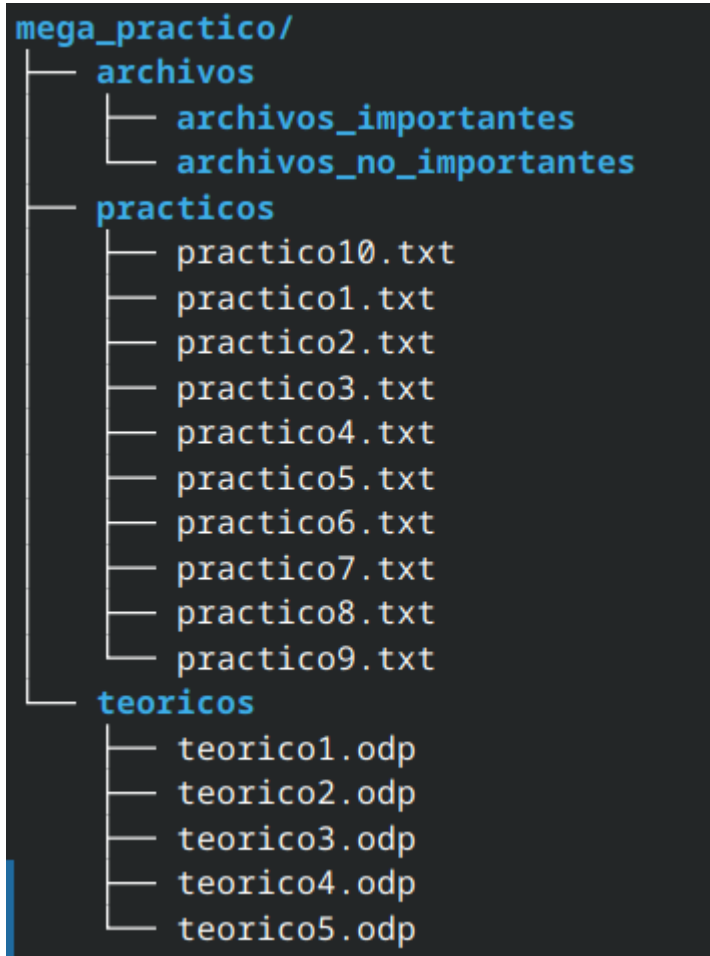


Prueba diagnóstica de ejercicios para Administración de Sistemas Operativos

1. Sabiendo que la home del usuario Don Ramón es `"/home/dramon"`:
 - a. Listar todos los archivos y directorios de la home
 - b. Listar todos los archivos y directorios pero con sus permisos y propiedades
 - c. Crear los directorios: "scripts", "archivos", "otros"
 - d. Crear en el directorio otros 1000 archivos vacíos y 1000 directorios.
 - e. Mover todos los archivos vacíos creados en el directorio "otros" al directorio "archivos"
2. Crear la siguiente estructura de directorios en la menor cantidad de comandos posibles:



- a.
 - b. ¿Con qué comando puedo ver la anterior estructura?
3. Realice un script que:
 - a. Limpie la pantalla
 - b. Muestre en pantalla el calendario actual
 - c. Espere 10 segundos
 - d. Muestre en pantalla el nombre del usuario actual, su home de trabajo y la hora actual del sistema en formato: *año-mes-día-hora:minuto:segundo*
 - e. Limpie la pantalla
 4. Realizar un script que muestre el contenido del archivo `/etc/passwd` por 10 segundos y luego pregunte si desea crear una copia en la home de trabajo actual
 5. Crear un script que permita borrar el árbol de directorios creado en el ejercicio 2, no se puede usar el comando `rm` para borrar los directorios, solamente `rmdir`, para los archivos si está permitido.

