

Clonezilla Live on RISC-V: Crafting Open Source Live Systems for Open Hardware

Steven Shiau, Ceasar Sun, Thomas Tsai

clonezilla.org

Apr 2025



Outline

- Introduction to Clonezilla
- RISC-V Architecture
- Clonezilla Live for RISC-V
- Future Outlook
- Q&A



Outline

- Introduction to Clonezilla
- RISC-V Architecture
- Clonezilla Live for RISC-V
- Future Outlook
- Q&A



System imaging and cloning - backup



You want to crash!!!
I show you how to crash!!!

image source: maggiesfarm.anotherdotcom.com
www.compsults.com, and jervisdabreo.com

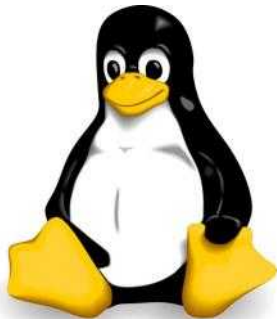


Massive system deployment



What is Clonezilla?

- A partition and disk imaging/cloning utility similar to True image® or Ghost®. GPL license
- **Clonezilla live** uses Debian/Ubuntu as underlying OS
- A bare metal recovery tool for



*1



*2



*3



*4

VMFS

VMware
ESX/ESXi

*5



*6

*Logo source: (1) Larry Ewing, Simon Budig and Anja Gerwinski, (2) Apple, (3) Microsoft, (4) Marshall Kirk McKusick, (5) VMWare (6) Distrowatch.com



Outline

- Introduction to Clonezilla
- RISC-V Architecture
- Clonezilla Live for RISC-V
- Future Outlook
- Q&A



RISC-V Architectures

- Open Source (Creative Commons or BSD).
Introduced in 2014
- 32-bit & 64-bit
- U. C.Berkeley -> RISC-V Foundation -> RISC-V International
- Vendors: SiFive, Andes, Ventana...
- Linux kernel support was from V5.17 in 2022.
Debian Supports riscv64 started in 2023.



Images source: wikipedia.org



Outline

- Introduction to Clonezilla
- RISC-V Architecture
- Clonezilla Live for RISC-V
- Future Outlook
- Q&A



Clonezilla Live for RISC-V

- Live system based on Debian Linux. Use Live-build from Debian to create it.
 - riscv64 still needs to be patched
 - https://salsa.debian.org/live-team/live-build/-/merge_requests/383
- Experimental releases are available
 - <https://free.nchc.org.tw/clonezilla-live/experimental/>



Progress about Clonezilla Live for RISC-V

- Clonezilla live riscv64 release started from 2024/Q4
- Current status :
 - Works well with Emulator (QEMU)
 - Trying to adapt to the physical machine
 - DC-ROMA RISC-V LAPTOP II
 - Two methods
 - U-boot -> EFI
 - U-boot direct boot



Image source: store.deepcomputing.io



Demo Clonezilla Live RISC-V @ QEMU

- Version : 3.2.1-25 ([Download](#))
- Related tools : [opensbi](#)、 u-boot-qemu 、 [guestfs-tools](#)
- Repack as a Clonezilla raw image by guestfs-tools
 - `$ virt-make-fs --verbose --partition=gpt --type=vfat imagebuild/ clonezilla-rv64.img`
- Startup in Qemu RISCV64 Emulator
 - `$ qemu-system-riscv64 -nographic -smp cpus=4 -machine virt -m 4.0G \`
`-bios /usr/lib/riscv64-linux-gnu/opensbi/generic/fw_jump.elf \`
`-kernel /usr/lib/u-boot/qemu-riscv64_smode/uboot.elf \`
`-object rng-random,filename=/dev/urandom,id=rng0 -device virtio-rng-device,rng=rng0 \`
`-append "console=ttyS0 rw root=/dev/vda1" \`
`-drive file=clonezilla-r.img,format=raw,if=virtio \`
`-device virtio-net-device,netdev=usernet -netdev user,id=usernet,hostfwd=tcp::22222-:22`



