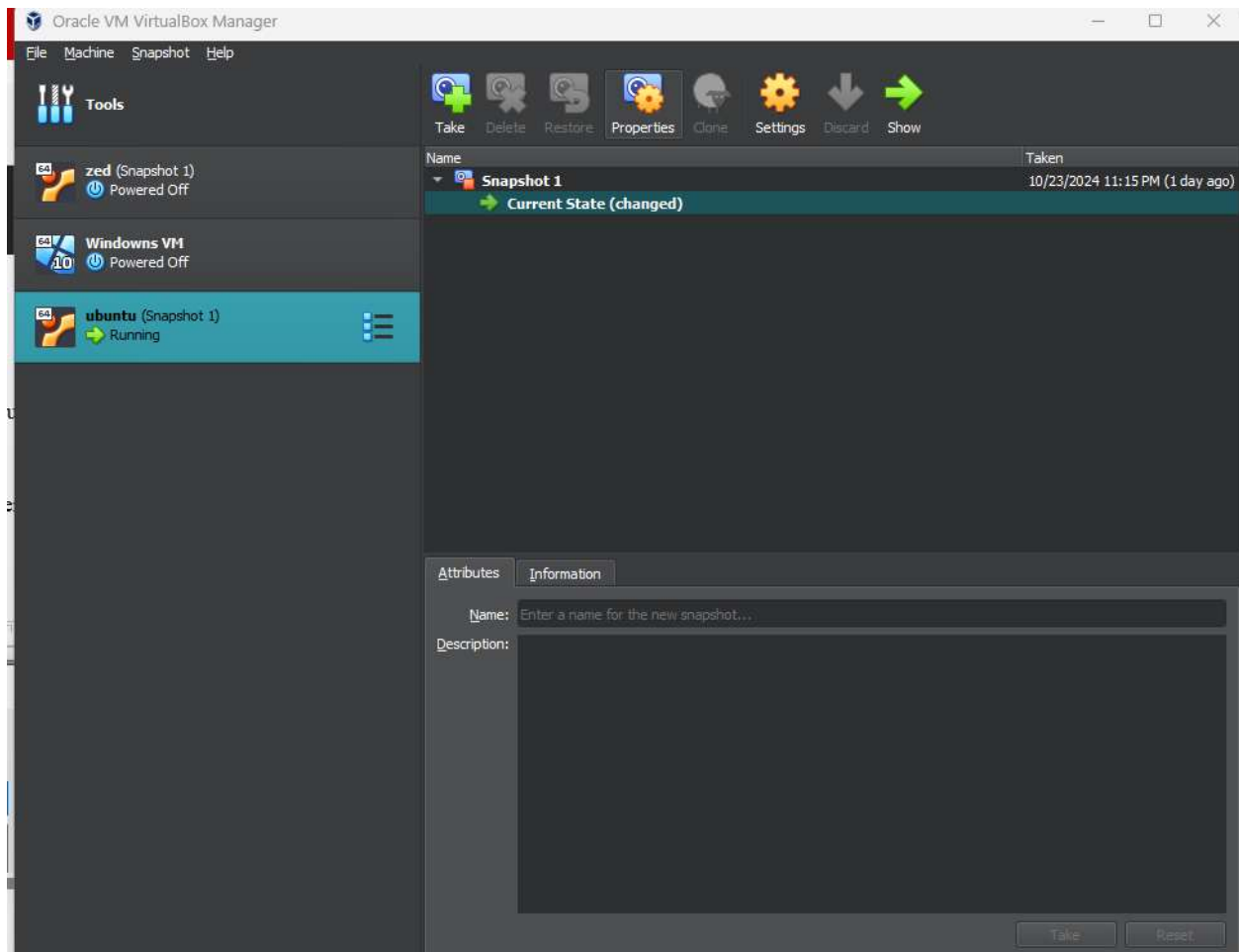


TP2 Unix

Farouk Benderouiche

1. Snapshot

Un snapshot est une "capture instantanée" d'un système ou d'un état de machine virtuelle à un moment précis. Cela permet de sauvegarder une copie exacte de la VM (incluant le système d'exploitation, les fichiers, et les configurations) pour pouvoir y revenir plus tard en cas de besoin.



