

WHAT: Get Debian on the Thinkpad X13s

HOW: Turning on a bunch of kernel modules



Outline

Status

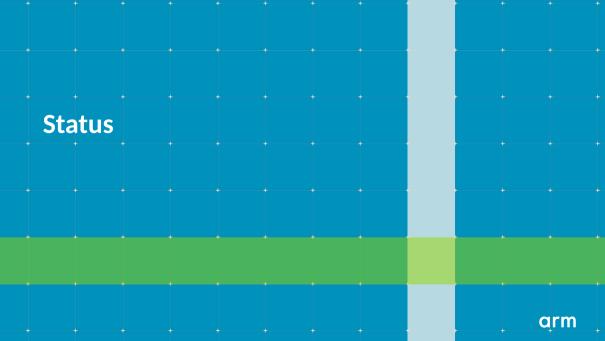
Enabling modules

Debian Installer

CD and **Live** Images

Next steps





Not supported

Bad news first

In progress:

EFI variables

Reliant on firmware changes:

- Virtualization
- Secure Boot
- Pointer Authentication



Supported

- Display
- NVMe
- Trackpoint / Touchpad
- Wi-Fi
- Bluetooth
- Sound
- Running 32 bit code



Debian on the X13s

Current status

- Fetch the daily arm64 netinst iso
- Boot with arm64.nopauth
- Force GRUB installation to the EFI removable media path? Yes.
- Before reboot:
 echo qnoc-sc8280xp > /target/etc/initramfs-tools/modules
 in-target update-initramfs -u -k all
- See https://wiki.debian.org/InstallingDebianOn/Thinkpad/X13s for the details

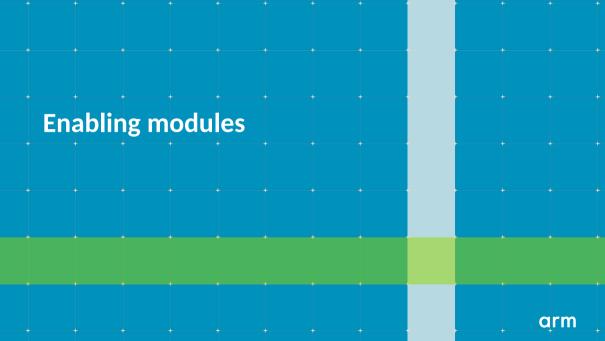


Debian on the X13s

Work done to get here

- Enabling kernel modules
- Debian Installer images
- Live images





Needed modules

- SC_DISPCC_8280XP
- SC_GCC_8280XP
- SC_GPUCC_8280XP
- QCOM_SPMI_ADC5
- INTERCONNECT_QCOM_OSM_L3
- INTERCONNECT_QCOM_SC8280XP
- LEDS_QCOM_LPG
- QCOM_IPCC
- QCOM_FASTRPC
- NVMEM_SPMI_SDAM
- PHY_OCOM_EDP
- PHY_QCOM_QMP_PCIE

- PHY_QCOM_USB_SNPS_FEMTO_V2
- PINCTRL_SC8280XP
- PINCTRL_SC8280XP_LPASS_LPI
- PINCTRL_LPASS_LPI
- POWER_RESET_QCOM_PON
- BATTERY_QCOM_BATTMGROCOM_O6V5_ADSP
- OCOM_O6V5_PAS
- QCOM_Q6V5_WCSS
- QCOM_SYSMON
- QCOM_LLCC
- QCOM_OCMEM

- OCOM_PMIC_GLINK
- QCOM_STATS
- QCOM_APR
- QCOM_ICC_BWMON
- SPI_QCOM_GENI
- TYPEC_MUX_GPIO_SBU
- QRTR_SMD
- SND_SOC_WCD938X_SDW
- SND_SOC_LPASS_WSA_MACRO
- SND_SOC_LPASS_VA_MACRO
- SND_SOC_LPASS_RX_MACRO
- SND_SOC_LPASS_TX_MACRO
- SND_SOC_ODSP6

