modifica la versión anterior para que se quede en un bucle preguntando la nota hasta que pase de nivel (con nota mayor de 8 y válida)  # Autor: Emmanuel Ramos Biado  # Fecha: 28-4-2022  # Limpia la pantalla  clear  numero=0  # Bucle  **until** [ $numero -gt 8 ] && [ $numero -le 10 ]  **do**  # Introduce el número  **read** -p "Introduce un número entre el 0 y el 10: " numero  # Si el número es menor que 0 o mayor que 10  **if** [ $numero -lt 0 ] || [ $numero -gt 10 ]  **then**  **echo** "Error"  # Si el número se encuentra entre el 5 y el 8  **elif** [ $numero -ge 5 ] && [ $numero -le 8 ]  **then**  **echo** "Aunque has aprobado, no es suficiente para avanzar de nivel."  # Si el número es menor que 5  **elif** [ $numero -lt 5 ]  **then**  **echo** "Debes estudiar más los contenidos teóricos y volverte a examinar"  # Si el número es mayor que 8  **elif** [ $numero -gt 8 ]  **then**  **echo** "El ejercicio está superado"  **fi**  **done** |

2º a) Se realizará un menú que tendrá las siguientes opciones:

1. Visualizar la fecha actual
2. Visualizar el contenido de la variable PATH
3. Visualizar el espacio en disco local
4. Salir (saldrá un mensaje de usuario diciendo “Progama finalizado”

|  |
| --- |
| #!/bin/bash  # Programa: Se realizará un menú que tendrá las siguientes opciones:  # 1. Visualizar la fecha actual  # 2. Visualizar el contenido de la variable PATH  # 3. Visualizar el espacio en disco local  # 4. Salir (saldrá un mensaje de usuario diciendo “Progama finalizado”  # Autor: Emmanuel Ramos Biado  # Fecha: 28-4-2022  # Limpia la pantalla  clear  opcion=0  #Limpia la pantalla  clear  # Muestra el menú de opciones  **echo** "1. Visualizar la fecha actual"  **echo** "2. Visualizar el contenido de la variable PATH"  **echo** "3. Visualizar el espacio en disco local"  **echo** "4. Salir"  # Introduce el número  **read** -p "Elige una opción [1-4]: " opcion  # En función de la opción seleccionada hará diferentes acciones  **case** $opcion **in**  # Muestra la fecha  1) date  **read**;;    # Muestra las variables de entorno  2) env  **read**;;    # Muestra el espacio del disco local  3) df -h  **read**;;    # Salida del programa  4) **echo** "Programa finalizado" ;;    # Esta opción sirve si has introducido una opción errónea  \*) **echo** "Error. Opción incorrecta" ;;    **esac** |

b) Se mejora la versión anterior para meterlo en un bucle y solo se saldrá al pulsar la opción 4. Se limpiará la pantalla al principio y cada vez que elijan una nueva opción.

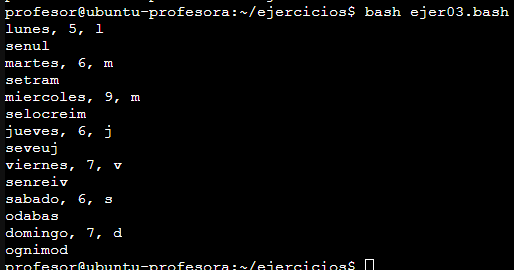
|  |
| --- |
| #!/bin/bash  # Programa: Se realizará un menú que tendrá las siguientes opciones:  # 1. Visualizar la fecha actual  # 2. Visualizar el contenido de la variable PATH  # 3. Visualizar el espacio en disco local  # 4. Salir (saldrá un mensaje de usuario diciendo “Progama finalizado”  # Autor: Emmanuel Ramos Biado  # Fecha: 28-4-2022  # Limpia la pantalla  clear  opcion=0  # Bucle  **until** [ $opcion -eq 4 ]  **do**  #Limpia la pantalla  clear    # Muestra el menú de opciones  **echo** "1. Visualizar la fecha actual"  **echo** "2. Visualizar el contenido de la variable PATH"  **echo** "3. Visualizar el espacio en disco local"  **echo** "4. Salir"    # Introduce el número  **read** -p "Elige una opción [1-4]: " opcion    # En función de la opción seleccionada hará diferentes acciones  **case** $opcion **in**  # Muestra la fecha  1) date  **read**;;    # Muestra las variables de entorno  2) env  **read**;;    # Muestra el espacio del disco local  3) df -h  **read**;;    # Salida del programa  4) **echo** "Programa finalizado" ;;    # Esta opción sirve si has introducido una opción errónea  \*) **echo** "Error. Opción incorrecta" ;;    **esac**  **done** |

3º Se recorrerá un bucle con los días de las semana y para cada día se mostrará el día así como la longitud del día y su inicial en una línea y en otra línea, se mostrará el día al revés. Para ello usar estas funciones:

${string:0:1} , me da el primer carácter; ${string:1:1} , me da el segundo carácter

