



SERVICIOS SYSTEMD / SYSTEM-SHUTDOWN:



OTROS DOCUMENTOS DE LINUX

[SERVICIOS:MONTAR/DESMONTAR UDS](#)[BASH_COMANDS](#)[BASH COMANDS - Extended](#)

INDICE DOCUMENTO

[SERVICIO DE INICIO - I](#)[Script .sh](#)[Permisos](#)[Script SYSTEMD](#)[Habilitar Serv.](#)
[Reiniciar Serv.](#)[SERVICIO DE APAGADO - I](#)[Script .sh](#)[Permisos](#)[Script SHUTDOWN](#)[Habilitar Serv.](#)
[Reiniciar Serv.](#)[SERVICIO DE INICIO - II](#)[Servicio Sin Archivo .sh](#)[SERVICIO DE APAGADO - II](#)[Servicio Sin Archivo .sh](#)

CONCEPTOS UTILES



CREAR UN SERVICIO



SERVICIO SYSTEMD

- CREAR SCRIPT
- DARLE PERMISOS EJECUCION
- CREAR ARCHIVO DE SERVICIO SYSTEMD
- HABILITAR EL SERVICIO AL INICIO
- PROBAR EL SCRIPT SIN REINICIAR
- REVISAR LOGS DEL SERVICIO
- RECARGAR Y REINICIAR EL SERVICIO
- SERVICIO SYSTEMD SIN SCRIPT .SH



SERVICIO SYSTEMD - SHUTDOWN

- CREAR SCRIPT
- DARLE PERMISOS EJECUCION
- CREAR ARCHIVO DE SERVICIO SHUTDOWN
- HABILITAR EL SERVICIO
- PROBAR EL SCRIPT SIN APAGAR
- REVISAR LOGS DEL SERVICIO
- RECARGAR Y REINICIAR EL SERVICIO
- SERVICIO SYSTEMD-SHUTDOWN SIN SCRIPT .SH





SERVICIOS SYSTEMD DE INICIO DE SISTEMA

SERVICIO SYSTEMD CON SCRIPT .SH

CREAR SCRIPT .SH

Creamos un script de bash en usr/bin que contenga las órdenes que queremos que se ejecuten al arrancar el sistema

Ruta **/usr/local/bin/**

Tipo Guardarlo como **.sh**

```
vim /usr/local/bin/file-name.sh
```

```
#!/bin/bash
sudo apt update
sudo apt upgrade
exit
```

DARLE PERMISOS EJECUCION

Le damos permisos de ejecución:

```
chmod +x file-name.sh
```

A veces tenemos que cambiar también el propietario, porque lo crea como 'root' y hay que pasarlo al usuario.

El 1º lo añade al usuario y e 2º al grupo, en mi caso: **gerar_kde:gerar_kde**

```
Chown user-name:group-name file-name.sh
```

podemos probarlo para ver que vaya bien

```
./file-name.sh
```

SCRIPT DE SERVICIO SYSTEMD

Creamos tambien un script de servicio en /etc/systemd/system/ que contenga la orden de ejecución del archivo sh

Ruta **/etc/systemd/system/**

Tipo Guardarlo como **service-name.service**

```
sudo vim /etc/systemd/system/service-name.service
```

```
[Unit]
Description=Montar unidad remota al inicio
```





```
After=network-online.target
Wants=network-online.target

[Service]
Type=simple
User=user-name
ExecStart=/usr/local/bin/file-name.sh
Restart=on-failure

[Install]
WantedBy=default.target
```



HABILITAR EL SERVICIO

Habilitamos el Servicio para que se inicie al arrancar:

```
sudo systemctl daemon-reexec
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable service-name.service
```



PROBAR SCRIPT SIN REINICIAR

Probamos si todo funciona correctamente

```
sudo systemctl start service-name.service
```



REVISAR EL PROCESO

Después de ejecutar el servicio podemos revisar si falla algo como permisos, rutas, o errores de rclone, con journalctl :

```
journalctl -u service-name.service --no-pager
```

Podemos crear un Archivo log en el que revisar como ha ido el proceso, o ver las causas si falla y que se sobrescriba cada vez que se ejecute el servicio.