UEFI BOOT MANAGER

```
farouk@faroukk:~$ [ -d /sys/firmware/efi ] && echo "session EFI " || echo "session non-efi"
session EFI
farouk@faroukk:~$
```

2.

```
farouk@faroukk:~$ efibootmgr
BootCurrent: 0004
Timeout: 0 seconds
BootOrder: 0004,0000,0001,0002,0003
Boot0000* UiApp
Boot0001* UEFI VBOX CD-ROM VB0-01f003f6
Boot0002* UEFI VBOX HARDDISK VB89cd83ef-25973fdc
Boot0003 EFI Internal Shell
Boot0004* ubuntu
farouk@faroukk:~$ efibootmgr -v
BootCurrent: 0004
Timeout: 0 seconds
BootOrder: 0004,0000,0001,0002,0003
Boot0000* UiApp FvVol(7cb8bdc9-f8eb-4f34-aaea-3ee4af6516a1)/FvFile(462caa21-7614-4503-836e-8ab6f4662331)
Boot0001* UEFI VBOX CD-ROM VB0-01f003f6 PciRoot(0x0)/Pci(0x1,0x1)/Ata(0,0,0)N....YM....R,Y.
Boot0002* UEFI VBOX HARDDISK VB89cd83ef-25973fdc PciRoot(0x0)/Pci(0xd,0x0)/Sata(0,65535,0)N....YM....R,Y.
Boot0003 EFI Internal Shell FvVol(7cb8bdc9-f8eb-4f34-aaea-3ee4af6516a1)/FvFile(7c04a583-9e3e-4f1c-ad65-e05268d0b4d1)
Boot0004* ubuntu HD(1,GPT,7e328f7a-902a-4181-95c7-0e53e8d1c853,0x800,0x165800)/File(\EFI\ubuntu\shimx64.efi)
farouk@faroukk:~$ efibootmgr
BootCurrent: 0004
Timeout: 0 seconds
```