load U-boot ⇒ uEFI

```
U-Boot 2022.01+dfsg-2ubuntu2.6 (Aug 30 2024 - 03:32:40 +0000)
```

```
CPU: rv64imafdcsu
Model: riscv-virtio,qemu
DRAM: 6 GiB
Flash: 32 MiB
Loading Environment from nowhere... OK
In: uart@10000000
Out: uart@10000000
Err: uart@10000000
Net: eth0: virtio-net#1
Hit any key to stop autoboot: 0
```

```
Device 0: 1a4f VirtIO Block Device
Type: Hard Disk
Capacity: 463.5 MB = 0.4 GB (949357 x 512)
... is now current device
Scanning virtio 0:1...
** Unable to read file / **
Failed to load '/'
libfdt fdt_check_header(): FDT_ERR_BADMAGIC
Scanning disk virtio-blk#9...
Scanning disk virtio-blk#10
```

```

Boot HART ID : 0
Boot HART Domain : root
Boot HART Priv Version : v1.12
Boot HART Base ISA : rv64imafdc
Boot HART ISA Extensions : 4, time, sstc
Boot HART PMP Count : 16
Boot HART PMP Granularity : 4
Boot HART PMP Address Bits: 54
Boot HART MHMP Count : 16
Boot HART MIDELEG : 0x00000000000001666
Boot HART MEDELEG : 0x0000000000000509

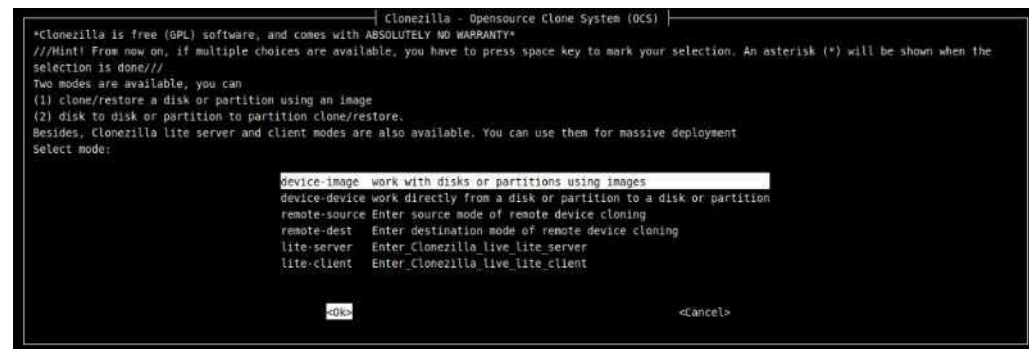
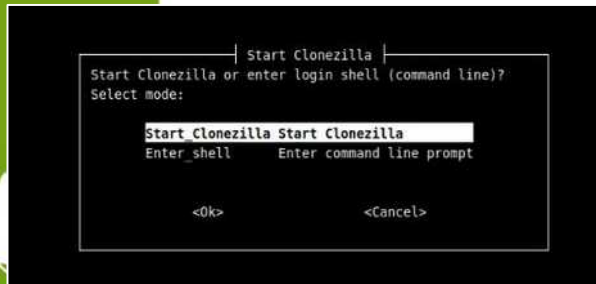
```

Use the **↑** and **↓** keys to select which entry is highlighted.
Press **enter** to boot the selected OS, or **o** to edit the commands before booting or **c** for a command-line.
The highlighted entry will be executed automatically in 30s.

