Breve explicación de que hace cada función implementada:

pedirRutaSiNecesario(): Esta es una función que creamos para evitar la repetición de un mismo código, dentro tiene un bucle while que verifica si la variable ruta es un directorio válido, en caso de que no lo sea, pedirá que se ingrese nuevamente.

opcion1():

1) Llama a pedirRutaSiNecesaro
2) Encuentra cuántos archivos hay dentro del directorio ruta a cualquier nivel de profundidad y los cuenta.
3) Vuelve a contar cuántos archivos hay dentro del directorio "ruta" pero solo dentro de esa carpeta, sin contar directorios hijos.
4) Calcula cuántos archivos están en subcarpetas haciendo la resta.
5) Para encontrar los archivos más pesados y menos pesados dentro de la ruta:
    - Usa un find que busca archivos, formatea la salida poniendo primero el tamaño en bytes del archivo y luego la ruta del archivo
    - Todo lo que encuentra el find se ordena de forma descendiente (viendo el tamaño en bytes).
    - Se toma el archivo que queda arriba del todo como el más grande y el que queda al final del todo como el más chico.
    - Estas cadenas se "limpian" para obtener solo el nombre del archivo sin toda la ruta.

opcion2():

1. Llama a pedirRutaSiNecesario,
2. Usa un find para obtener todos los archivos que están en el primer nivel de profundidad de la ruta indicada y guarda sus rutas en una variable local llamada archivos
3. Recorre los archivos y renombra cada uno añadiendo bck al final de su ruta.

opcion3():

1. Ejecuta df . -h para encontrar la información del disco que se está usando en formato legible
2. Usa tail -n 1 con lo que se devuelve para que solo se tome la última línea
3. Se toma la información del espacio usado del disco y se guarda en una variable
4. Se hace lo mismo para encontrar el espacio disponible en el disco,
5. Ambos valores se muestran por la terminal.
6. Busca el archivo mas pesado en todo el sistema:
    - Para encontrar el archivo más pesado se hace usa un find de todos los archivos que hay en la pc (desde /)

- Los formatea para que primero aparezca el tamaño en bytes y luego aparezca la ruta
- Se ordenan en orden descendente y se toma el primero guardandolo en una variable.
- Luego separamos en 2 variables separadas, el peso del archivo y la ruta
- Se Pasa el peso a formato iec
- Muestra por terminal el archivo más pesado con su ruta y peso formateado.

opcion4(): Esta función primero llama a pedirRutaSiNecesario para establecer la variable ruta, luego pide al usuario que ingrese una palabra para buscar sus apariciones en los archivos dentro de la ruta especificada, encuentra estas apariciones haciendo uso de grep y luego las muestra por la terminal

opcion5(): Esta función muestra por terminal el nombre del usuario usando la variable de entorno USER, luego te muestra con el comando uptime -s, la fecha y hora de encendido del dispositivo. Por último, con el comando date, te muestra la fecha de hoy, usando %A, %d y %B para darle formato.

opcion6(): Esta función pide al usuario la url de una web, luego la guarda en una variable local, luego llama a la función PedirRutaSiNecesario para establecer la ruta donde se guardará la url que pasó el usuario, luego escribe dentro de un archivo “archivoweb.txt” dentro del directorio que especificó el usuario la url.

opcion7(): esta función lo único que hace es pedirle una ruta al usuario y validar si es una ruta válida, si la ruta no es válida se la vuelve a pedir hasta que ingrese una que sí lo sea.

Comandos clave utilizaods:

find -type f
wc -l
find -maxdepth 1
find -type f printf “%s %p\n”
sort -nr
head -n 1
cut -d’ ‘ -f2-
xargs -r
basename
echo
mv
tail -n 1
awk ‘{print $3}’
df -h .

sudo
numfmt –to=iec
read -p
grep -rnw “$ruta” -e $palabra
uptime -s
date + “Hoy es: %A %d %B