3./4.

```

62331)
Boot0001* UEFI VBOX CD-ROM VB0-01f003f6      PciRoot(0x0)/Pci(0x1,0x1)/Ata(0,0,0)N.....YM....R,
Y.
Boot0002* UEFI VBOX HARDDISK VB89cd83ef-25973fdc      PciRoot(0x0)/Pci(0xd,0x0)/Sata(0,65535,0)N
.....YM....R,Y.
Boot0003 EFI Internal Shell      FvVol(7cb8bdc9-f8eb-4f34-aaea-3ee4af6516a1)/FvFile(7c04a583-9e3e-4
f1c-ad65-e05268d0b4d1)
Boot0004* ubuntu      HD(1,GPT,7e328f7a-902a-4181-95c7-0e53e8d1c853,0x800,0x165800)/File(\EFI\ub
untu\shimx64.efi)
farouk@faroukk:~$ efibootmgr
BootCurrent: 0004
Timeout: 0 seconds
BootOrder: 0004,0000,0001,0002,0003
Boot0000* UiApp
Boot0001* UEFI VBOX CD-ROM VB0-01f003f6
Boot0002* UEFI VBOX HARDDISK VB89cd83ef-25973fdc
Boot0003 EFI Internal Shell
Boot0004* ubuntu
farouk@faroukk:~$ sudo efibootmgr -o 0001,0002,0000,0004,0003
BootCurrent: 0004
Timeout: 0 seconds
BootOrder: 0001,0002,0000,0004,0003
Boot0000* UiApp
Boot0001* UEFI VBOX CD-ROM VB0-01f003f6
Boot0002* UEFI VBOX HARDDISK VB89cd83ef-25973fdc
Boot0003 EFI Internal Shell
Boot0004* ubuntu
farouk@faroukk:~$

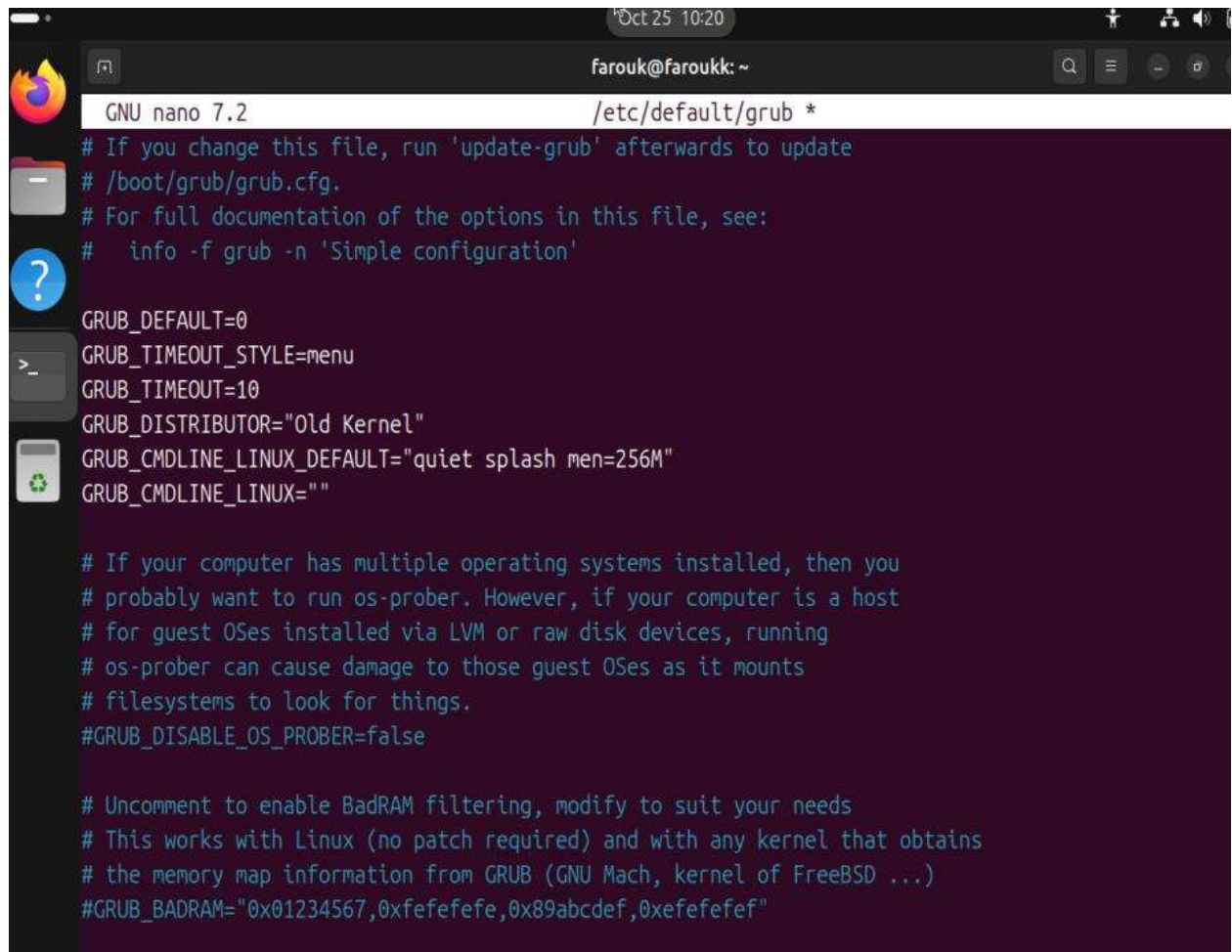
```

2. Affichage de menu GRUB

/etc/default/grub

GRUB_TIMEOUT=10

GRUB_DISTRIBUTOR ="Old Kernel"



The image shows a terminal window with a dark background. At the top, the title bar indicates the date and time as 'Oct 25 10:20' and the user as 'farouk@faroukk: ~'. The terminal is running the 'nano' text editor, editing the file '/etc/default/grub'. The file content is displayed in a light blue font on a dark background. The content includes comments about updating grub, documentation links, and various configuration options like GRUB_DEFAULT, GRUB_TIMEOUT_STYLE, GRUB_TIMEOUT, GRUB_DISTRIBUTOR, GRUB_CMDLINE_LINUX_DEFAULT, and GRUB_CMDLINE_LINUX. There are also comments about os-prober and BadRAM filtering.

```
GNU nano 7.2 /etc/default/grub *
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=menu
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR="Old Kernel"
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash mem=256M"
GRUB_CMDLINE_LINUX=""

# If your computer has multiple operating systems installed, then you
# probably want to run os-prober. However, if your computer is a host
# for guest OSes installed via LVM or raw disk devices, running
# os-prober can cause damage to those guest OSes as it mounts
# filesystems to look for things.
#GRUB_DISABLE_OS_PROBER=false

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"
```

