

[Mis cursos](#) / [SA - ASO I](#) / [Parciales](#) / [2º Parcial](#)

Comenzado el	lunes, 9 de diciembre de 2024, 17:16
Estado	Finalizado
Finalizado en	lunes, 9 de diciembre de 2024, 19:36
Tiempo empleado	2 horas 20 minutos
Calificación	77,0 de 100,0

Pregunta **1**

Finalizado

Se puntúa 3,5 sobre 3,5

El comando grep en GNU/Linux se utiliza principalmente para:

- ☐ a. Modificar el contenido de un archivo reemplazando patrones específicos.
- ☒ b. Filtrar líneas en un archivo o entrada estándar que coincidan con un patrón de texto.
- ☐ c. Buscar archivos duplicados en un directorio.
- ☐ d. Mover archivos entre diferentes directorios con opciones de filtrado.

Respuesta correcta

Pregunta **2**

Finalizado

Se puntúa 3,5 sobre 3,5

¿Cuál es la función principal de los comandos **head** y **tail** en GNU/Linux?

- ☐ a. Ordenar el contenido de un archivo en orden alfabético.
- ☐ b. Dividir un archivo grande en múltiples archivos más pequeños.
- ☒ c. Mostrar, respectivamente, las primeras y últimas líneas de un archivo o flujo de datos.
- ☐ d. Comparar dos archivos y mostrar las diferencias entre ellos.

Respuesta correcta

Pregunta **3**

Finalizado

Se puntúa 3,5 sobre 3,5

Tienes un archivo llamado **nombres.txt** con el siguiente contenido:

```
Carlos  
Ana  
Pedro  
Beatriz
```

Si ejecutas el comando **sort nombres.txt**, ¿qué resultado obtendrás?

- ☐ a. Pedro
Carlos
Beatriz
Ana
- ☐ b. Beatriz
Carlos
Ana
Pedro
- ☒ c. Ana
Beatriz
Carlos
Pedro
- ☐ d. Carlos
Ana
Pedro
Beatriz

Respuesta correcta

Pregunta **4**

Finalizado

Se puntúa 3,5 sobre 3,5

¿Cuál es el propósito principal de usar los comandos **tar** y **gzip** juntos en GNU/Linux?

- ☐ a. Comparar el contenido de dos archivos y encontrar diferencias entre ellos.
- ☐ b. Convertir un archivo binario a texto legible.
- ☐ c. Dividir un archivo grande en partes más pequeñas y numeradas.
- ☒ d. Combinar varios archivos en un único archivo comprimido para ahorrar espacio y facilitar su transporte.

Respuesta correcta

Pregunta **5**

Finalizado

Se puntúa 3,5 sobre 3,5

¿Qué es un sistema de archivos en el contexto de un sistema operativo?

- ☐ a. Es un tipo de archivo especial que contiene comandos ejecutables del sistema.
- ☒ b. Es el software que gestiona cómo se almacenan, organizan y recuperan los datos en un dispositivo de almacenamiento.
- ☐ c. Es el conjunto de herramientas que permite al usuario ejecutar aplicaciones en la terminal.
- ☐ d. Es una partición de disco configurada para ejecutar múltiples sistemas operativos.

Respuesta correcta

Pregunta **6**

Finalizado

Se puntúa 3,5 sobre 3,5

¿Cuál es el propósito principal del Virtual File System (VFS) en GNU/Linux?

- ☐ a. Crear copias de seguridad automáticas de todos los archivos del sistema.
- ☒ b. Proveer una interfaz abstracta que permite al sistema operativo acceder a diferentes sistemas de archivos de manera uniforme.
- ☐ c. Optimizar el espacio en disco al comprimir automáticamente los archivos almacenados.
- ☐ d. Convertir archivos binarios en texto legible para los usuarios del sistema.

Respuesta correcta

Pregunta **7**

Finalizado

Se puntúa 3,5 sobre 3,5

En el estándar Filesystem Hierarchy Standard (FHS), ¿qué tipo de contenido se almacena generalmente en el directorio **/var**?

