M1120 Reliability Test Setup

for DV stage - WYHQ/Reliability use only

Hugo / Reliability Engineering

2023/11/27

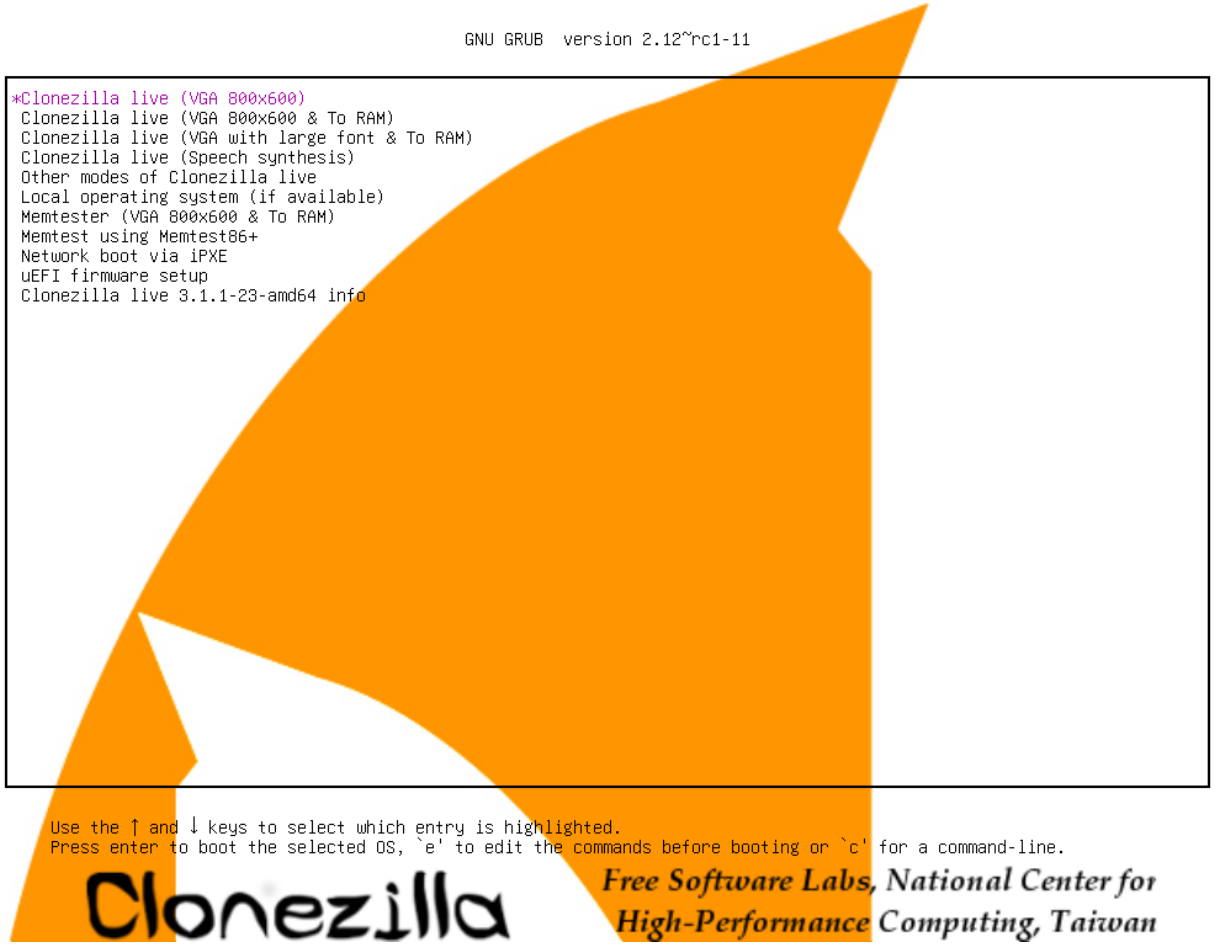
Table of Contents

