



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

## Práctica 3 - Asegurar Bootloader

ASIGNATURA

Administración de Sistemas Unix/Linux

ALUMNO

Alexis de Jesus Arizmendi López - 318176110

PROFESOR

Yeudiel Hernández Torres

AYUDANTES

Virgilio Castro Rendón  
Raúl Ríos Ciriaco

## Shell de root desde el bootloader

1. Al presionar la tecla **e** en el inicio de Debian, accedes al menú de edición de las entradas del GRUB (GRand Unified Bootloader). Este menú te permite modificar temporalmente los parámetros de arranque de Debian, como las opciones del kernel o las rutas de los dispositivos. Los cambios que realices en este menú no se guardan de forma permanente, solo se aplican para esa sesión de arranque.

The screenshot shows the GRUB configuration editor. The top status bar reads "GNU GRUB version 2.06-13+deb12u1". The main area contains a shell script-like configuration:

```
setparams 'Debian GNU/Linux'

load_video
insmod gzio
if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; \
fi
insmod part_msdos
insmod ext2
set root='hd0,msdos1'
if [ x$feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1\
--hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 08b30755-6b13-422d\
-ae64-fa3966f5adce
else
    search --no-floppy --fs-uuid --set=root 08b30755-6b13-422d-ae6\

Minimum Emacs-like screen editing is supported. TAB lists
completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a
command-line or ESC to discard edits and return to the GRUB
menu.
```

2. **init=/bin/sh**

Esta es una opción del kernel que se utiliza para especificar qué programa debe ejecutarse como proceso de inicialización (PID 1) durante el arranque. En este caso, se le está diciendo al kernel que, en lugar de iniciar el sistema normalmente, cargue una shell `/bin/sh`. Este método se utiliza frecuentemente para entrar en un modo de recuperación o para realizar tareas de mantenimiento, ya que carga un entorno mínimo con acceso a la shell.

The screenshot shows the GRUB configuration editor. The top status bar reads "GNU GRUB version 2.06-13+deb12u1". The main area contains a shell script-like configuration:

```
set root='hd0,msdos1'
if [ x$feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1\
--hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 08b30755-6b13-422d\
-ae64-fa3966f5adce
else
    search --no-floppy --fs-uuid --set=root 08b30755-6b13-422d-ae6\
4-fa3966f5adce
fi
echo      'Loading Linux 6.1.0-28-amd64 ...'
linux    /boot/vmlinuz-6.1.0-28-amd64 root=UUID=08b30755-6b1\
3-422d-ae64-fa3966f5adce ro init=/bin/sh quiet_
echo      'Loading initial ramdisk ...'
initrd   /boot/initrd.img-6.1.0-28-amd64

Minimum Emacs-like screen editing is supported. TAB lists
completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a
command-line or ESC to discard edits and return to the GRUB
menu.
```

---

3. Presionamos **ctrl x**

4. **mount**

El comando **mount** se utiliza para montar sistemas de archivos en Linux. Esto significa que se asocia un sistema de archivos con un punto de montaje en el árbol de directorios, lo que permite acceder a su contenido. Sin argumentos, **mount** muestra una lista de todos los sistemas de archivos montados en ese momento.

```
[ 0.251713] [Firmware Bug]: cpu 0, try to use APIC520 (LVT offset 2) for vector or 0x14, but the register is already in use for vector 0x8 on this cpu
[ 0.016148] [Firmware Bug]: cpu 1, try to use APIC520 (LVT offset 2) for vector or 0x14, but the register is already in use for vector 0x8 on this cpu
[ 0.016148] [Firmware Bug]: cpu 2, try to use APIC520 (LVT offset 2) for vector or 0x14, but the register is already in use for vector 0x8 on this cpu
[ 0.016148] [Firmware Bug]: cpu 3, try to use APIC520 (LVT offset 2) for vector or 0x14, but the register is already in use for vector 0x8 on this cpu
[ 1.430213] piix4_smbus 0000:00:07.3: SMBus Host Controller not enabled!
/dev/sdal: clean, 36139/655360 files, 423400/2612224 blocks
/bin/sh: 0: can't access tty: job control turned off
# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relative,size=4028964k,nr_inodes=1007241,mode=755,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=809640k,mode=755,inode64)
/dev/sdal on / type ext4 (ro,relative)
```

5. **mount -o remount,rw /**

Este comando vuelve a montar el sistema de archivos raíz **/** con permisos de lectura y escritura. Por defecto, algunos sistemas pueden arrancar con el sistema de archivos raíz montado en modo de solo lectura para realizar chequeos o reparaciones antes de permitir modificaciones. Este comando cambia ese estado a lectura-escritura (**rw**), lo que permite modificar archivos en el sistema de archivos raíz.

