System76 Galago Pro review

Glenn Holmer
Milwaukee Linux Users Group, 2019-01-12
Agenda

- laptop review
- my experiences installing Debian and learning about EFI
System76

- specializes in computers that run Linux well
- plan to make more of their computers in-house
- now making cases and I/O boards in their new factory in Denver for desktop machines ("Thelio")
Galago Pro laptop being reviewed

- 4GHz CPU, 16G memory, 512G NVMe SSD
- 14” matte 1080p screen (Intel UHD Graphics 620) 13.3” HiDPI also available (3200 x 1800)
- USB 3.1: 2 type A, 1 type C/Thunderbolt
- full-size SD card, “fold-out” full-size Ethernet
- video output: HDMI, mini DisplayPort
- slot for Kensington lock, unused SIM card port
first impressions: the good

- machine is blindingly fast with a gorgeous screen
- sensitive trackpad; set up to use one-, two-, and three-finger tap for left-, right-, and middle-click
- special keys worked out-of-box (in Debian!)
- battery life up to 7 hours depending on usage
- no problems multi-booting Debian and Ubuntu
first impressions: the not as good

- had to download version of Debian with wireless firmware included (not a laptop issue)
- needed to change font DPI from 96 to 120 (screen resolution almost too good!)
- trackpad buttons were quite stiff at first, trackpad doesn’t respond well in upper corners
- no caps lock light
initial configuration

- Ubuntu 18.04 installed (Pop!_OS also available)
- 512M EFI partition (no secure boot)
- 4G swap partition at end of disk
- remaining space allocated for Ubuntu (ext4)
after installation of Debian

- 512M EFI partition
- master GRUB partition (more about this later)
- Ubuntu partition shrunk to 16G
- 2 X (256M ext2 boot, 32G XFS root) for Debian
- swap partition expanded to 16G for hibernate
- remainder for “common” partition (XFS)
FUN AND GAMES WITH EFI

Caveat utilitor!
learning EFI using KVM/QEMU

- install `ovmf` to create VMs with EFI firmware
- NVRAM at `/var/lib/libvirt/qemu/nvram`
- must copy NVRAM if creating snapshots or copying disk images (which you’ll want to do in case of mishap)
EFI basics

- meant to replace BIOS/MBR
- EFI partition must be fat32, usually about 512M. Most distros mount this partition at /boot/efi.
- Each vendor puts their boot files in a directory named for the vendor (e.g. “debian”, “ubuntu”).
- EFI stores multiple boot entries (although most users won’t interact with them directly).
EFI setup

• like BIOS setup; there should be a GRUB menu item for it (or use `fwsetup` from GRUB prompt)

• you can also view EFI boot entries from Linux by installing `efibootmgr`

• EFI variables can be inspected by installing `efivar` (or viewing `/sys/firmware/efi/efi/efivars`)
EFI shell

- should be a menu entry for it on EFI setup screens
- can install one if you don’t have it
- archaic DOS-like interface, but scripting is supported
- many utilities, can edit boot entries with `bcfg`
EFI multi-boot

- Linux distros typically add GRUB menu entries for other operating systems found on disk
- can also use rEFInd in place of GRUB, or systemd-boot (kernel wrapped w/EFI stub), or even elilo

- possible to install GRUB in its own partition and chain to config files in different partitions...
master GRUB partition — the holy grail

- `grub-install --target=x86_64-efi \ --efi-directory=/boot/efi \ --boot-directory /mnt/grub \ --removable`

- `--removable` installs GRUB loader to EFI/BOOT in EFI system partition (avoiding vendor directories)

- don’t use `--bootloader-id` with `--removable`

- must create `grub.cfg` and add EFI boot entry
“chainloading” a GRUB config file

```bash
menuentry 'My Groovy OS' {
    set root=(hd0,gpt5)
    configfile /grub/grub.cfg
}
```

- can also chain to an EFI executable:

```bash
insmod chain
chainloader "\EFI\BOOT\BOOTX64.EFI"
```
editing EFI boot items... if you dare

- try `efibootmgr` (from Linux):
  ```bash
  efibootmgr -c \\
  -d /dev/nvme0n1 -p 9 \\
  -l "\EFI\BOOT\BOOTX64.EFI" -L "MASTER_GRUB"
  ```

- or `bcfg` (from an EFI shell):
  ```bash
  bcfg boot add \EFI\BOOT\BOOTX64.EFI \\
  "MASTER_GRUB"
  ```

- and hope it doesn’t get overwritten at next boot!
EFI firmware may override your intentions in an effort to protect you from yourself... but this makes it harder for you to get it to do what you want.

*and you probably won’t, since it appears for only a split-second*
EFI takeaways

• I am new at this... I’m sure there is much more for me to learn in the future.

• Be careful! Standard disclaimers apply: don’t type in commands unless you know what they do, &c.

• Practice on a VM using a glove box and hazmat suit to minimize danger.
QUESTIONS