Clone Linux OS

with Clonezilla expert mode

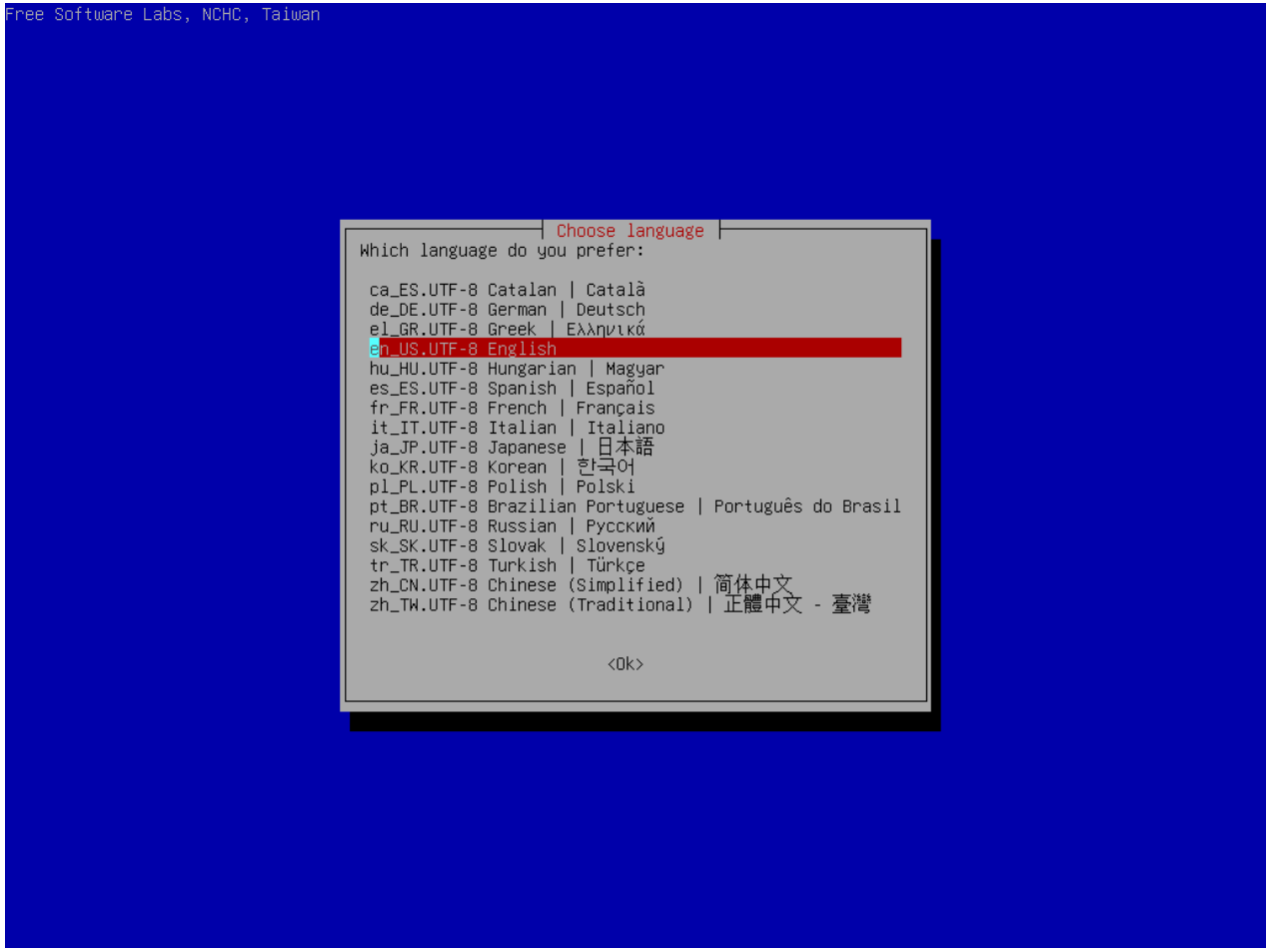
Clone Linux OS

1) Boot into Clonezilla through USB (you can download the image from here)