[illegible]

Screenshot of Clonezilla live RISC-V @ QEMU (Continued)

Run Clonezilla Main Procedure



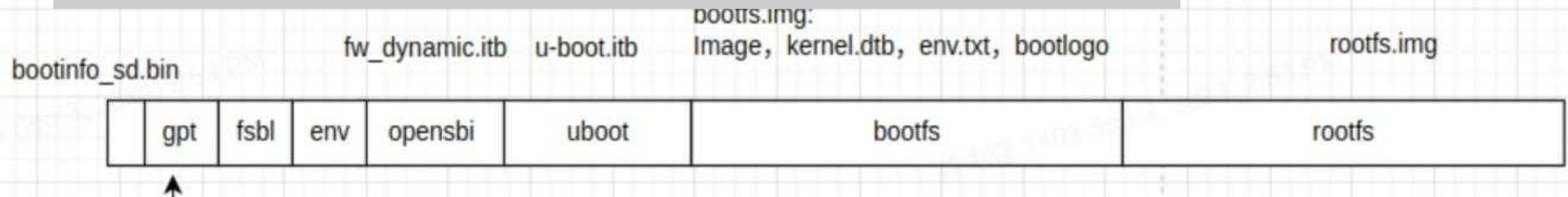
Clonezilla Live RISC-V @ Hardware

- Hardware : DC-ROMA RISC-V LAPTOP II
 - Octa-core RISC-V CPU : RISCV64 8-core CPU
 - SpacemiT K1 SoC
- Used Clonezilla Version : 3.2.1-21 ([Download](#))
- Related tools/projects :
 - [DC-ROMA_Gen2_LAPTOP_K1_RV-L2A](#)
 - [Bianbu Linux](#)
 - [u-boot](#)、 [opensbi](#) (actually to use : [Bianbu@gitee](#))
- We tried 2 booting progress to be solution :
 - W1 : u-boot → uEFI → GRUB Menu → Boot-up
 - W2 : u-boot → direct to load kernel/initrd → Boot-up

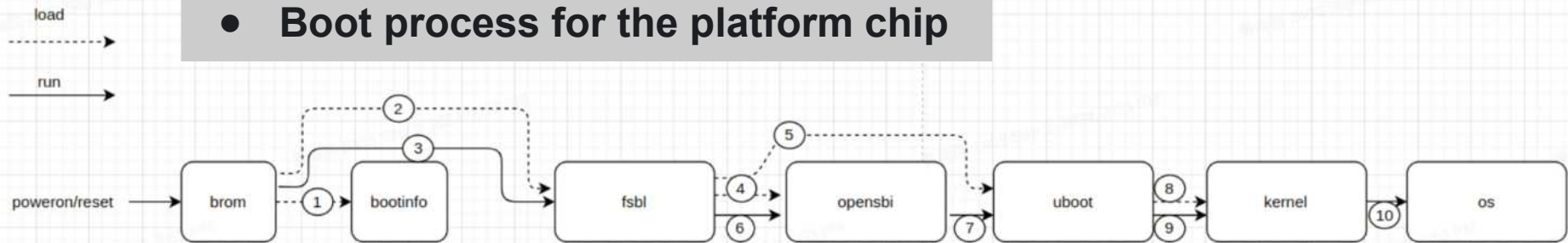


Quick Introduction of K1 SoC Boot Procedure

Firmware layouts for the K1 series SoC of SD Card



Boot process for the platform chip



- We have to consider :
 - Rebuild [u-boot](#) / [opensbi](#) (based on : [Bianbu@gitee](#))
 - Configure u-boot procedure to be more flexible

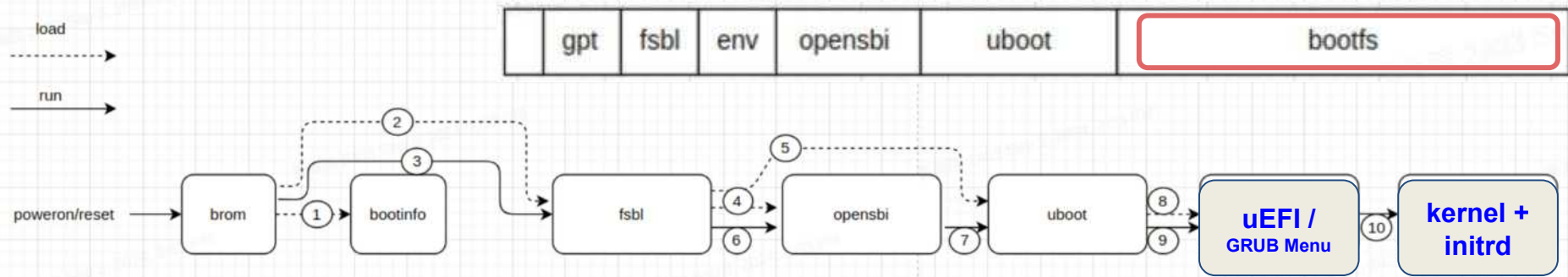
Reference: <https://bianbu-linux.spacemit.com/en/device/boot/>