- ☐ a. Archivos ejecutables de comandos básicos del sistema operativo.
- ☐ b. Archivos de configuración estática del sistema operativo.
- ☐ c. Librerías compartidas necesarias para ejecutar aplicaciones.
- ☒ d. Archivos de datos variables como logs, colas de impresión y datos de aplicaciones en ejecución.

Respuesta correcta

Pregunta **8**

Finalizado

Se puntúa 3,5 sobre 3,5

Tienes el archivo **original.txt** en tu directorio actual, y ejecutas el siguiente comando:

ln -s original.txt enlace.txt

¿Qué sucede al ejecutar el comando?

- ☒ a. Se crea un enlace simbólico llamado **enlace.txt**, que apunta a **original.txt**.
- ☐ b. Se elimina **original.txt** y se renombra a **enlace.txt**.
- ☐ c. Se crea una copia independiente de **original.txt** llamada **enlace.txt**.
- ☐ d. Se fusionan los contenidos de **original.txt** y **enlace.txt** en un único archivo.

Respuesta correcta

Pregunta **9**

Finalizado

Se puntúa 3,5 sobre 3,5

¿Cuál de las siguientes afirmaciones sobre la gestión de usuarios en GNU/Linux es correcta?

- ☐ a. Al eliminar un usuario con **userdel**, su carpeta personal siempre se elimina automáticamente.
- ☐ b. El comando **useradd** y **adduser** realizan exactamente las mismas acciones y no presentan diferencias.
- ☐ c. No es posible modificar el grupo principal de un usuario después de su creación.
- ☒ d. El archivo **/etc/passwd** contiene información básica sobre los usuarios, como su nombre y UID.

Respuesta correcta

Pregunta **10**

Finalizado

Se puntúa 3,5 sobre 3,5

¿Cuál de las siguientes afirmaciones sobre la gestión de grupos en GNU/Linux es correcta?

- ☐ a. Un usuario solo puede pertenecer a un grupo a la vez.
- ☐ b. No es posible agregar un usuario a un grupo después de su creación.
- ☒ c. El archivo **/etc/group** contiene información sobre todos los grupos del sistema y sus miembros.
- ☐ d. El comando **groupdel** puede eliminar cualquier grupo, incluso si todavía tiene miembros asignados.

Respuesta correcta

Pregunta **11**

Finalizado

Se puntúa 0,0 sobre 3,5

¿Cuál de las siguientes afirmaciones sobre el archivo **/etc/passwd** es correcta?

- ☐ a. Ninguna de las opciones es correcta.
- ☐ b. Almacena las configuraciones del sistema de archivos para cada usuario.
- ☒ c. Contiene las contraseñas cifradas de los usuarios del sistema.
- ☐ d. Es utilizado para registrar las actividades de los usuarios en el sistema.

Respuesta incorrecta.

Pregunta **12**

Finalizado

Se puntúa 3,5 sobre 3,5

Supón que tienes un archivo llamado **documento.txt** y necesitas que su propietario sea cambiado a **juan** y que todos los usuarios del sistema solo puedan leerlo. ¿Qué combinación de comandos puedes utilizar para lograr esto?

☐ a.

```
chown juan documento.txt  
chmod 755 documento.txt
```

☐ b.

```
chmod juan documento.txt  
chown 444 documento.txt
```

☒ c.

```
chown juan documento.txt  
chmod 444 documento.txt
```

☐ d.

```
chmod 744 documento.txt  
chown juan documento.txt
```

Respuesta correcta

Pregunta **13**

Finalizado

Se puntúa 3,5 sobre 3,5

¿Cuál es el propósito principal de los parámetros en un script de Bash?

- ☐ a. Asignar variables globales accesibles por todos los scripts del sistema.
- ☒ b. Proveer un mecanismo para recibir entradas del usuario al momento de ejecutar el script.
- ☐ c. Permitir la ejecución de scripts en segundo plano sin interacciones del usuario.
- ☐ d. Almacenar los resultados de comandos ejecutados dentro del script.

Respuesta correcta

Pregunta **14**

Finalizado

Se puntúa 3,5 sobre 3,5

¿Qué operador lógico se utiliza en Bash para evaluar si dos condiciones son verdaderas al mismo tiempo?