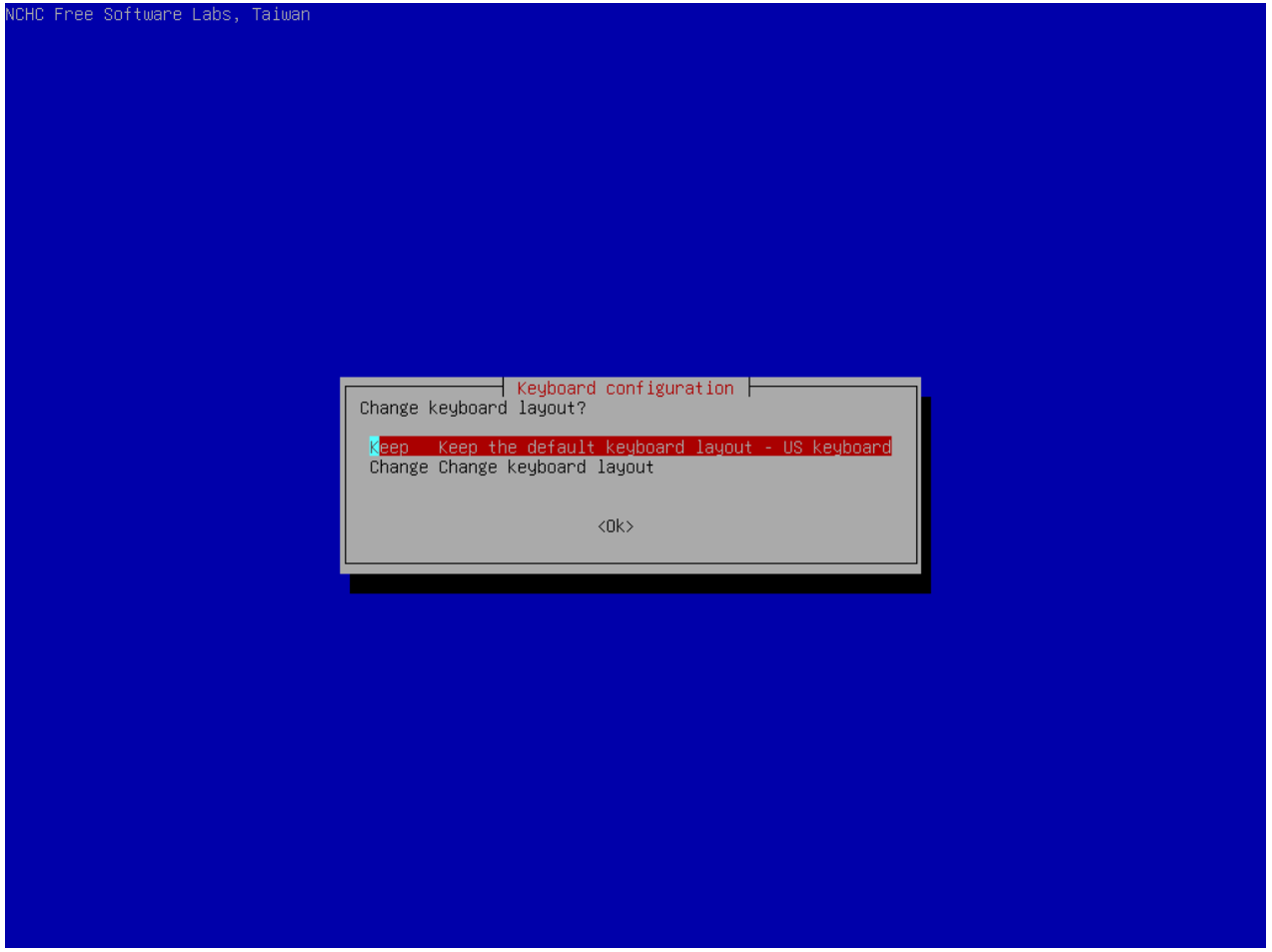
Clone Linux OS

2) Choose your preferred language