6. **passwd - 123**

```
# Mount -o remount,rw /
# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relative,size=4028964k,nr_inodes=1007241,mode=755,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=809640k,mode=755,inode64)
/dev/sdal on / type ext4 (rw,relative,errors=reMount-ro)
# passwd
New password:
Retype new password:
passwd: password updated successfully
#
```

7. Modificar opciones avanzadas **sudo nano /etc/grub.d/10\_linux**

El comando **sudo nano /etc/grub.d/10\_linux** abre el archivo **10\_linux**, ubicado en el directorio **/etc/grub.d/**, en el editor de texto **nano** con permisos de superusuario.

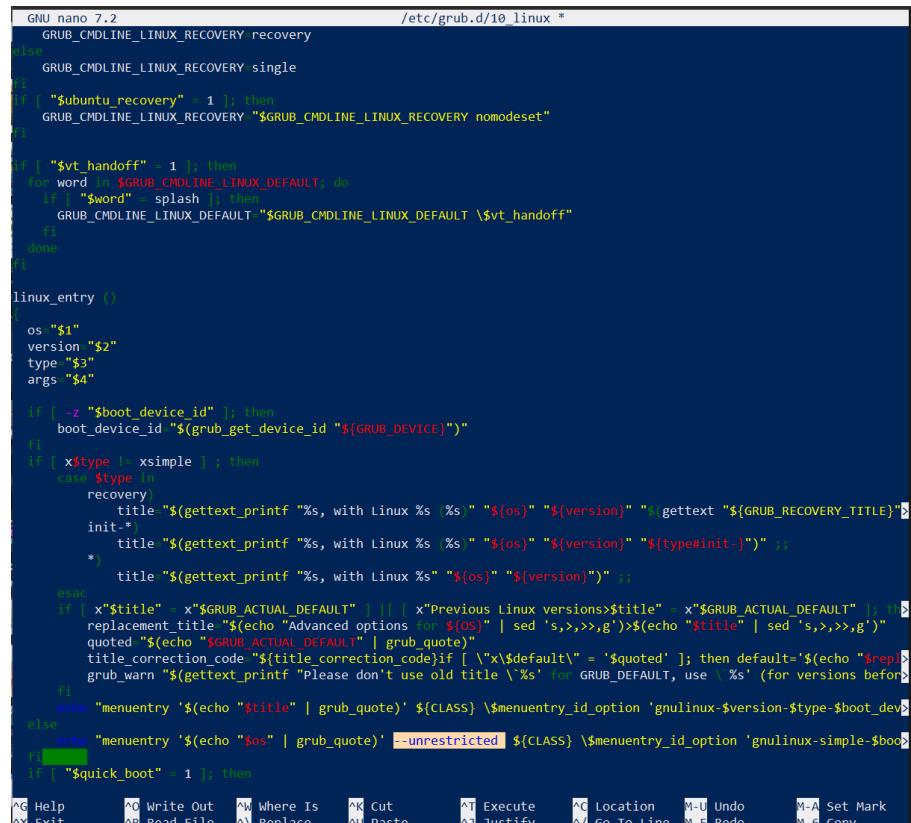
Este archivo es parte de la configuración de GRUB (GRand Unified Bootloader) en sistemas Linux. Específicamente, el archivo **10\_linux** se encarga de detectar y generar entradas en el menú de GRUB para los sistemas operativos Linux instalados en la máquina. Cuando ejecutas **sudo update-grub**, el contenido de este archivo se utiliza para construir el archivo de configuración principal de GRUB.