Hardware :

W1 : u-boot → uEFI → GRUB Menu → Boot-up

- Modify boot process

a. Rebuild [u-boot](#) / [opensbi](#) →

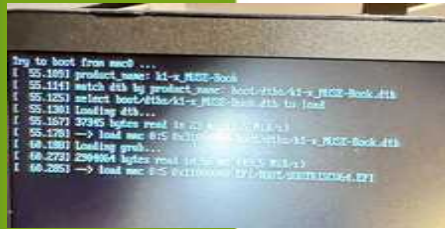
- Enable uEFI function / u-boot boot menu (*p4 + *p2)
- Enhance u-boot script (*p2) : to allow assign parameters to u-boot by configure file (plain text)
- Use fw_dynamic.its as opensbi firmware (*p3)
 - fw_dynamic.its 、 fw_jump.bin 、 fw_payload.bin
- Replace partition data (*p4, *p2, *p3) by dd

b. Adjust [bootfs](#) partition for EFI booting flag

- EFI boot flag / Convert to VFAT

c. Unpack Clonezilla ZIP into [bootfs](#)

Screenshot



- Full record video
 - https://youtu.be/oS_XrUQVTjc



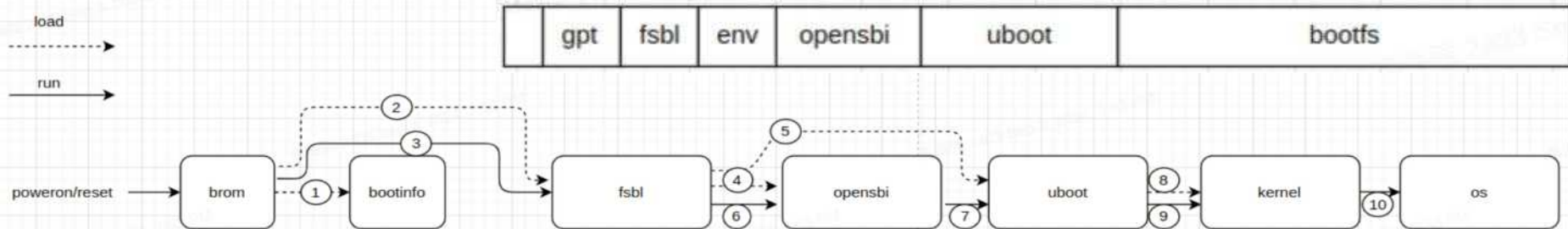
- Status:
 - Success to boot via BOOTRISCV64.EFI / grub.cfg
 - MMC block device don't detect
 - **System panic** at initramfs mount root sub function
 - Hard drive (e.g., mmcblk) required vender's drivers under initramfs stage



Hardware :

W2 : u-boot → kernel+initrd → Boot-up

- Use u-boot → bootm (load kernel+initrd) → bootup



- Based on [Bianbu linux](#) Project
- Use the Bianbu linux kernel/firmwares for clonezilla live
 - Based on clonezilla live RISC-V 64 image (v3.2.1-21)
 - Remaster it with [Spacemit-k1 linux kernel](#) / [k1x-vpu-firmware](#) debs by ocs-live-swap-kernel command (+ *manual patches*)
- Replace kernel / initrd / boot parameters
 - ~(*mmcblk0p5)/env_k1-x.txt (details on next page)



