Install Debian arm64 into a micro SD

You might follow the instructions shown in https://libreboot.org/docs/uboot/uboot-debian-bookworm.html with the difference that we'll clone the installer to a USB stick instead to a micro SD, since we will use a SD as the place where we will be installing Debian to. Most USB stick are type A, so you will need an adapter or a hub to connect the stick to the USB type C on the computer. We will reserve the computer's internal mmc drive intact as the place we will be installing Trisquel into. You should be able to select the USB stick on the libreboot boot process no problem, and the installation process should be really straight forward, just select the micro SD as the target for the installation, it's name should be "mmcblk1", since number 0 is the internal mmc drive. In my case, choosing to encrypt partitions on the installation on the external SD made the installer hang every time I tried, but since I'm not planning to use Debian for any other reason but to install Trisquel I didn't mind and start the process without it.

Get trisquel scripts

Install debootstrap on a working trisquel computer (no matter if x86 or arm64), and browse /usr/share/debootstrap/scripts/. What you want to do is to copy the scripts there to get them to the micro SD Debian in the Chromebook you want to install Trisquel into. You might copy all of them but my guess is that only the following are really needed:

- aramo
- trisquel
- trisquel-common

You might use a USB stick copy the files over, but you should know that the "aramo" file is a sym-link, and if your stick is formatted FAT you won't be able to copy it. It seems that "aramo" is just a sym-link to the "trisquel" so you might re created the sym-link once you copy it back to Debian. Or if you are lazy as I am, just format your stick as NTFS.

Start a root session

Start Debian from the micro SD on the chromebook, then open terminal an run "su --" to start a root session.

Install debootstrap

Run "apt install debootstrap".

Paste the scripts we got from trisquel

We'll paste the files we got from Trisquel to the same route /usr/share/debootstrap/scripts/ on Debian. To make it easy, it is possible to use the following command to move all files from one place to another:

mv <route of trisquel scripts>/* /usr/share/debootstrap/scripts/

Prepare partitions

We will now prepare the internal mmc drive for Trisquel. In this tutorial I'll show you how to use an encrypted partition for it. I might re write this tutorial some other time to do it into an non encrypted one, which would probably reduce the complexity, but you might figure it out on your own.

Lets install some requirements before we continue:

apt install cryptsetup lvm2

Then lets create a new gpt partition table for our internal mmc:

sudo parted /dev/mmcblk0

mklabel gpt

quit

We will now create the partitions. Run the following commands:

sudo fdisk /dev/mmcblk0

And use the following options: n (to create a new partition), <hit enter> (to select default), <hit enter> (again to select default), +537M (to make it 537MB size), Y (to remove signature), t (to change partition type), 1 (to make it EFI)

We have created the EFI partition, but we'll be making 2 more, so we proceed as follow:

n, <hit enter>, <hit enter>, +1G (for boot partition)

n, <hit enter>, <hit enter>, <hit enter> (to use the entire left space as the LURK encrypted partition)

This is where the encrypted partition installation differs from a regular one. In a regular one you would create a swap partition instead of the boot one.

Finally use fdisk with "w" to apply changes an exit fdisk. Now we are going to format every partition with the corresponding format by running:

sudo mkfs.fat -F 16 /dev/mmcblk0p1 <--- For the first partition.

sudo mkfs.ext2 /dev/mmcblk0p2 <--- For the second

The third partition will be our encrypted LURK partition. After using the following command it will ask you for a password. Make sure to remember it.

sudo cryptsetup luksFormat --hash=sha512 --key-size=512 --cipher=aes-xts-plain64 --verify-passphrase /dev/mmcblk0p3

Next we'll open the encrypted partition:

sudo cryptsetup luksOpen /dev/mmcblk0p3 CryptDisk

Then it is suggested that we use the following command to completely erase every possible data left in this partition, which would increase our privacy and OPSEC. Depending on wether or not we have used the computer for a while with our personal info, it is up to you to decide it is worth it or not, since it will take quite some time. Although considering our mmc is so little, its probably not that bad.