CREAR ARCHIVO DE LOG

Podemos crear un Archivo log en el que revisar como ha ido el proceso, o ver las causas si falla.

Ruta **/var/log/**

Tipo Guardarlo como **log-name.log**

```
sudo touch /var/log/log-name.log
```

Asegúrate de que el archivo de log se pueda escribir, para lo que cambiamos el dueño y el grupo, porque lo crea como 'root'





El 1º lo añade al usuario y e 2º al grupo, en mi caso:

gerar_kde:gerar_kde

```
chown user-name:group-name log-name.log
```



RECARGAR Y REINICIAR SERVICIO

Habilitamos el Servicio para que se inicie al arrancar:

```
sudo systemctl daemon-reexec
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable service-name.service
```



CAMBIOS POSTERIORES

Si queremos hacer cambios en el servicio, modificaremos el script .sh y sobre el servicio no necesitas modificar nada más, únicamente volver a reiniciarlo

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable service-name.service
```

SERVICIO SYSTEMD SIN SCRIPT .SH

El proceso es el mismo, lo único es que no creamos un script de bash en un archivo y luego lo ejecutamos desde un archivo .service, si no que directamente creamos el .service.



CREAR SCRIPT DE SERVICIO SYSTEMD

El proceso es el mismo, lo único es que no creamos un script de bash en un archivo y luego lo ejecutamos desde un archivo .service, si no que directamente creamos el .service.

En este caso, no es necesario que el script tenga la extensión

En Linux, los scripts que se ejecutan automáticamente por el sistema son identificados por su nombre y permisos, no por la extensión. Scripts de

/etc/systemd/system/

/lib/systemd/system-shutdown/

Lo importante es que:

- El archivo sea ejecutable (chmod +x).
- Tenga una shebang válida (#!/bin/bash al principio).
- Esté en el directorio correcto como (/etc/systemd/system/) o (/lib/systemd/system-shutdown/)

Creamos también un script de servicio en /etc/systemd/system/ que contenga la orden a ejecutar

Ruta **/etc/systemd/system/**

Tipo Guardarlo como **file-name** o como **file-name.sh**





```
sudo vim /etc/systemd/system/file-name
```

```
#!/bin/bash
/etc/systemd/system/file-name
sudo apt update
sudo apt upgrade
exit
```



DARLE PERMISOS EJECUCION

Le damos permisos de ejecución:

```
chmod +x file-name
```

podemos probarlo para ver que vaya bien

```
./file-name
```



HABILITAR EL SERVICIO

Habilitamos el Servicio para que se inicie al arrancar:

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable file-name
```



PROBAR SCRIPT SIN REINICIAR

Probamos si todo funciona correctamente

```
sudo systemctl start file-name
```



RECARGAR Y REINICIAR SERVICIO

Habilitamos el Servicio para que se inicie al arrancar:

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable file-name
```





SERVICIOS SHUTDOWN DE APAGADO DE SISTEMA

SERVICIO SHUTDOWN

El proceso es exactamente el mismo, pero cambia la ruta del servicio. El mayor cambio se produce a nivel de logs.

Como en los servicios de inicio podemos hacerlo de 2 maneras:

Opción 1	file-name.sh	/usr/bin/
	+	
	file-name.service	/lib/systemd/system-shutdown/
Opción 2	[file-name] o [file-name.sh] /lib/systemd/system-shutdown/	



PROCESO



CREAR SCRIPT



DARLE PERMISOS DE EJECUCION



CREAR ARCHIVO DE SERVICIO SHUTDOWN



HABILITAR EL SERVICIO



CAMBIOS A NIVEL DE LOGS

El mayor cambio se produce a nivel de logs.

Durante el apagado, muchos sistemas de archivos ya pueden estar desmontados cuando se ejecutan los scripts en /lib/systemd/system-shutdown.

Por tanto, escribir en /var/log/ no siempre es seguro ni efectivo en ese momento.

Usa un archivo de log en /run/ (que está en tmpfs, en memoria):

Ruta **/run/**

Tipo Guardarlo como **file-name.log**

