

GRUB 2 ISO Boot

Fernão Vellozo Ribas

#FiqueEmCasaUseDebian
Curitiba/PR - 21 de maio de 2020

about:me



/fernao.vellozo



/fernaovellozo



@fernaovellozo

<https://cryptostratus.net/#palestras>

fernao@cryptostratus.net

about:topics

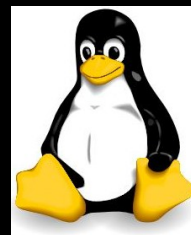
Apenas para fins educativos: cada um é responsável por seus atos e escolhas, tanto online como offline;

Por default: segurança, privacidade e liberdade para users; transparência e responsabilidade para vendors;

Por que usar software livre? Porque “um sistema deve ser seguro ainda que tudo sobre tal sistema, exceto a chave, seja de conhecimento público”.

(Auguste Kerckhoffs - La Cryptographie Militaire, 1883)

GRUB → GRand Unified Bootloader, mas também pode significar, entre outras coisas, “rango” ou “larva”



GRUB 2

PC Architecture:

- x86_64 || i686
- BIOS || *EFI
- MBR || GPT
- Física || Virtual

Funciona também em
ARM, MIPS e QEMU,
entre outros

Antes:

- LiLo
- SysLinux
- Grub 0.9x (legacy)

Agora:

- Grub 2
- SysLinux
- SystemD-Boot

Power On → POST → BIOS || UEFI → Bootloader

Em uma instalação existente, serão adicionadas entradas para live boot de Archlinux, Parabola, Debian e Tails, assim como de reboot e shutdown

Para salvar as alterações:

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

(em algumas distros, usa-se “grub-update” ou “update-grub”, mas o efeito é o mesmo)

Arquivos de interesse:

- /boot/grub/grub.cfg (não editar diretamente)
- /etc/default/grub
- /etc/grub.d/40_custom

/etc/default/grub

GRUB_DEFAULT=0

GRUB_TIMEOUT=10 ## tempo de exposição do menu

GRUB_CMDLINE_LINUX_DEFAULT=quiet splash ## opções extra em modo normal

GRUB_CMDLINE_LINUX={ not today - main kernel parameters }

GRUB_PRELOAD_MODULES="part_gpt part_msdos ext2 loopback lvm cryptodisk luks"

#GRUB_ENABLE_CRYPTODISK=y ## uncomment se houver disco encriptado

GRUB_COLOR_NORMAL="green/black"

GRUB_COLOR_HIGHLIGHT="green/white"

#GRUB_BACKGROUND="/boot/grub/path/to/wallpaper.png"

#GRUB_THEME="/boot/grub/path/to/gfxtheme"

GRUB_GFXMODE={ auto || 1024x768 } ## resolução de tela

GRUB_GFXPAYLOAD_LINUX=keep ## manter a resolução durante o boot do SO

/etc/grub.d/40_custom

```
#!/bin/sh
exec tail -n +3 $0

set extdisk="0bfafac6-c450-4a1f-bc61-c5cc76132418"
set kmsg="..Loading Kernel.."
set imsg="..Loading Initram.."

menuentry "Restart the computer" {
    echo "...System rebooting..."
    reboot
}

menuentry "Shutdown the computer" {
    echo "...System shutting down..."
    halt
}

menuentry "Some Distro" --class iso{
    search --set=root --fs-uuid ${extdisk}
    loopback loop ${isopath}/path/to/some_distro.iso
    linux (loop)/path/to/vmlinuz
    echo ${kmsg}
    initrd (loop)/path/to/initrd
    echo ${img}
}
```

Muito cuidado com ctrl-c/ctrl-v, pois caracteres como aspas simples ou duplas podem ter codificações diferentes;

Pode-se usar variáveis globais para referenciar o UUID do dispositivo que contém as imagens, que pode ser disco fixo, pendrive e até mesmo servidores nfs ou http;

A imagem iso é montada como dispositivo loopback e configurada como raiz no contexto do Grub;

Executa-se o kernel seguido da initram, iniciando o sistema live a partir desse momento;

Os parametros do kernel variam loucamente de distro para distro, embora sigam determinados padrões;

Para extrair o UUID da partição que armazena as imagens:

```
# isouuid=`lsblk -o UUID -nodeps -n /dev/sdaX`  
# echo $isouuid
```