U-Boot Booting Procedure Enhancement

Enables flexible configuration of Clonezilla live boot parameters

- Re generate u-boot env.bin and replace ***p2**
 - `~/uboot-2022.10$ ./tools/mkenvimage -s 0x4000 -o ./ocs-u-boot-env-default.bin ./ocs-patch-u-boot-env-default.txt`
 - `sudo dd if=ocs-u-boot-env-default.bin of=/dev/mmcblk0p2 conv=fsync`
- Edit u-boot environment file : `~(*p5)/env_k1-x.txt`

```
# File : env_k1-x.txt
knl_name=vmlinuz-6.6.36
ramdisk_name=initrd-dev-ocs_re.img
```

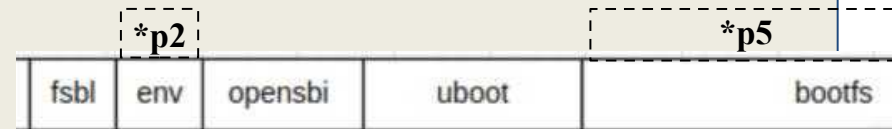
```
dtb_dir=spacemit/6.6.63
console=
loglevel=6
```

```
# Clonezilla live default
```

```
cmdlineargs=boot=live union=overlay username=user config components ocs_1_cpu_udev
noswap edd=on nomodeset enforcing=0 noeject locales="en_US.UTF-8" keyboard-layouts=en
ocs_live_run="ocs-live-general" ocs_live_extra_param="-a -b azsx -cc dfsd" ocs_live_batch="no"
vga=788 net.ifnames=0 nosplash i915.blacklist=yes radeonhd.blacklist=yes nouveau.blacklist=yes
vmwgfx.enable_fbdev=1
```

```
## Clonezilla live to RAM
```

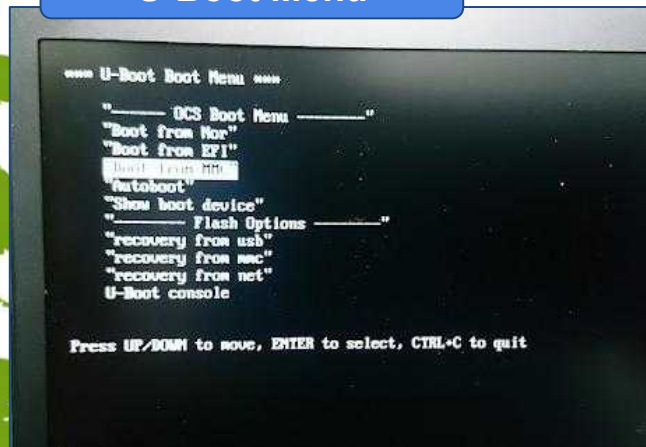
```
cmdlineargs=boot=live (... same as the above ...) ocs_live_run="ocs-live-general"
ocs_live_extra_param="" ocs_live_batch="no" toram=live,EFI,boot,disk ( ... same as the above...)
```