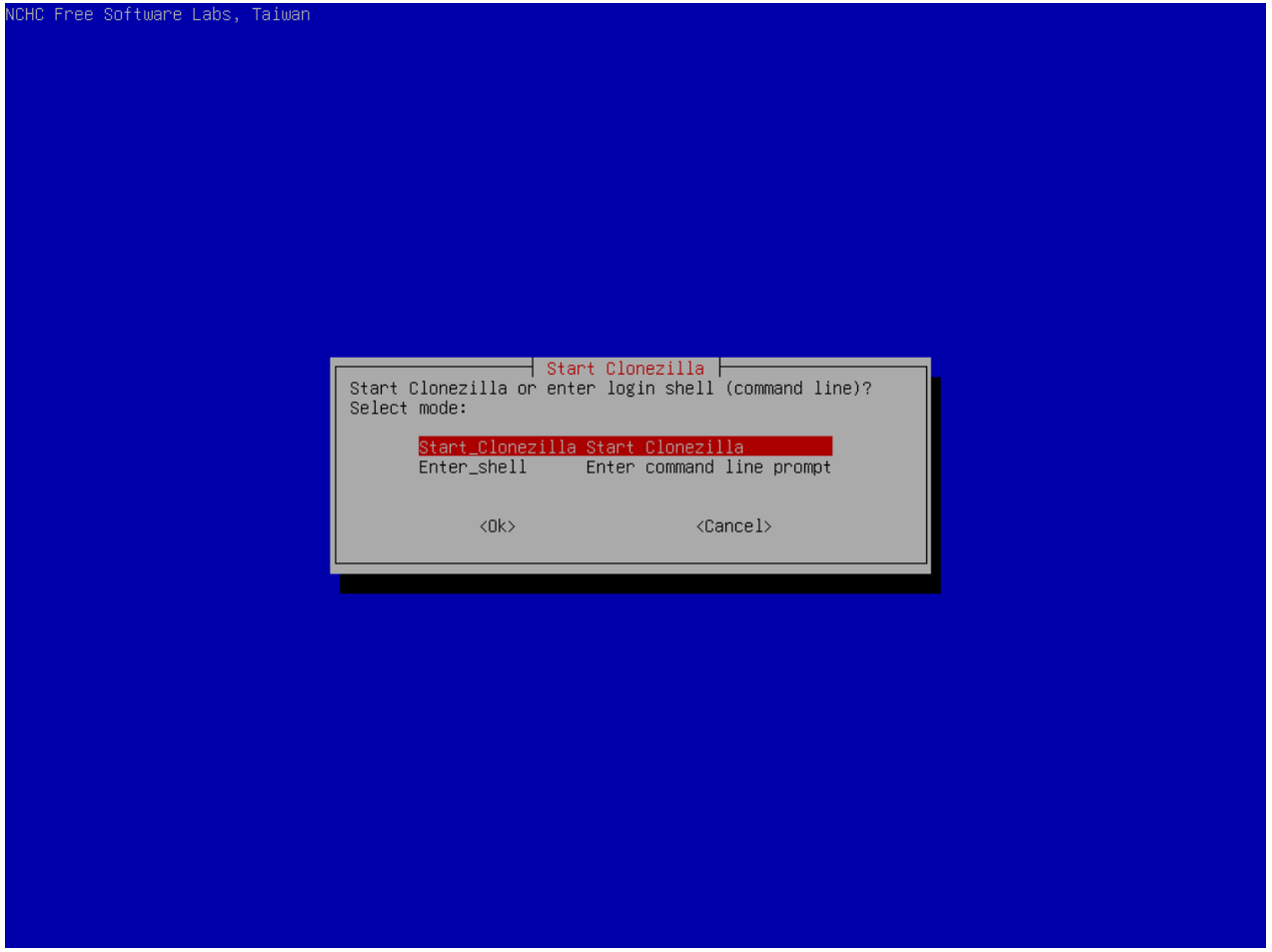
Clone Linux OS

3) Choose your keyboard layout