8. Vamos a la linea 132

---

## 9. --unrestricted antes de la etiqueta \${CLASS}.

En el contexto de GRUB, la opción –unrestricted se puede agregar a la configuración de una entrada de menú para hacerla accesible sin restricciones, es decir, no requiere autenticación (contraseña) para ser seleccionada desde el menú de GRUB. La etiqueta CLASS en los scripts de GRUB se utiliza para definir clases de entrada de menú, que pueden ser utilizadas para aplicar configuraciones específicas a diferentes entradas del menú de arranque. Colocar –unrestricted antes de \${CLASS} significa que esa entrada de menú específica se podrá seleccionar libremente en GRUB, incluso si otras entradas requieren autenticación.



```
GNU nano 7.2                               /etc/grub.d/10_linux *

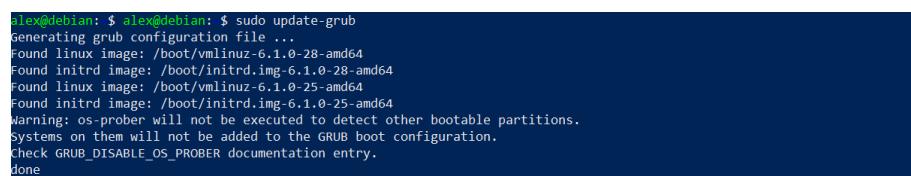
GRUB_CMDLINE_LINUX_RECOVERY=recognition
else
    GRUB_CMDLINE_LINUX_RECOVERY=single
fi
if [ "$ubuntu_recovery" = 1 ]; then
    GRUB_CMDLINE_LINUX_RECOVERY="$GRUB_CMDLINE_LINUX_RECOVERY nomodeset"
fi

if [ "$vt_handoff" = 1 ]; then
    for word in $GRUB_CMDLINE_LINUX_DEFAULT; do
        if [ "$word" = splash ]; then
            GRUB_CMDLINE_LINUX_DEFAULT="$GRUB_CMDLINE_LINUX_DEFAULT \$vt_handoff"
        fi
    done
fi

linux_entry ()
{
    os "$1"
    version "$2"
    type "$3"
    args "$4"

    if [ -z "$boot_device_id" ]; then
        boot_device_id "$(grub_get_device_id "$GRUB_DEVICE")"
    fi
    if [ x$type != xsimple ]; then
        case $type in
            recovery
                title "$(gettext_printf "%s, with Linux %s" "${os}" "${version}")" "${gettext "${GRUB_RECOVERY_TITLE}"}"
            init*
                title "$(gettext_printf "%s, with Linux %s" "${os}" "${version}")" "${(type#init-)}" ;;
            *)
                title "$(gettext_printf "%s, with Linux %s" "${os}" "${version}")" ;;
        esac
        if [ "$title" = "$GRUB_ACTUAL_DEFAULT" ] || [ x"Previous Linux versions$title" = x"$GRUB_ACTUAL_DEFAULT" ]; then
            replacement_title "$(echo "Advanced options for ${os}" | sed 's,>,>>g')$(echo "$title" | sed 's,>,>>g')"
            quoted="$(echo "$GRUB_ACTUAL_DEFAULT" | grub quote)"
            title_correction_code "${{title_correction_code}}if [ \"x$default\" = \"$quoted\" ]; then default='$(echo "$replacement_title" | sed 's,>,>>g')'; grub_warn "$(gettext_printf "Please don't use old title \\%s' for GRUB_DEFAULT, use \\%s' (for versions before 2.02)." "$title" "$replacement_title")'" ;;
        else
            echo "menuentry '$(echo "$os" | grub_quote)' --unrestricted ${CLASS} \${menuentry_id_option 'gnulinux-$version-$type-$boot_device_id' $args}"
        fi
    else
        if [ "$quick_boot" = 1 ]; then
            # ...
        fi
    fi
}
```

## 10. Despues de haber guardado los cambios, ponemos el comando sudo update-grub



```
alex@debian: ~ alex@debian: ~$ sudo update-grub
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-6.1.0-28-amd64
Found initrd image: /boot/initrd.img-6.1.0-28-amd64
Found linux image: /boot/vmlinuz-6.1.0-25-amd64
Found initrd image: /boot/initrd.img-6.1.0-25-amd64
warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
```

---

# Protección con Contraseña de GRUB

## 1. sudo grub-mkpasswd-pbkdf2

El comando sudo grub-mkpasswd-pbkdf2 se utiliza para generar una contraseña cifrada para usar en la configuración de GRUB. Específicamente, este comando genera un hash PBKDF2 (Password-Based Key Derivation Function 2) de la contraseña que proporcionas. Esta hash se puede usar para proteger el menú de GRUB con una contraseña.

```
alex@debian: $ sudo grub-mkpasswd-pbkdf2
Enter password:
Reenter password:
PBKDF2 hash of your password is grub.pbkdf2.sha512.10000.DAC6778D4B6B5CF441395D69156434581A773FA7FB389AA6D7172A5DA54CCE
EEF7A28A239C8CD620FB98FAE9D40AFBDF920E873205F695C268F928BE186261A.C06AD6AF0A8C6376619007A917C2F6C647AD6CE903299BACB92A18
[REDACTED]
```

## 2. sudo nano /etc/grub.d/40\_custom

El comando sudo nano /etc/grub.d/40\_custom abre el archivo 40\_custom en el directorio /etc/grub.d/ usando el editor de texto nano con permisos de superusuario.

El archivo 40\_custom es un archivo de configuración utilizado por GRUB para añadir entradas personalizadas al menú de arranque. Este archivo se usa para definir entradas de menú adicionales o personalizadas que no se generan automáticamente por otros scripts en /etc/grub.d/.

## 3. Colocamos el usuario root y la contraseña encriptada. Finalmente, guardamos el script y usamos sudo update-grub

```
#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries. Simply type the
# menu entries you want to add after this comment. Be careful not to change
# the 'exec tail' line above.
set superusers "root"
password_pbkdf2 root grub.pbkdf2.sha512.10000.DAC6778D4B6B5CF441395D69156434581A773FA7FB389AA6D7172A5DA54CCEEEF7A28A23>
```

## 4. Reiniciamos el sistema, comprobamos que al momento de que inicie el sistema, será de forma normal, pidiendo nuestro usuario y contraseña.

```
Debian GNU/Linux 12 debian tty1

debian login:
```

## 5. Volvemos a reiniciar el sistema y comprobaremos que al intentar ingresar en el menú de edición de las entradas del GRUB, esta pedirá el usuario root y la contraseña que creamos para generar la contraseña encriptada.

```
Enter username:
root
Enter password:
-
```