3. Intitulé dans le menu

- 1.
- 2.

```
10_linux
/etc/grub.d

26 gtzpayload_dynamic="1"
27 vt_handoff="1"
28
29 . "$pkgdatadir/grub-mkconfig_lib"
30
31 export TEXTDOMAIN=grub
32 export TEXTDOMAINDIR="${datarootdir}/locale"
33
34 CLASS="--class gnu-linux --class gnu --class os"
35
36 if [ "x${GRUB_DISTRIBUTOR}" = "x" ] ; then
37     OS=GNU/Linux
38 else
39     case ${GRUB_DISTRIBUTOR} in
40         Ubuntu|Kubuntu)
41             OS="${GRUB_DISTRIBUTOR}"
42             ;;
43         *)
44             OS="${GRUB_DISTRIBUTOR} GNU/Linux"
45             ;;
46     esac
47     CLASS="--class $(echo ${GRUB_DISTRIBUTOR} | tr 'A-Z' 'a-z' | cut -d'
48     ' -f1|LC_ALL=C sed 's,[^[:alnum:]]_,_,g') ${CLASS}"
49 fi
50 # loop-AES arranges things so that /dev/loop/X can be our root device,
```

3.

```
GNU nano 7.2 /etc/grub.d/40_custom
#!/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.

menuentry "Shutdown" {
    echo "system shutting down..."
    halt
}

menuentry "Reboot" {
    echo "system rebooting ..."
    reboot
}
```

4. Ajout d'un nouveau noyau

1.

```
farouk@faroukk:~$ uname -r
6.5.0-44-generic
farouk@faroukk:~$
```

2.


```

farouk@faroukk:~$ cd Downloads
farouk@faroukk:~/Downloads$ ls
linux-headers-5.19.0-051900_5.19.0-051900.202207312230_all.deb
linux-headers-5.19.0-051900-generic_5.19.0-051900.202207312230_amd64.deb
linux-image-unsigned-5.19.0-051900-generic_5.19.0-051900.202207312230_amd64.deb
linux-modules-5.19.0-051900-generic_5.19.0-051900.202207312230_amd64.deb
farouk@faroukk:~/Downloads$ sudo dpkg -i *.deb
[sudo] password for farouk:
Selecting previously unselected package linux-headers-5.19.0-051900.
(Reading database ... 144079 files and directories currently installed.)
Preparing to unpack linux-headers-5.19.0-051900_5.19.0-051900.202207312230_all.deb ...
Unpacking linux-headers-5.19.0-051900 (5.19.0-051900.202207312230) ...
Selecting previously unselected package linux-headers-5.19.0-051900-generic.
Preparing to unpack linux-headers-5.19.0-051900-generic_5.19.0-051900.202207312230_amd64.deb
Unpacking linux-headers-5.19.0-051900-generic (5.19.0-051900.202207312230) ...
Selecting previously unselected package linux-image-unsigned-5.19.0-051900-generic.
Preparing to unpack linux-image-unsigned-5.19.0-051900-generic_5.19.0-051900.202207312230_amd64.deb
Unpacking linux-image-unsigned-5.19.0-051900-generic (5.19.0-051900.202207312230) ...
Selecting previously unselected package linux-modules-5.19.0-051900-generic.
Preparing to unpack linux-modules-5.19.0-051900-generic_5.19.0-051900.202207312230_amd64.deb
Unpacking linux-modules-5.19.0-051900-generic (5.19.0-051900.202207312230) ...
Setting up linux-headers-5.19.0-051900 (5.19.0-051900.202207312230) ...
Setting up linux-headers-5.19.0-051900-generic (5.19.0-051900.202207312230) ...
Setting up linux-image-unsigned-5.19.0-051900-generic (5.19.0-051900.202207312230) ...
Setting up linux-modules-5.19.0-051900-generic (5.19.0-051900.202207312230) ...