sudo dd if=/dev/zero of=/dev/mapper/CryptDisk bs=4M

With the second to last command we run, we named our LURK partition as 'CryptDisk', and we use the following command to create virtual volume vg0:

sudo pvcreate /dev/mapper/CryptDisk

sudo vgcreate vg0 /dev/mapper/CryptDisk

Now we create both swap and the system's root partitions inside vg0. If we desire to have a separate home partition this is where we would do it. I have no idea how to choose a size for swap. I'm choosing to have it haft the size of ram, since it seems is the default size chosen by default on trisquel's installer.

sudo lvcreate -n swap -L 2G vg0

sudo lvcreate -n root -l +100%FREE vg0

We'll now format this partitions:

sudo mkfs.ext4 /dev/vg0/root

sudo mkswap /dev/vg0/swap

Use debootstrap to install trisquel

First will create a directory that we'll use to mount the root partition:

mkdir /mnt/deboot

And then we mount it:

sudo mount -t ext4 /dev/vg0/root /mnt/deboot

Finally we can make debootstrap do its magic on it:

sudo debootstrap --arch arm64 aramo /mnt/deboot https://mirror.fsf.org/trisquel/

You may chose a different mirror that is closer to you.

Preparing the chroot environment

Copy the mounted file systems table. It keeps the df command happy. (It will be overwritten upon boot.)

sudo cp /etc/mtab /mnt/deboot/etc/mtab

Until your new install is booting on it's own, we'll borrow these from the host.

sudo mount -o bind /dev /mnt/deboot/dev

sudo mount -o bind /proc /mnt/deboot/proc

sudo mount -o bind /sys /mnt/deboot/sys

Enter chroot

Enter trisquel’s just installed system with:

sudo chroot /mnt/deboot /bin/bash

Prepare apt

Add additional sources:

echo "deb https://mirror.fsf.org/trisquel aramo-security main" >> /etc/apt/sources.list

echo "deb https://mirror.fsf.org/trisquel aramo-updates main" >> /etc/apt/sources.list

Update and upgrade:

apt update

apt upgrade

Generate fstab

We are going to add some lines to /etc/fstab:

echo "/dev/mapper/vg0-root / ext4 errors=remount-ro 0 1" >> /etc/fstab

echo "/dev/mapper/vg0-swap none swap sw 0 0" >> /etc/fstab

For the root and the swap partitions, then:

echo "UUID=$(blkid -s UUID -o value /dev/mmcblk0p2) /boot ext2 defaults 0 2" >> /etc/fstab

echo "UUID=$(blkid -s UUID -o value /dev/mmcblk0p1) /boot/efi vfat umask=0077 0 1" >> /etc/fstab

For boot and efi partitions. We also need to add the LURKS encrypted partition to /etc/crypttab:

echo "CryptDisk UUID=$(blkid -s UUID -o value /dev/mmcblk0p3) none luks,discard" >> /etc/crypttab

Mount boot and efi

Mount boot partition:

mount /dev/mmcblk0p2 /boot

Now create an efi mount point in /boot:

mkdir /boot/efi

And mount efi partition in it:

mount /dev/mmcblk0p1 /boot/efi

Personalize your installation

Give a name to the computer:

echo "<name-your-host>" > /etc/hostname

Configure your locales. Make sure you select C.UTF-8 and UTF-8 along side any other you may want to add:

dpkg-reconfigure locales

Create a password for root.

passwd

Create a normal user:

adduser <your-user-name>

Install a kernel and desktop

Install a kernel with:

apt install linux-generic

I recommend mate against kde because of this computer’s memory:

apt install trisquel-base-recommended lightdm-gtk-greeter trisquel-base trisquel-recommended trisquel

But you could use this for kde:

apt install triskel triskel-recommended sddm trisquel-base trisquel-base-recommended

Grub Installation

Install the grub package:

apt install grub-efi-arm64

Next we are going to install grub in /boot and /boot/efi with removable media support and nvr off:

grub-install --boot-directory=/boot --efi-directory=/boot/efi --no-nvram –removable

Finally run:

update-grub

Run “exit” to leave chroot, and “exit” again to exit su --, and close the terminal window.

You are done

You should be able to reboot you system and boot trisquel. Normally libreboot gives you the SD card as the first entry, so the second entry has our trisquel boot.

References

This is where I found how to do things:

https://askubuntu.com/questions/991875/is-there-a-program-to-install-ubuntu-from-a-linux-system

https://askubuntu.com/questions/918021/encrypted-custom-install