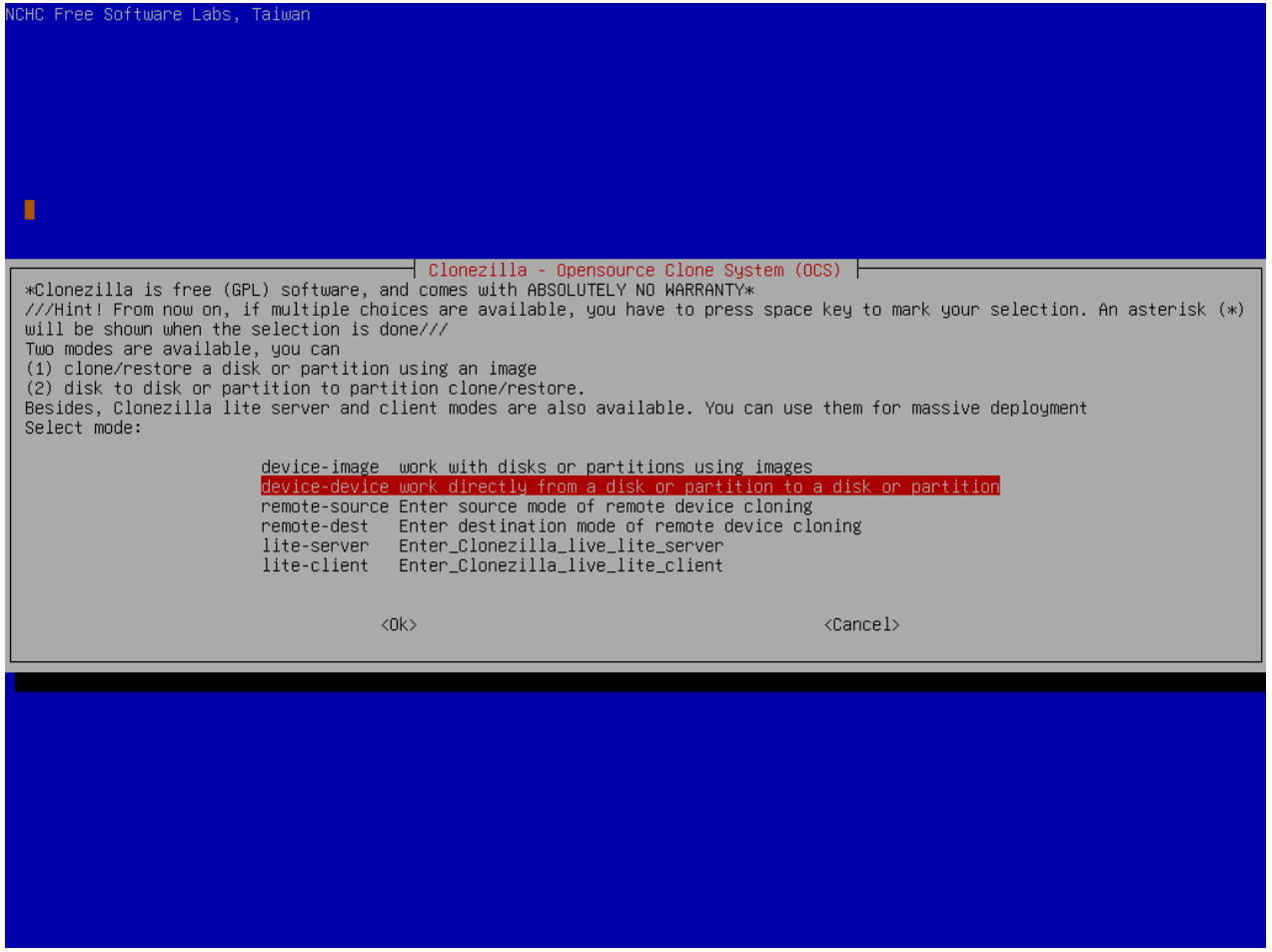
Clone Linux OS

4) Start Clonezilla