```

3.

```

Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
farouk@faroukk:~/Downloads$ sudo apt-get -f install
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 78 not upgraded.
farouk@faroukk:~/Downloads$

```

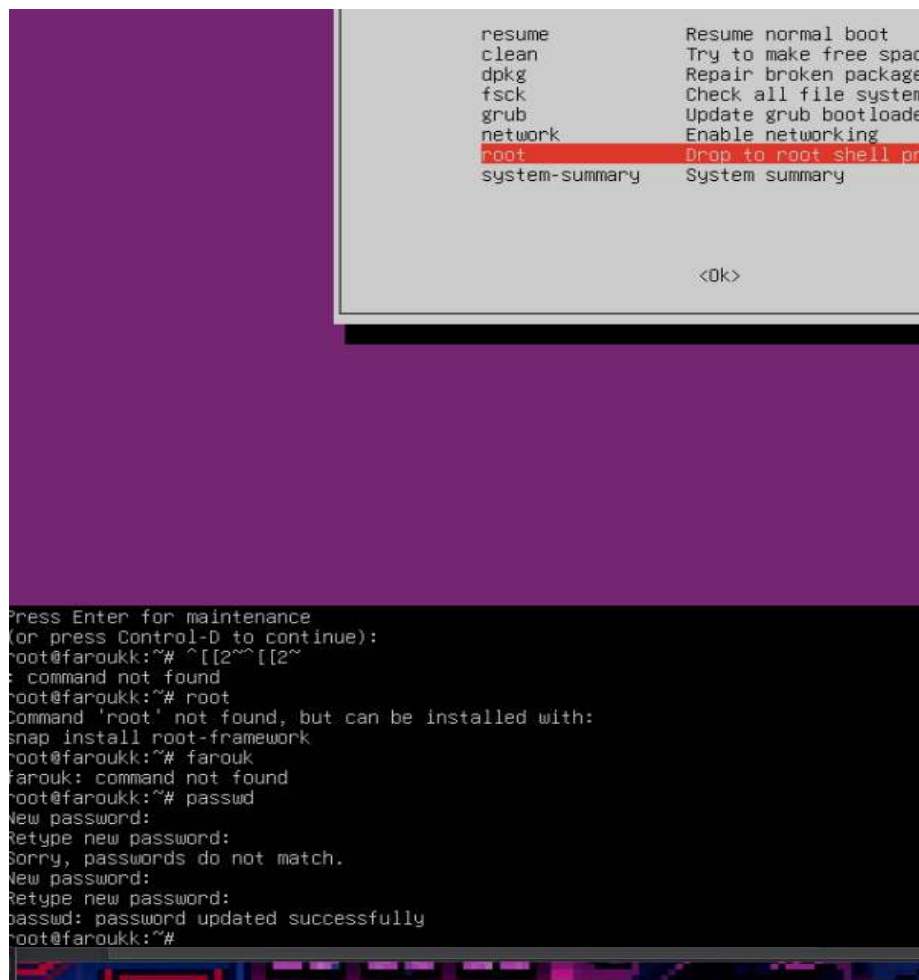
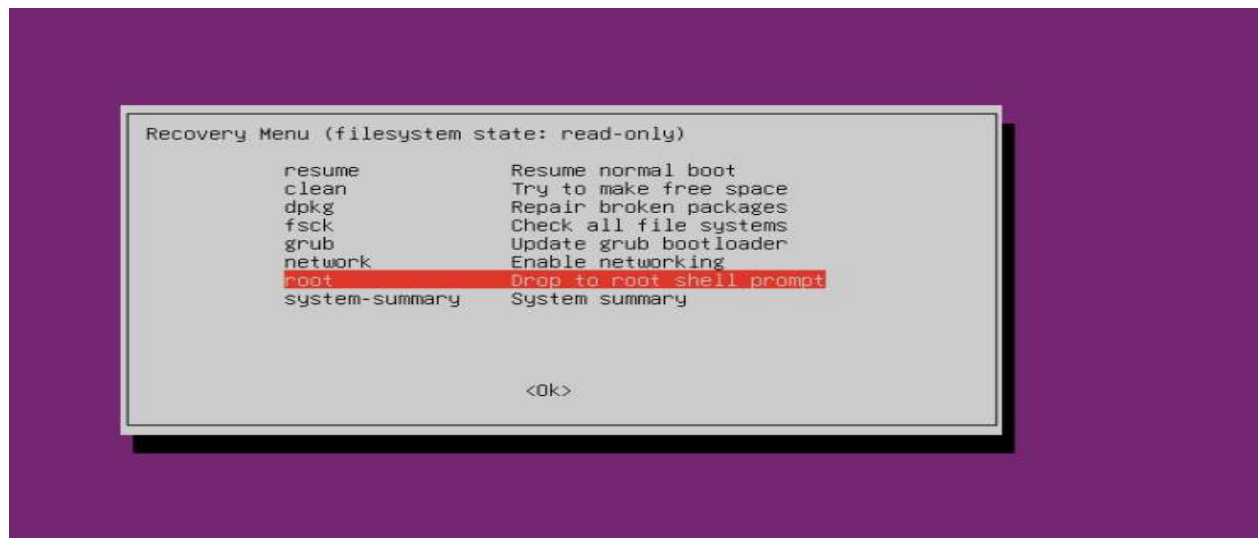
4.

```
Open  *40_custom  Save  -  +  x
      /etc/grub.d
2 exec tail -n +3 $0
3 # This file provides an easy way to add custom menu entries.  Simply
4 # type the
5 # menu entries you want to add after this comment.  Be careful not to
6 # change
7 # the 'exec tail' line above.
8
9 menuentry "Shutdown" {
10     echo "system shutting down..."
11     halt
12 }
13
14 menuentry "Reboot" {
15     echo "system rebooting ..."
16     reboot
17 }
18
19 menuentry 'New_kernel, with Linux 5.19.0-051900-generic' {
20     set root = 'hd0,1'
21     linux /boot/vmlinuz-5.19.9-0-0-generic root=/dev/sda1 ro quiet
22     splash
23     initrd /boot/initrd.img-5.19.0.0-generic
24 }
sh  Tab Width: 8  Ln 16, Col 1  INS
```