```
Sudo touch /run/log-name.log
```

Asegúrate de que el archivo de log se pueda escribir

```
chown user-name:group-name log-name.log
```



EL SCRIPT

Script para desmontar una Unidad

Ruta **/lib/systemd/system-shutdown/**

Tipo Guardado sin extensión, como Script **file-name**

```
#!/bin/bash
```





```
#!/lib/systemd/system-shutdown/desmontar_datos
DATOS_MOUNT="/media/DATOS"
LOG_FILE="/run/desmontar_datos.log"
echo "[Shutdown] Intentando desmontar unidad D..." >> "$LOG_FILE"
desmontar_si() {
    MOUNT_PATH="$1"
    if mount | grep -q "$MOUNT_PATH"; then
        umount -l "$MOUNT_PATH" && \
            echo "Desmontado correctamente: $MOUNT_PATH" >> "$LOG_FILE" || \
            echo "Fallo al desmontar: $MOUNT_PATH" >> "$LOG_FILE"
    else
        echo "No estaba montado: $MOUNT_PATH" >> "$LOG_FILE"
    fi
}
desmontar_si "$DATOS_MOUNT"
exit 0
```



MEJORAS DEL SCRIPT



LOG SE GUARDE EN DISCO ANTES DEL APAGADO

Para que ese log se guarde en disco antes del apagado para conservarlo tras reiniciar, necesitas mover el archivo de log desde la RAM (/run/) a un disco persistente antes de que el sistema se apague del todo.

Lo más seguro es escribir el log en un archivo dentro de /root/ o / directamente (porque el sistema de archivos raíz permanece disponible hasta el final del apagado).

Añadimos al Script

```
#!/bin/bash
#!/lib/systemd/system-shutdown/desmontar_datos
DATOS_MOUNT="/media/DATOS"
PERSISTENT_LOG="/root/desmontar_datos.log"
echo "[Shutdown] $(date '+%Y-%m-%d %H:%M:%S') - Intentando desmontar unidad D..." >> "$PERSISTENT_LOG"
desmontar_si() {
    MOUNT_PATH="$1"
    if mount | grep -q "$MOUNT_PATH"; then
        umount -l "$MOUNT_PATH" && \
            echo "Desmontado correctamente: $MOUNT_PATH" >> "$LOG_FILE" || \
            echo "Fallo al desmontar: $MOUNT_PATH" >> "$LOG_FILE"
    else
        echo "No estaba montado: $MOUNT_PATH" >> "$LOG_FILE"
    fi
}
desmontar_si "$DATOS_MOUNT"
exit 0
```



LIMITAR TAMANO ARCHIVO LOG

Si te preocupa el crecimiento del archivo, puedes limitarlo manualmente agregando al principio del script:



```
if [ -f "$PERSISTENT_LOG" ] && [ "$(wc -l < "$PERSISTENT_LOG")" -gt 500 ];
then
    tail -n 200 "$PERSISTENT_LOG" > "$PERSISTENT_LOG.tmp" && mv
"$PERSISTENT_LOG.tmp" "$PERSISTENT_LOG"
fi
```



COPIAR LOG, P.EJ. A CARPETA USUARIO

Para que se copie el log a una ruta como /home/gerar_kde/.logs/desmontar_datos.log.

Creamos la carpeta .logs en la carpeta del usuario

Ruta **/home/gerar_kde/.logs**