Prueba diagnóstica de ejercicios para Administración de Sistemas Operativos

6. Crear un script que reciba por parámetro 5 nombres y evalúe si alguno de ellos coincide con el nombre del usuario que está logueado actualmente.
7. Crear un script que reciba 4 nombres por parámetro y los muestre en pantalla con un salto de línea entre nombre y nombre . Los nombres tienen que demorar en aparecer en la pantalla dos segundos. Todos los nombres deben de ser mostrados en minúscula, si el usuario escribe el nombre en mayúsculas es necesario pasarlo a minúsculas.
8. Crear un script que copie los archivos passwd y hosts a un directorio en nuestra home llamado "copia_de_archivos". El script también deberá cambiar sus permisos de tal forma que el dueño tenga permisos de escritura, lectura y modificación, el grupo tenga permisos de lectura y los otros no tengan permisos.
9. Crear un script que compare un número que ingresa el usuario, si el número es menor a 12 debe de limpiar la pantalla y decir: "ingreso un número menor a 12, vuelva a intentarlo", si el número es mayor a 12 deberá de limpiar la pantalla y decir que "el número ingresado es mayor a 12". Al finalizar el script se debe de imprimir el valor que ingresó el usuario seguido de su nombre y la hora.
10. Crear un script que compruebe si el directorio ~/archivos existe, si el directorio existe dentro del deberá crear los siguientes directorios: "Java, VB.NET, SH, Python y Ruby". Dentro de cada directorio deberá de crear cuatro directorios más llamados: Pruebas, Resueltos, Ejemplos y Finales. En caso de que el directorio no exista deberá primero crearlo y luego crear el resto de los directorios.
11. Crear un script que verifique si el archivo .vimrc ubicado en la home exista, si el archivo no existe deberá de crearlo y agregar la siguiente información dentro de él:
 - a. syntax on
 - b. set tabstop
 - c. set number
 - d. set shiftwidth
 - e. set autoindent
 - f. set encoding
 - g. Si el archivo existe, solamente lo agrega al final.
12. Crear un script que tenga una variable llamada pin, la cual su valor será 1234, el usuario deberá de ingresar un número y este se deberá de comparar con el pin, si el pin es correcto deberá de listar todo el contenido de su home en forma de árbol, en caso de no coincidir con el pin deberá de decirle al usuario que el pin ingresado es incorrecto.
13. Explique con sus palabras para qué sirven los comandos dnf y yum. De ejemplos de usos en su máquina virtual o servidor que esté trabajando.
14. Se sabe que el servicio para trabajar con el demonio de mariadb es mariadb.service. Explique con sus palabras con qué comandos puede:
 - a. Detener el servicio
 - b. Reiniciar el servicio
 - c. Iniciar el servicio
 - d. Indicar que arranque con el sistema siempre
15. Dado un número ingresado por el usuario, utilizar la estructura for para realizar un contador que comience en el número 1 hasta el número que ingresó el usuario. Ejemplo: Si el usuario ingresa el número 5, el resultado sería: 1, 2, 3, 4, 5

Prueba diagnóstica de ejercicios para Administración de Sistemas Operativos

16. Al ejercicio anterior agregarle una condición para que verifique que el número ingresado por el usuario sea menor a 10. Si el número ingresado por el usuario es mayor a 10 deberá detener el programa y avisarle al usuario.
17. Crear un script que reciba por parámetro posicional un número y guarde en un archivo una lista comenzando por ese número hasta el 50, si el número ingresado es mayor a 50 deberá de mostrar un mensaje de error.
18. No son un ejercicio, soy solamente un mensaje de aliento para los próximos ejercicios.
19. Crear un script que muestra la tabla de multiplicar de un número ingresado por el usuario.
20. Al ejercicio anterior modificarlo con un menú para que el usuario pueda elegir las tablas del 1 al 9.
21. Crear un script que contenga el siguiente menú y sus funcionalidades.
 - a. Crear un directorio en la home del usuario
 - b. Crear un archivo en la home usuario
 - c. Saludar al usuario de manera amistosa
 - d. Buscar información en un archivo de la home
 - e. Buscar un archivo en la home del usuario
 - f. Generar una lista de todos los archivos y directorios de la home del usuario
 - g. Ver la lista anterior (se debe de verificar que existe)
22. Explique con sus palabras el comando tar y de un ejemplo de uso.
23. Dada la siguiente ruta: /home/usuario/archivos/:
 - a. Empaquetar el directorio archivos
 - b. Ver el contenido del empaquetado
 - c. Desempaquetar el directorio archivos
 - d. Empaquetar y comprimir el directorio archivos
 - e. Ver el contenido del archivo empaquetado y comprimido
 - f. Desempaquetar y descomprimir el archivo.
24. Buscar todos los archivos que sean imágenes .jpg de nuestro equipo.
25. Encontrar todos los audios .mp3 que haya en el directorio personal del usuario o en sus subdirectorios.
26. Buscar todos los directorios que comiencen con la letra "M" mayúscula, y que estén en los tres primeros niveles de directorios.
27. Encontrar los archivos comprimidos .tar que existan en los tres primeros niveles de directorios.
28. Buscar todos los archivos y directorios que comienzan con la palabra "pas" pero que no terminan en "swd".
29. Encontrar los archivos y directorios que contengan el String "conf" que estén bajo /etc o bajo /var.
30. Buscar todos los archivos que tengan hasta 5 MiB de tamaño.
31. Buscar todos los archivos que tengan hasta 5 MiB de tamaño, pero que hayan sido creados en la última semana.
32. Buscar todos los archivos ejecutables que estén bajo /bin y listarlos mostrando todas sus propiedades.
33. Buscar todos los archivos ejecutables que estén bajo /bin y listarlos en una sola columna ordenada alfabéticamente inversa.
34. Encontrar los archivos ocultos que hay en nuestro sistema.