- ☒ a. &&
- ☐ b. !
- ☐ c. ==
- ☐ d. ||

Respuesta correcta

Pregunta **15**

Finalizado

Se puntúa 3,5 sobre 3,5

¿Cuál de las siguientes afirmaciones sobre variables de entorno en GNU/Linux es correcta?

- ☒ a. La variable **\$PATH** define los directorios donde el sistema busca comandos ejecutables.
- ☐ b. Las variables de entorno son exclusivas para scripts y no se utilizan en la terminal interactiva.
- ☐ c. Las variables de entorno no pueden ser modificadas por los usuarios.
- ☐ d. \$IFS es una variable que almacena los parámetros pasados al script.

Respuesta correcta

Pregunta **16**

Finalizado

Se puntúa 3,5 sobre 3,5

¿Cuál de las siguientes opciones representa un bucle que imprime los números del 1 al 5 en Bash?

☐ a.

```
case $i in
    1|2|3|4|5) echo $i ;;
esac
```

☐ b.

```
if [ $i -le 5 ]; then
    echo $i
fi
```

☐ c.

```
while [ $i -lt 5 ]; do
    echo $i
done
```

☒ d.

```
for i in {1..5}; do
    echo $i
done
```

Respuesta correcta

Pregunta **17**

Finalizado

Se puntúa 3,9 sobre 6,0

El siguiente [script](#) pide al usuario ingresar por teclado la ruta de un archivo. Si el archivo no existe, lo crea. En cualquier caso muestra un mensaje por pantalla con la ruta ingresada por el usuario.

Descarga el script y comprueba que funcione correctamente. En caso de encontrar errores, corrígelos, sube el archivo corregido agregando "v.2" al final del nombre y antes de la extensión .sh, y comenta cuáles fueron los errores.

chmod +x archivo.sh es el comando para poder ejecutar

 [Punto1.sh](#)

Comentario:

Evaluación y Puntaje

1. Detección de errores (50%)

- ✓ Identificaste correctamente el problema con **\$2** y lo reemplazaste por **\$1**.
- No detectaste la falta de comillas en las variables, lo que aún puede causar fallos.
- No detectaste que el script no verifica si el directorio existe antes de **touch**.

Subpuntaje: 35/50

2. Corrección del código (30%)

- ✓ Implementaste correctamente la corrección del uso incorrecto de **\$2**.
- No corregiste la falta de comillas en la función y en la llamada a **existe_archivo**.
- No agregaste la verificación de directorios antes de **touch**.

Subpuntaje: 15/30

3. Claridad en la explicación de errores y correcciones (20%)

- ✓ Explicaste correctamente el problema con **\$2**.
- ◆ No mencionaste la importancia de usar comillas ni de verificar si los directorios existen.

Subpuntaje: 15/20

Pregunta **18**

Finalizado

Se puntúa 6,4 sobre 8,0

Escribe un script que permita al usuario ingresar por teclado, de una sola vez, los datos de una persona:

<nombre> <apellido> <edad>

Luego, escribir esos datos en un archivo llamado `datos_personas.txt`, separados por punto y coma. Si el archivo no existe, se lo debe crear.

Preguntar al usuario si se quiere ingresar más datos, y repetir el procedimiento hasta que el usuario indique que no quiere seguir ingresando datos.

Finalmente, listar el archivo ordenado por apellido.

 [_Punto2.sh](#)

Comentario:

Evaluación y Puntaje

1. Cumplimiento del enunciado (70%)

- ✓ Permite ingresar los datos en una línea.
- ✓ Guarda los datos en `datos_personas.txt`, en el formato correcto.
- ✓ Repite el proceso hasta que el usuario decida salir.
- Usa `awk` innecesariamente antes de `sort`.

Puntaje: 60/70

2. Calidad técnica (30%)

- ✓ Implementa un bucle para repetición.
- ✓ Usa correctamente `;` como separador en el archivo.
- Error de sintaxis en el `if`.
- Espacios adicionales en la escritura del archivo.
- No valida correctamente la entrada de datos antes de guardarlos.