5. Démarrage en mode « Recovery »

1./2./3.

```
Old_kernel, with Linux 6.5.0-44-generic
Old_kernel, with Linux 6.5.0-44-generic (recovery mode)
Old_kernel, with Linux 5.19.0-051900-generic
*Old_kernel, with Linux 5.19.0-051900-generic (recovery mode)
```

6. Protection de grub2 par un mot de passe

1./2.

```
GNU nano 7.2 /etc/grub.d/00_header *
echo else
make_timeout "${GRUB_HIDDEN_TIMEOUT}" "${GRUB_TIMEOUT}" "${GRUB_TIMEOUT_STYLE}"
echo fi
else
make_timeout "${GRUB_HIDDEN_TIMEOUT}" "${GRUB_TIMEOUT}" "${GRUB_TIMEOUT_STYLE}"
fi

if [ "x${GRUB_BUTTON_CMOS_ADDRESS}" != "x" ] && [ "x${GRUB_BUTTON_CMOS_CLEAN}" = "xyes" ]; then
    cat <<EOF
cmosclean ${GRUB_BUTTON_CMOS_ADDRESS}
EOF
fi

# Play an initial tune
if [ "x${GRUB_INIT_TUNE}" != "x" ]; then
    echo "play ${GRUB_INIT_TUNE}"
fi

if [ "x${GRUB_BADRAM}" != "x" ]; then
    echo "badram ${GRUB_BADRAM}"
fi

set superusers="root"
password_pbkdf2 root grub.pbkdf2.sha512.10000.30ED40AA263B74B0963514DFB47E9CE50268F09B1D058431B2C>

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

7. Modifier les paramètres de démarrage de noyau

1.

```
fi
echo 'Loading Linux 6.5.0-44-generic ...'
linux /boot/vmlinuz-6.5.0-44-generic root=UUID=354dc2f-a002-4500-98c9-a7030d02657e ro quiet spl\
ash mem=256M $vt_handoff
echo 'Loading initial ramdisk ...'
```

2.

```
Oct 24 23:00
farouk@farouk: ~
GNU nano 7.2 /etc/default/grub *
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
# info -f grub -n 'Simple configuration'

GRUB_DEFAULT="New Kernel"
GRUB_TIMEOUT_STYLE=menu
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash mem=256M"
GRUB_CMDLINE_LINUX=""

# If your computer has multiple operating systems installed, then you
# probably want to run os-prober. However, if your computer is a host
# for guest OSes installed via LVM or raw disk devices, running
# os-prober can cause damage to those guest OSes as it mounts
# filesystems to look for things.
#GRUB_DISABLE_OS_PROBER=false

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```