Prueba diagnóstica de ejercicios para Administración de Sistemas Operativos

35. Contar cuántos archivos ocultos hay en nuestro sistema. Solo debe aparecer el número, nada más.
36. Encontrar los archivos .cpp que tengamos en nuestro directorio personal o sus subdirectorios.
37. A partir del directorio actual (cualquiera que sea) buscar los archivos editables.
38. Encontrar un directorio que se llama Imágenes, imágenes, imagenes o Imagenes, usando la menor cantidad de caracteres para este comando.
39. Buscar los dispositivos que son discos de estado sólido de nuestro equipo.
40. Buscar si existen archivos de tamaño mayor que 100 MiB en el directorio /usr.
41. Buscar todos los directorios que contengan la palabra "sys" dentro de nuestro sistema, sin importar si es al principio, al medio o al final.
42. Buscar todos los archivos que fueron modificados hace 30 minutos en nuestro sistema operativo
43. Buscar todos los archivos que se crearon hace 20 días en nuestra computadora
44. Buscar todos los archivos y carpetas dentro de nuestra home de trabajo que no poseen contenido.
45. Crear un script que dado el nombre de un archivo que nos ingresa el usuario realice las siguientes operaciones:
 - a. Buscar una palabra clave usando el filtro grep en el archivo y mostrar las coincidencias.
 - b. Buscar cuantas veces se repite una palabra clave en el archivo y mostrar las coincidencias.
 - c. Buscar todas las líneas que no son la palabra clave que ingresó el usuario y mostrarlas en pantalla.
 - d. Buscar en qué líneas se encuentra la palabra clave y mostrarlas en pantalla.
46. Crear un script que muestre todas las líneas de un archivo ingresado por el usuario que comienzan por la letra a
47. Crear un script que me permita mostrar todas las líneas que no son comentarios dado un archivo que nos ingresa el usuario por parámetro posicional.
48. Crear un script que nos muestre si nuestro usuario está dentro del archivo sudoers, es necesario verificar si el archivo existe antes de mostrar los datos del usuario.
49. Crear un script que tome el archivo .bashrc y nos muestre cuantas veces se usó la estructura IF
50. Al anterior script modificarlo para que sea reusable para cualquier archivo que nos pasa el usuario.
51. Se sabe que en la raíz, dentro del directorio /etc, existe un archivo llamado passwd, verificar si este archivo realmente existe. Si el archivo existe verificar cuantas veces figura nuestro nombre de usuario dentro de él. En caso de que no exista deberá de mostrar un mensaje de que no se encuentra el archivo.
52. Crear un script que permita agregar usuarios al sistema, el script deberá:
 - a. Chequear que el usuario no exista
 - b. Chequear que el grupo al que se lo va a agregar exista
 - c. Pasar el nombre que escriba el usuario a minúscula
53. Al script anterior agregarle las opciones para buscar si un usuario existe y si existe eliminarlo, agregar un menú en caso de no tenerlo.
54. Crear un script que dado un archivo que contiene números desordenados que nos pasa el usuario, nos muestre la salida ordenada de menor a mayor.

Prueba diagnóstica de ejercicios para Administración de Sistemas Operativos

55. Crear un script que dado un número ingresado por el usuario cuente los números de uno en uno hasta llegar a 50 y los muestre en pantalla. Si el usuario ingresa un número mayor a 50 se le avisará que ingrese un número menor y se detendrá el programa.
56. Realizar la programación de las siguientes tareas en un archivo de texto plano, la tarea que se ejecutará será "miScript.sh"
 - a. • Programar un trabajo para ejecutarse en el minuto 40 de cada hora de cada día.
 - b. Programar un trabajo para ejecutarse cada día a las 21:35hs.
 - c. Programar un trabajo para ejecutarse de lunes a viernes a las 22:31hs.
 - d. Programar un trabajo para ejecutarse los martes y los jueves a las 21:32hs
 - e. Programar un trabajo para ejecutarse los días 10 y 20 de todos los meses a las 10:00hs.
 - f. Programar un trabajo para ejecutarse cada 15 minutos.
 - g. Programar un trabajo para ejecutarse cada día a las 00:00hs
 - h. Programar un trabajo para ejecutarse cada primer día de mes a las 01:00hs
 - i. Programar un trabajo para ejecutarse durante los días laborables a las 20:00hs
 - j. Ejecutar un trabajo el primer viernes de cada mes a las 00:00hs
57. Explique con sus palabras cómo sincronizar un directorio de su home principal a un servidor remoto.
58. Explique con sus palabras cómo agregar grupos al sistema.