Puntaje: 20/30

Pregunta **19**

Finalizado

Se puntúa 2,8 sobre 4,0

Escribe un script llamado crear_usuarios.txt que realice las siguientes tareas:




1. Pedir por teclado el ingreso de un nuevo nombre de usuario.
2. Verificar si existe el archivo usuarios.txt, sino, crearlo.
3. Si el usuario existe en el archivo, muestre en pantalla un mensaje diciendo que existe, y terminar el script.
4. Si no existe:
 - pedir una contraseña para el usuario, que no debe ser visible mientras se ingresa.
 - controlar que la contraseña no sea vacía, sino, ingresarla nuevamente.
 - escribir en el archivo una línea con el nombre de usuario y la contraseña, separadas por punto y coma.
5. Mostrar un mensaje en pantalla luego de la creación del usuario.

 [_crear_usuarios.sh](#)

Comentario:

Evaluación y puntaje:

1. Cumplimiento del enunciado (50%)

-  Verifica y crea el archivo si no existe.
-  La verificación del usuario no es precisa.
-  El formato de escritura no cumple con el enunciado.



Subpuntaje: 30/50

2. Calidad técnica (30%)

-  Uso incorrecto de `grep` para verificar el usuario.
-  Buena lógica para validar la contraseña, aunque necesita ajustes menores.

Subpuntaje: 20/30

3. Estructura y legibilidad (20%)

-  El código es claro y fácil de seguir.
-  Mensajes podrían mejorarse para mayor claridad y formalidad.

Subpuntaje: 20/20

Pregunta **20**

Finalizado

Se puntúa 11,4 sobre 12,0

Descarga y extrae el contenido del archivo [libros.tar.gz](#). Luego, escribe un script llamado `backup.sh` que realice lo siguiente:

1. Solicite al usuario ingresar el nombre de un directorio a respaldar.
2. Verifique si el directorio ingresado existe:
 - Si no existe, muestre un mensaje de error y termine el script.
3. Cree un archivo comprimido `.tar.gz` con el contenido del directorio ingresado. El nombre del archivo debe incluir el sufijo `_backup`. Por ejemplo: `directorio_backup.tar.gz`.
4. Guarde el archivo comprimido en un directorio llamado respaldos. Si este directorio no existe, debe crearse automáticamente.

 [_Backup.sh](#)

Comentario:

Evaluación y Puntaje

1. Cumplimiento del enunciado (70%)

- ✓ Solicita al usuario el directorio.
- ✓ Verifica la existencia del directorio antes de continuar.
- ✓ Crea un archivo comprimido con el nombre correcto.
- ✓ Almacena el respaldo en `respaldos/`, creándolo si es necesario.
- ✓ Muestra mensajes adecuados de confirmación.
- ✓ No tiene errores de lógica ni de ejecución.

Puntaje: 70/70

2. Calidad técnica (30%)

- ✓ Implementa correctamente `tar`.
- ✓ Usa `mkdir -p` (después de la corrección).
- No maneja correctamente los espacios en nombres de directorios. (-5)

Puntaje: 25/30

Pregunta **21**

Sin contestar

Se puntúa como 0 sobre 14,0

Escribe un script llamado `procesar_backup.sh` que debe hacer lo siguiente:

1. Solicitar al usuario un usuario y contraseña.
 - Si el usuario no existe en el archivo `usuarios.txt`, mostrar en pantalla un mensaje, pedir que se lo cree, y terminar el script.
 - La contraseña no debe ser visible al ingresarla.
 - Si el usuario existe pero la contraseña es incorrecta, solicitarla nuevamente.
 - El script solo puede avanzar si el usuario existe y la contraseña es correcta.
2. Solicitar al usuario un nombre de archivo de backup.
 - Verificar que el archivo existe, sino pedir el nombre nuevamente.
3. Crear un directorio llamado `dir_temporal` donde se extraiga el contenido del backup.
4. Mover los archivos de texto a un subdirectorío llamado `reviews`, y mover los archivos de imagen a otro subdirectorío llamado `portaldas`.
5. Mostrar en pantalla un mensaje al completar todas las tareas.

◀ [Archivos para el primer parcial.](#)

Ir a...