${#string} da la longitud del string

Deberá mostrarse algo así:



|  |
| --- |
| #!/bin/bash  # Programa: Se recorrerá un bucle con los días de las semana y para cada día se mostrará el día así como la longitud del día y su inicial en una línea y en otra línea, se mostrará el día al revés.  # Autor: Emmanuel Ramos Biado  # Fecha: 29-4-2022  # Limpieza de la pantalla  clear  dia=""  diareves=""  **for** dia **in** lunes martes miercoles jueves viernes sabado domingo  **do**  diareves=""  **echo** "$dia, ${#dia}, ${dia:0:1}"  **let** longitud=${#dia}  **while** [ $longitud -gt 0 ]  **do**  **let** longitud=$longitud-1  diareves+=${dia:$longitud:1}  **done**  **echo** $diareves  **done** |

4º Se hará un programa que admite 2 parámetros. Si se ponen más o menos parámetros al llamarlo se dará error y se acabará el programa.

Los dos parámetros serán un directorio y un fichero. Se comprobará si el primero es un directorio y si es así en el mismo directorio se creará el fichero vacío si no existe. En caso de que exista se informará y no se creará.

Si el directorio no existe se le preguntará al usuario si desea crearlo y si es así lo creará y se hará el mismo proceso que en el paso anterior para el fichero.

|  |
| --- |
| #!/bin/bash  # Programa: Se hará un programa que admite 2 parámetros. Si se ponen más o menos parámetros al llamarlo se dará error y se acabará el programa. Los dos parámetros serán un directorio y un fichero. Se comprobará si el primero es un directorio y si es así en el mismo directorio se creará el fichero vacío si no existe. En caso de que exista se informará y no se creará. Si el directorio no existe se le preguntará al usuario si desea crearlo y si es así lo creará y se hará el mismo proceso que en el paso anterior para el fichero.  # Autor: Emmanuel Ramos Biado  # Fecha: 29-4-2022  # Limpieza de la pantalla  clear  opcion=""  # Si no se introducen 2 parámetros  **if** [ $# -ne 2 ]; **then**  **echo** "Error. Debes introducir un total de 2 parámetros"  **exit**  **fi**  # Si existe el directorio  **if** [ -d $1 ]; **then**  **echo** "Ya existe un directorio con el mismo nombre"  # Si existe el fichero  **if** [ -f **.**/$1/$2 ]; **then**  **echo** "Ya existe un fichero con el mismo nombre"  **else**  # Si no existe el fichero  **read** -p "No existe ese fichero, ¿desea crearlo? (S/N)" opcion  **if** [ $opcion=="S" ] || [ $opcion=="**s**" ]; then  touch ./$1/$2  fi  fi    # Si no existe el directorio  else  read -p "No existe ese directorio, ¿desea crearlo? (S/N)" opcion    if [ "$opcion"=="S" ] || [ "$opcion"=="s" ]; then  mkdir ./$1    # Si existe el fichero  if [ -f ./$1/$2 ]; then  echo "Ya existe un fichero con el mismo nombre"  fi    # si no existe el fichero  else  read -p "No existe ese fichero, ¿desea crearlo? (S/N)" opcion  if [ "$opcion"=="S" ] || [ "$opcion"=="s" ]; then  touch ./$1/$2  fi  fi  fi |

5º Se hará un programa que admita un número variable de parámetros.

Primero se informará del número de parámetros introducidos. Luego, se recorrerá con un bucle for y por cada uno de ellos creará un fichero vacío si es que no existe. Los ficheros se crearán en un directorio llamado *misficheros5,* que en caso de que no exista se creará al principio.(sólo si se ha introducido algún parámetro). En caso de crear el directorio se informará que se ha creado y en caso de que el directorio ya exista se informará que los ficheros se van a crear en ese directorio.

Además por cada fichero se informará si se crea o no se crea porque ya existe y se parará el programa para que el usuario pulse una tecla.

|  |
| --- |
| #!/bin/bash  # Programa: Se hará un programa que admita un número variable de parámetros. Primero se informará del número de parámetros introducidos. Luego, se recorrerá con un bucle for y por cada uno de ellos creará un fichero vacío si es que no existe. Los ficheros se crearán en un directorio llamado misficheros5, que en caso de que no exista se creará al principio.(sólo si se ha introducido algún parámetro). En caso de crear el directorio se informará que se ha creado y en caso de que el directorio ya exista se informará que los ficheros se van a crear en ese directorio. Además por cada fichero se informará si se crea o no se crea porque ya existe y se parará el programa para que el usuario pulse una tecla.  # Autor: Emmanuel Ramos Biado  # Fecha: 29-4-2022  # Limpieza de la pantalla  clear  # Indica el número de parámetros introducidos  **echo** "Se ha introducido $# parámetros"  # Crea el directorio misficheros5 si no existe  **if** **!** [ -d misficheros5 ]; **then**  mkdir **.**/misficheros5  **echo** "Se ha creado el directorio misficheros5"  **else**  **echo** "Los archivos se van a introducir en el directorio misficheros5"  **fi**  # Creación de los ficheros  **for** i **in** $\*  **do**  **if** [ -f **.**/misficheros5/$i ]; **then**  **echo** "Ese fichero ya existe"  **else**  touch **.**/misficheros5/$i  **echo** "Se ha creado el fichero $i"  **done** |