Debian / Tails



```
menuentry "Debian Mate" {
  search --set=root --fs-uuid ${extdisk}
  set isofile="/iso/debian-live-10.4.0-amd64-mate.iso"
  loopback loop ${isofile}
  echo ${kmsg}

  linux (loop)/live/vmlinuz-4.19.0-9-amd64 boot=live findiso=${isofile} iso-
scan/filename=${isofile}

  echo ${imgmsg}

  initrd (loop)/live/initrd.img-4.19.0-9-amd64
}

menuentry "Tails" {
  set isofile="/iso/tails.iso"
  search --no-floppy -f --set=root $isofile
  loopback loop $isofile

  linux (loop)/live/vmlinuz boot=live iso-scan/filename=${isofile} findiso=${
isofile} apparmor=1 nopersistence noprompt timezone=Etc/UTC
block.events_dfl_poll_msecs=1000 splash noautologin module=Tails
slab_nomerge slub_debug=FZP mce=0 vsyscall=none page_poison=1

  initrd (loop)/live/initrd.img
}
```



ArchLinux / Parabola



```
menuentry "Archlinux Live" {
  search --set=root --fs-uuid ${extdisk}
  set isofile="/iso/archlinux-2020.05.01-x86_64.iso"
  loopback loop ${isofile}
  echo ${kmsg}

  linux (loop)/arch/boot/x86_64/vmlinuz img_loop=${isofile} earlymodules=loop
  archisolabel=ARCH_202005 archbasedir=arch img_dev=/dev/disk/by-uuid/${extdisk}

  echo ${img}

  initrd (loop)/arch/boot/x86_64/archiso.img
}
```

```
menuentry "Parabola LXDE" {
  search --set=root --fs-uuid ${extdisk}
  set isofile="/iso/parabola-openrc-lxde-2020.01.18-dual.iso"
  loopback loop ${isofile}
  echo ${kmsg}

  linux (loop)/parabola/boot/x86_64/vmlinuz img_loop=${isofile}
  parabolaisolabel=PARA_202001 parabolabasedir=parabola img_dev=/dev/disk/by-
  uuid/${extdisk}

  echo ${img}

  initrd (loop)/parabola/boot/x86_64/parabolaiso.img
}
```



Kali / Ubuntu



```
menuentry "Kali x86_64 Mate" {
  search --set=root --fs-uuid ${isouuid}
  set isofile="/iso/kali-linux-mate-2018.4-amd64.iso"
  loopback loop ${isofile}
  echo ${kmsg}
  linux (loop)/live/vmlinuz boot=live img_dev=${isopath} findiso=${isofile}
  noconfig=sudo username=root hostname=viceroy timezone=Etc/BRT
  echo ${imgmsg}
  initrd (loop)/live/initrd.img
}

menuentry "Ubuntu x86_64 LTS" {
  search --set=root --fs-uuid ${isouuid}
  set isofile="/iso/ubuntu-18.04.2-desktop-amd64.iso"
  loopback loop ${isofile}
  echo ${kmsg}
  linux (loop)/casper/vmlinuz boot=casper iso-scan/filename=${isofile} quiet splash
  echo ${imgmsg}
  initrd (loop)/casper/initrd
}
```

Fedora / Arch encrypted

```
menuentry "Fedora x86_64" {
  search --set=root --fs-uuid ${guuid}
  set isofile="/iso/fedora-workstation-live-x86_64-29-1.2.iso"
  set isolabel="Fedora-WS-Live-29-1-2"
  loopback loop ${isofile}
  echo ${kmsg}
  linux (loop)/isolinux/vmlinuz root=live:CDLABEL=${isolabel} rd.live.image
iso-scan/filename=${isofile}
  echo ${imgmsg}
  initrd (loop)/isolinux/initrd.img
}

menuentry "Archlinux Live x86_64" {
  set isofile="/iso/archlinux-2020.05.01-x86_64.iso"
  linux (crypto0)/arch/vmlinuz img_dev=/dev/disk/by-uuid/${extdisk} img_loop=${isofile}
archisobasedir=arch archisolabel=ARCH_202005
  initrd (crypto0)/blackarch/archiso.img
}
```

Referencias:

<https://www.gnu.org/software/grub/index.html>

<https://wiki.archlinux.org/index.php/GRUB>

<https://wiki.gentoo.org/wiki/GRUB2>

<https://help.ubuntu.com/community/Grub2/ISOBoot>

<https://github.com/thias/glim>

Obrigado :)