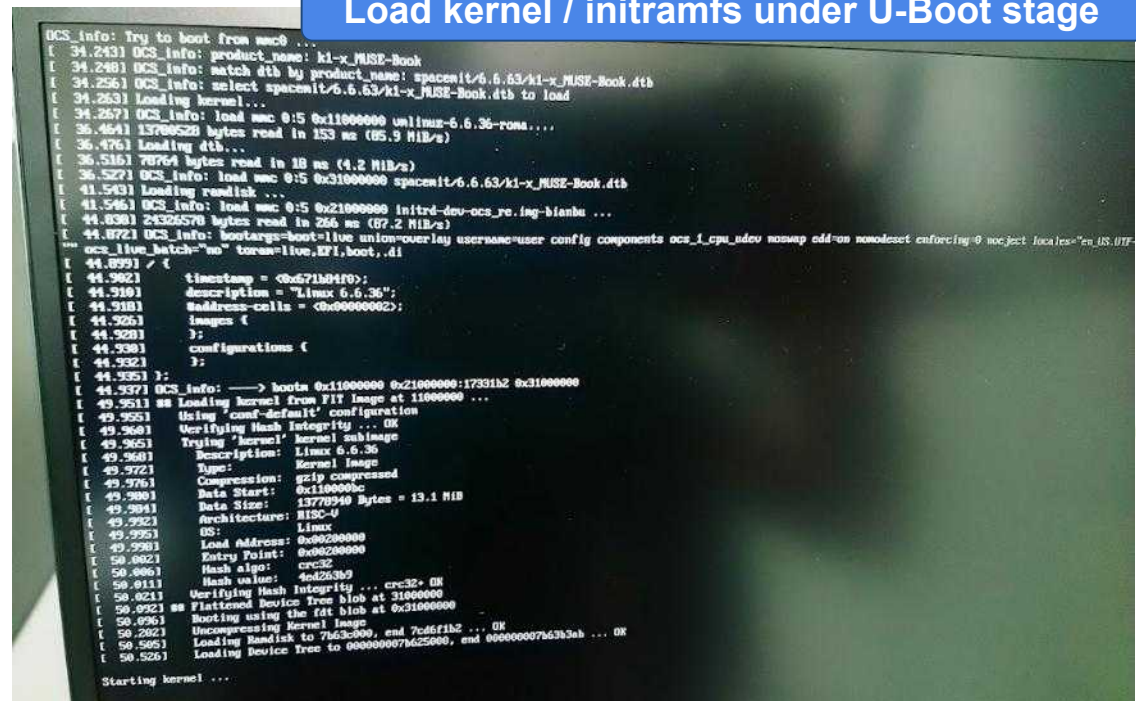
W2 Screenshots

- U-Boot Menu → Parse boot parameters
- Bootm (load kernel+initrd) → bootup

U-Boot Menu



Load kernel / initramfs under U-Boot stage



```

PS. Next time you can run this command directly:
/usr/sbin/ccs-sr -g2 -c -j2 -x2p -l 2348 -sfack -senc -plu -p choose savedid:2925-04-23-01-Img nvme0n1
This command is also saved as this file name for later use if necessary: /tmp/ccs-2925-04-23-01-Img-2925-04-23-01-29
=====
Press "Enter" to continue...
Activating the partition info in /proc... done!
Selected device [nvme0n1] found!
The selected devices: nvme0n1
Searching for data/snap/extended partition(s)...
Finding all disks and partitions...
Excluding busy partition.....
Excluding linux raid member partition.....
Unaccounted partitions (including extended or snap): nvme0n1p1 nvme0n1p2 nvme0n1p3 nvme0n1p4 nvme0n1p5 nvme0n1p6
Collecting info..... done!
The data partition to be saved: nvme0n1p1 nvme0n1p2 nvme0n1p3 nvme0n1p4 nvme0n1p5 nvme0n1p6
Activating the partition info in /proc... done!
Selected device [nvme0n1p1] found!
Selected device [nvme0n1p2] found!
Selected device [nvme0n1p3] found!
Selected device [nvme0n1p4] found!
Selected device [nvme0n1p5] found!
Selected device [nvme0n1p6] found!
The selected devices: nvme0n1p1 nvme0n1p2 nvme0n1p3 nvme0n1p4 nvme0n1p5 nvme0n1p6
=====
The following step is to save the hard disk/partition(s) on this machine as an image:
=====
Machine: Unknown product name
nvme0n1 (192408) [1TB_SSD] [pci-0002_01_00_0-nvme-1] [C2737N2100035]
nvme0n1p1 (256K) [fshl] [in_1TB_SSD] [pci-0002_01_00_0-nvme-1] [C2737N2100035]
nvme0n1p2 (64K) [env] [in_1TB_SSD] [pci-0002_01_00_0-nvme-1] [C2737N2100035]
nvme0n1p3 (128) [boot] [in_1TB_SSD] [pci-0002_01_00_0-nvme-1] [C2737N2100035]
nvme0n1p4 (256) [boot] [in_1TB_SSD] [pci-0002_01_00_0-nvme-1] [C2737N2100035]
nvme0n1p5 (256K) [ext4] [bootfs] [in_1TB_SSD] [pci-0002_01_00_0-nvme-1] [C2737N2100035]
nvme0n1p6 (953.66) [ext4] [rootfs] [in_1TB_SSD] [pci-0002_01_00_0-nvme-1] [C2737N2100035]
=====
-> /home/partimg/2925-04-23-01-Img...
Are you sure you want to continue? (y/n) _