```
Sudo mkdir /home/gerar_kde/.logs
```

Asegúrate de que el archivo de log se pueda escribir

```
chown user-name:group-name /home/gerar_kde/.logs
```

Y añadimos al Script

```
#!/bin/bash
# /lib/systemd/system-shutdown/desmontar_datos

DATOS_MOUNT="/media/DATOS"
PERSISTENT_LOG="/root/desmontar_datos.log"
DEST_DIR="/home/gerar_kde/.logs"
DEST_LOG="$DEST_DIR/desmontar_datos.log"

# Limitar tamaño del log en /root
if [ -f "$PERSISTENT_LOG" ] && [ "$(wc -l < "$PERSISTENT_LOG")" -gt 500 ];
then
    tail -n 200 "$PERSISTENT_LOG" > "${PERSISTENT_LOG}.tmp" && mv "$
{PERSISTENT_LOG}.tmp" "$PERSISTENT_LOG"
fi

# Registro inicial
echo "[Shutdown] $(date '+%Y-%m-%d %H:%M:%S') - Intentando desmontar unidad D..." >> "$PERSISTENT_LOG"

# Función para copiar el log si existe el directorio destino
copiar_log_a_home() {
    if [ -d "$DEST_DIR" ]; then
        cp "$PERSISTENT_LOG" "$DEST_LOG"
        echo "Log copiado a $DEST_LOG" >> "$PERSISTENT_LOG"
    else
        echo "Directorio de destino no encontrado: $DEST_DIR" >>
"$PERSISTENT_LOG"
    fi
}

# Función de desmontaje
desmontar_si() {
    MOUNT_PATH="$1"
    if mount | grep -q "$MOUNT_PATH"; then
        copiar_log_a_home
        umount -l "$MOUNT_PATH" && \
            echo "Desmontado correctamente: $MOUNT_PATH" >> "$PERSISTENT_LOG" || \
            echo "Fallo al desmontar: $MOUNT_PATH" >> "$PERSISTENT_LOG"
    else
        echo "No estaba montado: $MOUNT_PATH" >> "$PERSISTENT_LOG"
    fi
}
```





```

    fi
}
desmontar_si "$DATOS_MOUNT"
exit 0

```

El log se escribe en /root/desmontar_datos.log (como copia confiable). Limitando el tamaño tal que: Si el archivo existe y tiene más de 500 líneas. Conservar solo las últimas 200 líneas.

Luego, se asegura de que /home/gerar_kde/.logs exista antes de copiar, si existe lo copia y si no existe, el mensaje correspondiente se agrega al log en /root/.



RENOMBRAR POR FECHA LA COPIA DEL LOG

Para que también el log en /home/gerar_kde/.logs/ se renombre por fecha cada vez, por ejemplo: desmontar_datos_2025-05-13.log

añadimos al Script

```

#!/bin/bash
# /lib/systemd/system-shutdown/desmontar_datos

DATOS_MOUNT="/media/DATOS"
PERSISTENT_LOG="/root/desmontar_datos.log"
DEST_DIR="/home/gerar_kde/.logs"
DATE_SUFFIX="$(date '+%Y-%m-%d')"
DEST_LOG="$DEST_DIR/desmontar_datos_${DATE_SUFFIX}.log"

# Limitar tamaño del log en /root
if [ -f "$PERSISTENT_LOG" ] && [ "$(wc -l < "$PERSISTENT_LOG")" -gt 500 ]; then
    tail -n 200 "$PERSISTENT_LOG" > "${PERSISTENT_LOG}.tmp" && mv "${PERSISTENT_LOG}.tmp" "$PERSISTENT_LOG"
fi

# Registro inicial
echo "[Shutdown] $(date '+%Y-%m-%d %H:%M:%S') - Intentando desmontar unidad D..." >> "$PERSISTENT_LOG"