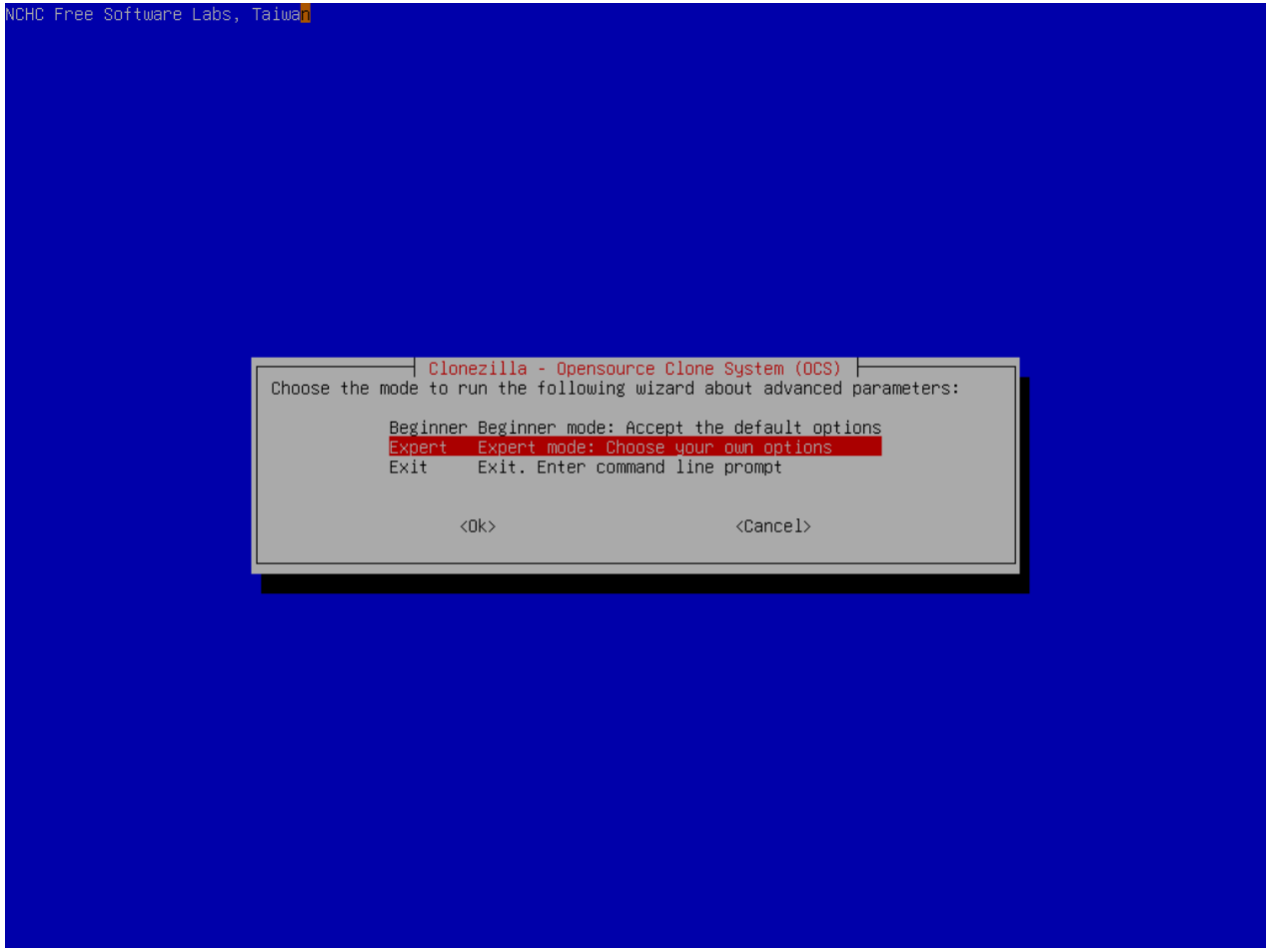
Clone Linux OS

5) Choose ‘device-device’ mode