```

```

Partitione-
Reading Super Block
Calculating bitmap... Please wait...
done!
File system: EXTFS
Device size: 268.4 MB = 65536 Blocks
Space in use: 101.9 MB = 24880 Blocks
Free Space: 166.5 MB = 40656 Blocks
Block size: 4096 Byte
Syncing... OK)
Partition successfully cloned the device (/dev/nvme0n1p5) to
the image (-).

Total Time: 00:00:02 Remaining: 00:00:00
Ave. Rate: 3.06GB/min

Data Block Process:
██████████████████████████████████████ 100.00%

Total Block Process:
██████████████████████████████████████ 100.00%
```

```

Partclone
Partclone v0.3.36 http://partclone.org
Starting to clone device (/dev/nvme0n1p6) to image (-)
Reading Super Block
Calculating bitmap... Please wait...
done!
File system: EXTFS
Device size: 6.4 GB = 1572864 Blocks
Space in use: 4.5 GB = 1107303 Blocks
Free Space: 1.9 GB = 465561 Blocks
Block size: 4096 Byte

Elapsed: 00:00:06 Remaining: 00:02:12 Rate: 1.97GB/min
Current Block: 47964 Total Block: 1572064

Data Block Process:
 4.33%

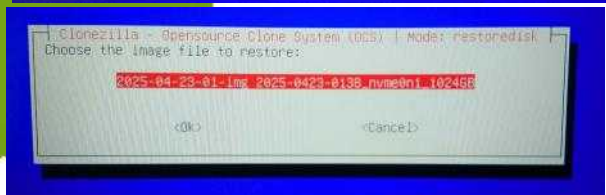
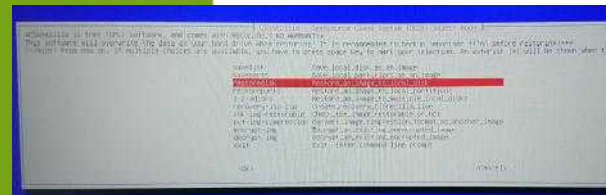
Total Block Process:
 3.05%

```

[illegible]