# Función para copiar el log a /home/gerar_kde/.logs con nombre por fecha
copiar_log_a_home() {
    if [ -d "$DEST_DIR" ]; then
        cp "$PERSISTENT_LOG" "$DEST_LOG"
        echo "Log copiado a $DEST_LOG" >> "$PERSISTENT_LOG"
    else
        echo "Directorio de destino no encontrado: $DEST_DIR" >>
"$PERSISTENT_LOG"
    fi
}

# Función de desmontaje
desmontar_si() {
    MOUNT_PATH="$1"
    if mount | grep -q "$MOUNT_PATH"; then
        copiar_log_a_home
        umount -l "$MOUNT_PATH" && \
            echo "Desmontado correctamente: $MOUNT_PATH" >> "$PERSISTENT_LOG" || \
            echo "Fallo al desmontar: $MOUNT_PATH" >> "$PERSISTENT_LOG"
    else
        echo "No estaba montado: $MOUNT_PATH" >> "$PERSISTENT_LOG"
    fi
}

desmontar_si "$DATOS_MOUNT"
exit 0

```

El log principal Permanece en /var/log/montar_drive.log como archivo principal, limitado a cierto número de líneas.





Se copia a /home/gerar_kde/.logs/ con un nombre por fecha, para mantener un historial diario.



NUEVO SCRIPT

Script para montar Google Drive con rClone

Ruta **/usr/local/bin/**

Tipo **.sh montar_drive.sh**

Log **/var/log/montar_drive.log**

```
#!/bin/bash
# /usr/local/bin/montar_drive.sh

MOUNT_POINT="/home/gerar_kde/One_Drive"
LOG_FILE="/var/log/montar_onedrive.log"
TIMESTAMP="$(date '+%Y-%m-%d %H:%M:%S')"
```

Limitar tamaño del log principal

```
if [ -f "$LOG_FILE" ] && [ "$(wc -l < "$LOG_FILE")" -gt 500 ]; then
    tail -n 200 "$LOG_FILE" > "${LOG_FILE}.tmp" && mv "${LOG_FILE}.tmp"
"$LOG_FILE"
fi
```

Redirigir salida estándar y de errores al log principal

```
exec > "$LOG_FILE" 2>&1
```

Verifica que el punto de montaje exista

```
if [ ! -d "$MOUNT_POINT" ]; then
echo "Error: el directorio $MOUNT_POINT no existe."
exit 1
fi
```

Verifica si hay conectividad antes de montar

```
if ! ping -c 1 -W 2 www.googleapis.com &> /dev/null; then
    echo "No hay conexión a internet. Reintentará más tarde..."
    exit 2
fi
```

Ejecuta rclone en modo foreground (sin --daemon, lo controla systemd)

```
exec /usr/bin/rclone mount \
    gdrive: "$MOUNT_POINT" \
    --vfs-cache-mode writes \
    --allow-other \
    --dir-cache-time 1h \
    --poll-interval 1m \
    --timeout 1m \
    --umask 002 \
    --log-level INFO \
    --log-file "$LOG_FILE"
```



MEJORAS DEL SCRIPT



LIMITAR LOG PRINCIPAL Y COPIARLO CON UN NOMBRE POR FECHA

El log principal Permanece en /var/log/montar_drive.log como archivo principal, limitado a cierto número de líneas.

Además Se copia a /home/gerar_kde/.logs/ con un nombre por fecha, para mantener un historial diario.



Cambiamos esta parte que es la que manejaba la salida log

```
LOG_FILE="/var/log/montar_drive.log"
exec > "$LOG_FILE" 2>&1
```

Lo cambiamos y Ponemos este bloque al inicio del script, justo antes de que empiece a escribir logs o realizar acciones importantes.