Getting the kernel

- Clone https://salsa.debian.org/kernel-team/linux/
- Checkout the appropriate branch (sid? bookworm? master?)
- ./debian/bin/genorig.py \$UPSTREAM_KERNEL_URL

Stable kernels:

https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git

RC kernels:

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git



Enabling modules

- Edit debian/config/arm64/config
- If the setting was n, set it to m or y
- If the setting is not there, just add it somewhere (anywhere)



Fixing Kconfig

Because there's actually an order

- Clone https://salsa.debian.org/kernel-team/kernel-team/
- Run kernel-team/utils/kconfigeditor2/process.py



Building the kernel



Pro tip

Use a custom abi name

• Edit debian/config.local/defines as follows:

```
[abi] abiname: 0.my-custom-kernel
```

- Run debian/bin/gencontrol.py
- Packages renamed to: linux-image-6.6.0-0.my-custom-kernel-arm64-unsigned





Modules and the Installer

- An initrd that uses a bunch of special Debian packages called udebs
- Some udebs are used for core Installer functionalities (eg: configuring grub)
- Other udebs contain kernel modules
 - kernel-image-6.4.0-4-amd64-di_6.4.13-2_amd64.udeb
 - nic-wireless-modules-6.4.0-4-amd64-di_6.4.13-2_amd64.udeb
 - usb-modules-6.4.0-4-amd64-di_6.4.13-2_amd64.udeb
- Which udebs should be shipping our modules?



Modules and udebs

- Static maps in the kernel source
 - debian/installer/modules/arm64/kernel-image
 - debian/installer/modules/arm64/nic-wireless-modules
 - debian/installer/modules/arm64/usb-modules
- "Inheritance" supported, include directives
 - #include <kernel-image>
 - drivers/soc/**
- Some of these udebs go in the initrd, others are loaded later by the Installer
- How to find out?



Installer Images

Netboot vs Netinst

- Netboot images: mini.iso
 - GRUB, kernel, initrd (initrd 40M)
 - Not mounting the USB drive!
 - Enough modules in the initrd to use the display, keyboard and NETWORK!
 - All the udebs and debs downloaded via the network
 - nic-wireless-modules and dependencies in the initrd
- Netinst images: debian-testing-arm64-netinst.iso
 - Full fledged Debian CD (initrd 21M)
 - Lots of debs and udebs
 - The initrd has to mount the USB drive
 - nic-wireless-modules and dependencies out of the initrd



udebs and Installer Images

Which modules go into the initrd?

- Clone https://salsa.debian.org/installer-team/debian-installer/
- build/pkg-lists/netboot/arm64.cfg (nic-modules, nic-wireless-modules)
- build/pkg-lists/cdrom/arm64.cfg (isofs-modules, cdrom-core-modules)
- Modules needed by the display really should end up in the initrd https://salsa.debian.org/kernel-team/linux/-/merge_requests/853



Building a custom Installer ISO

With your own kernel!

- /path/to/kernel-team/scripts/debian-test-sign linux_6.6 rc3-1 exp1_arm64.changes
- sbuild dist=unstable extra-package=\$PWD linux-signed-arm64_6.6 rc3+1 exp1.dsc
- Copy all kernel udebs under build/localudebs/
- cd build
- fakeroot make clean_netboot build_netboot
- Fresh ISO under dest/netboot/mini.iso



Writable Installer USB stick

With your own customizations!

- The ISO format is read-only
- Building a full-fledged custom ISO (netinst or bigger) is tricky. You'll see!

Maybe we can get away with:

- · fat32 filesystem on a USB stick
- kpartx -v -a debian-testing-amd64-netinst.iso; mount /dev/mapper/loop0p1
- rsync stuff to USB stick