W2 : Screenshots of Restore

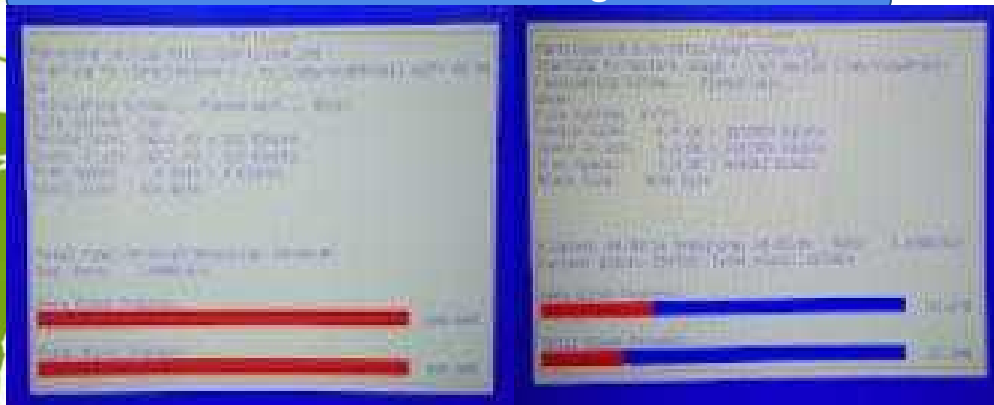
Restore



Ready to Restore



Partition restoring



Post-resizing



Image fiels for DC- ROMA L2A

Name	Size	Type
2025-04-23-DC-Roma_L2A	30 items	Folder
blkdev.list	1.4 kB	Text
blkid.list	1.4 kB	Text
clonezilla-img	13.3 kB	Text
dev-fs.list	287 bytes	Text
disk	8 bytes	Text
Info-dmi.txt	134 bytes	Text
Info-img-id.txt	236 bytes	Text
Info-img-size.txt	58 bytes	Text
Info-lshw.txt	18.3 kB	Text
Info-lspci.txt	653 bytes	Text
Info-OS-prober.txt	481 bytes	Text
Info-packages.txt	197 bytes	Text
Info-saved-by-cmd.txt	104 bytes	Text
Info-smart.txt	3.0 kB	Text

nvme0n1-chs.sf	37 bytes	Text
nvme0n1-gpt.gdisk	17.9 kB	Binary
nvme0n1-gpt.sgdisk	872 bytes	Text
nvme0n1-gpt-1st	17.4 kB	Binary
nvme0n1-gpt-2nd	16.4 kB	Binary
nvme0n1-mbr	512 bytes	Binary
nvme0n1p1.dd-ptcl-img....	110.3 kB	Audio
nvme0n1p2.dd-ptcl-img....	2.2 kB	Audio
nvme0n1p3.dd-ptcl-img....	128.9 kB	Audio
nvme0n1p4.dd-ptcl-img....	674.0 kB	Audio
nvme0n1p5.ext4-ptcl-im...	28.7 MB	Audio
nvme0n1p6.ext4-ptcl-im...	1.5 GB	Audio
nvme0n1-pt.parted	646 bytes	Text
nvme0n1-pt.parted.comp...	564 bytes	Text
nvme0n1-pt.sf	1.1 kB	Text
parts	60 bytes	Text



Conclusion for Clonezilla live RISC-V64

- What we do
 - U-Boot Booting Procedure Enhancement
 - Deploy Clonezilla to real RISC-V 64 machine: DC Roma , module : L2A
 - A successful use case to adopt vender's project (e.g., Bianbu Linux) kernel / drivers /firmware into original Clonezilla live
- What we release:
 - Original Clonezilla RISC-V64 version [\[here\]](#) [Wikipedia](#)
 - Clonezilla live image for [DC-ROMA L2 \[here\]](#)
 - Include : fsbl /u-boot / Clonezilla live customized image
 - Please read README.txt
 - An applicable Clonezilla Ubuntu image for DC-ROMA L2A [\[here\]](#)



Analysis

- Clonezilla live RISC-V64 arch
 - Still in early stage
 - Emulator works well but it still lacks drivers from Debian for physical machine...
 - It's still a pain to deal with booting for physical machine
 - Customized Linux kernel might be necessary
 - Debian/Ubuntu supports RISC-V boards
 - SiFive HiFive Unleashed, Allwinner Nezha, and StarFive VisionFive...
 - Desktop and server running RISC-V are not so common as ARM64 or X86-64. Use cases for Debian/Ubuntu, hence Clonezilla live are less



Image source: crowdsupply.com; linux-sunxi.org; starfivetech.com



Outline

- Introduction to Clonezilla
- RISC-V Architecture
- Clonezilla Live for RISC-V
- Future Outlook
- Q&A



Future Outlook

- Improvements and enhancements
 - Along with the underlying OS: Debian Linux
 - Massive deployment for RISC-V64
- The rise of RISC-V-64
 - Cost/Access: RISC-V's openness lowers hardware costs , making it viable for budget-strapped schools.
 - Power Efficiency: RISC-V in edge



Clonezilla

Demo about Clonezilla live RISC-V64



Questions ?

Great!



?????