```
LOG_FILE="/var/log/montar_drive.log"
DATE_SUFFIX="$(date '+%Y-%m-%d')"
DEST_DIR="/home/gerar_kde/.logs"
DEST_LOG="$DEST_DIR/montar_drive_${DATE_SUFFIX}.log"
# Limitar tamaño del log principal
if [ -f "$LOG_FILE" ] && [ "$(wc -l < "$LOG_FILE")" -gt 500 ]; then
    tail -n 200 "$LOG_FILE" > "${LOG_FILE}.tmp" && mv "${LOG_FILE}.tmp"
"$LOG_FILE"
fi

# Este exec reemplazará el script por rclone solo si llega hasta aquí
exec /usr/bin/rclone mount \
    gdrive: "$MOUNT_POINT" \
    --vfs-cache-mode writes \
    --allow-other \
    --dir-cache-time 1h \
    --poll-interval 1m \
    --timeout 1m \
    --umask 002 \
    --log-level INFO \
    --log-file "$LOG_FILE"

# Si exec falla, se ejecutará esta parte (lo cual no debería pasar normalmente)
# Copia el log al directorio del usuario (solo si no fue reemplazado por exec)
if [ -d "$DEST_DIR" ]; then
    cp "$LOG_FILE" "$DEST_LOG"
    echo "[Info] Log copiado a $DEST_LOG" >> "$LOG_FILE"
else
    echo "[Warning] No se pudo copiar el log a $DEST_DIR (no existe)" >>
"$LOG_FILE"
fi
exit 0
```

```
#!/bin/bash
#/usr/local/bin/montar_drive.sh
```





```

LOG_FILE="/var/log/montar_drive.log"
DATE_SUFFIX="$(date '+%Y-%m-%d')"
DEST_DIR="/home/gerar_kde/.logs"
DEST_LOG="$DEST_DIR/montar_drive_${DATE_SUFFIX}.log"

# Limitar tamaño del log principal
if [ -f "$LOG_FILE" ] && [ "$(wc -l < "$LOG_FILE")" -gt 500 ]; then
    tail -n 200 "$LOG_FILE" > "${LOG_FILE}.tmp" && mv "${LOG_FILE}.tmp"
"$LOG_FILE"
fi

# Registro inicial
echo "[${TIMESTAMP}] Inicio del montaje de Google Drive" >> "$LOG_FILE"

# Verifica si ya está montado
if mountpoint -q "$MOUNT_POINT"; then
    echo "Google Drive ya está montado en $MOUNT_POINT" >> "$LOG_FILE"
    exit 0
fi

# Verifica que el punto de montaje exista
if [ ! -d "$MOUNT_POINT" ]; then
    echo "Error: el directorio $MOUNT_POINT no existe." >> "$LOG_FILE"
    exit 1
fi

# Verifica si hay conectividad antes de montar
if ! ping -c 1 -W 2 www.googleapis.com && /dev/null; then
    echo "No hay conexión a internet. Reintentará más tarde..." >> "$LOG_FILE"
    exit 2
fi

# Montar con rclone
echo "[$(date '+%Y-%m-%d %H:%M:%S')] Ejecutando rclone mount..." >>
"$LOG_FILE"

# Este exec reemplazará el script por rclone solo si llega hasta aquí
exec /usr/bin/rclone mount \
    gdrive: "$MOUNT_POINT" \
    --vfs-cache-mode writes \
    --allow-other \
    --dir-cache-time 1h \
    --poll-interval 1m \
    --timeout 1m \
    --umask 002 \
    --log-level INFO \
    --log-file "$LOG_FILE"

# Si exec falla, se ejecutará esta parte (lo cual no debería pasar normalmente)
# Copia el log al directorio del usuario (solo si no fue reemplazado por exec)
if [ -d "$DEST_DIR" ]; then
    cp "$LOG_FILE" "$DEST_LOG"

```





```
    echo "[Info] Log copiado a $DEST_LOG" >> "$LOG_FILE"
else
    echo "[Warning] No se pudo copiar el log a $DEST_DIR (no existe)" >>
"$LOG_FILE"
fi
exit 0
```