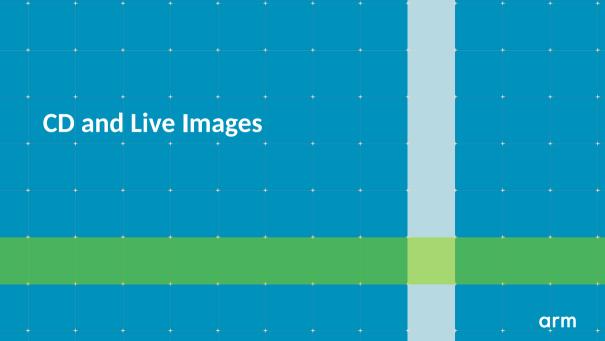


Writable Installer USB stick

Ideas for things to play with

- Replace GRUB/SHIM
- Change grub configuration (eg: add arm64.nopauth to kernel command line)
- Preseeding: create a /preseed.cfg file on the USB stick, boot with preseed/file=/cdrom/preseed.cfg
- All possible settings on https://preseed.einval.com
- d-i preseed/late_command string qnoc-sc8280xp >
 /target/etc/initramfs-tools/modules; in-target update-initramfs
 -u -k all





Building an actual CD Image

Prerequisistes

- In debian-installer/build: fakeroot make build_cdrom_grub build_cdrom_gtk
- Full mirror of the Debian archive:

```
debmirror --rsync-extra=doc --getcontents --nosource
--method=http --host=deb.debian.org --root=:debian --arch=arm64
--dist=sid --di-dist=dists --di-arch=arches --diff=use
--section=main,contrib,non-free-firmware,main/debian-installer
/srv/mirror/debian/
```

- You may get away with --exclude='dbg|gcc' but don't be too smart
- apt-get source debian-cd; apt build-dep debian-cd



Building an actual CD Image

- Patch tools/apt-selection for unsigned repos (https://bugs.debian.org/896638)
- -o Acquire::AllowInsecureRepositories=true seems needed too
- Edit CONF.sh (LOCALDEBS)
- Edit easy-build.sh (DI_DIR, CODENAME, ...)
- ARCHIVE_UNSIGNED=1./easy-build.sh CD arm64
- ARCHIVE_UNSIGNED=1./easy-build.sh NETINST arm64



Building a Live Image

Standard

- You have to, no images built for arm64 due to unreproducible cross-built images. salsa/live-team/live-build/-/merge_requests/294
- · apt install live-build
- lb config --distribution sid --updates false --archive-areas 'main non-free-firmware' --bootloaders grub-efi
- · lb build

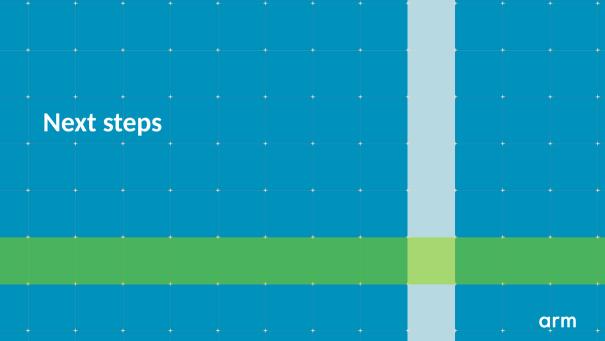


Building a Live Image

Customizations

- Call lb config with --bootappend-live arm64.nopauth modprobe.blacklist=qcom_q6v5_pas
- We also need a custom initrd! qcom-ipcc, qnoc-sc8280xp, gcc-sc8280xp pinctrl-sc8280xp, qrtr, qrtr-mhi
- Add them to config/includes.chroot_after_packages/etc/initramfs-tools/modules
- echo live-task-lxde > config/package-lists/desktop.list.chroot
- Custom packages? Put the debs in config/packages.chroot/





Debian on the X13s

Next steps

Step

- Firmware update to fix pointer authentication
- softdep to specify dependencies between modules
- efivar support: QCOM_QSEECOM QCOM_QSEECOM_UEFISECAPP

No more

- Booting with arm64.nopauth
- Manually listing modules in /etc/initramfs-tools/modules
- Installing grub to the EFI removable path



Conclusions

- You can use Debian on the Lenovo Thinkpad X13s today
- Hardware enablement is an iterative process
- The Debian Installer is pretty cool actually
- Hopefully Live images for arm64 soon!



Thank You!

Danke!

Merci!

谢谢!

ありがとう!

Gracias!

Kiitos! 감사합니다

धन्यवाद