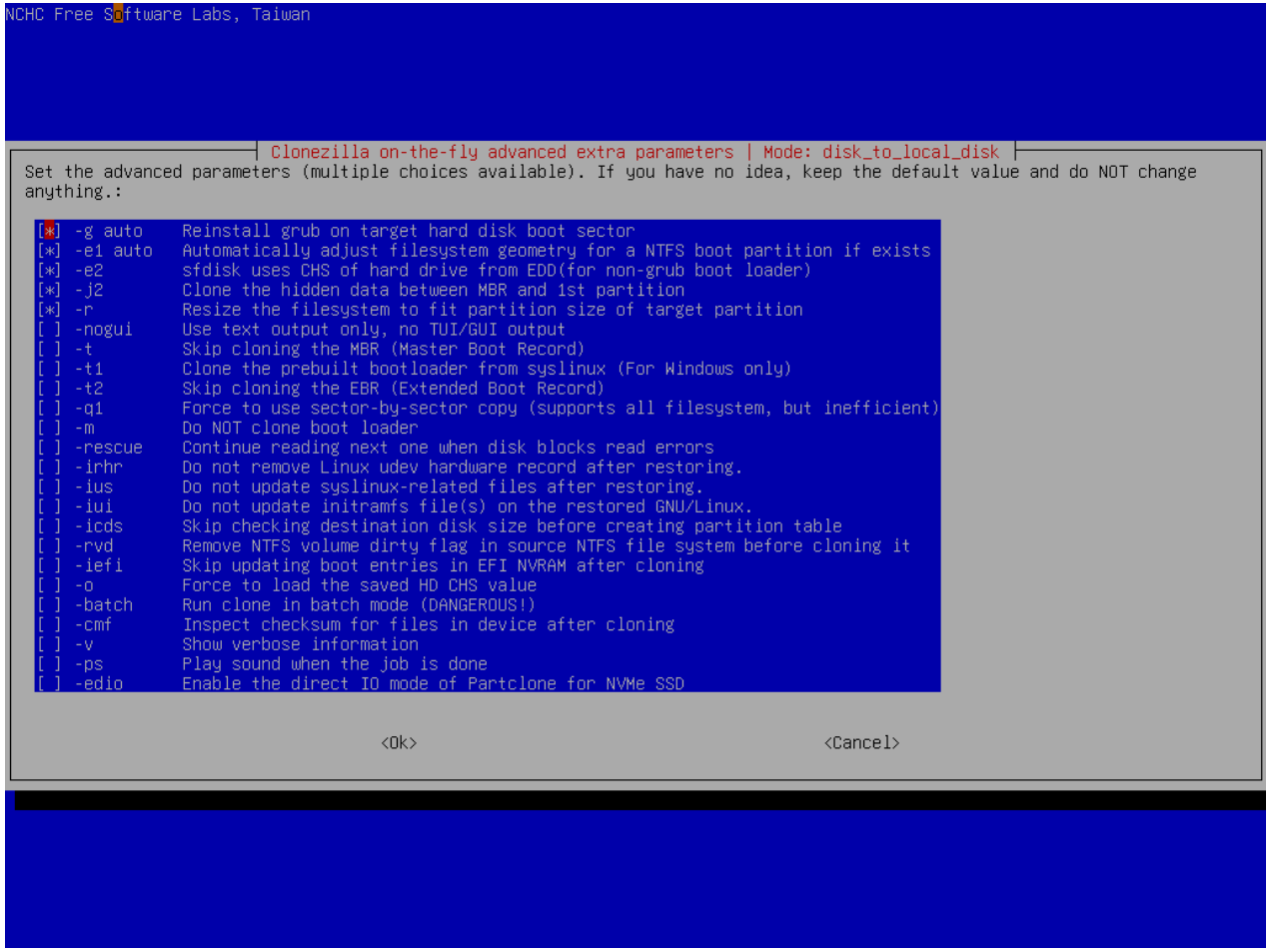
Clone Linux OS

6) Choose ‘Expert’ mode



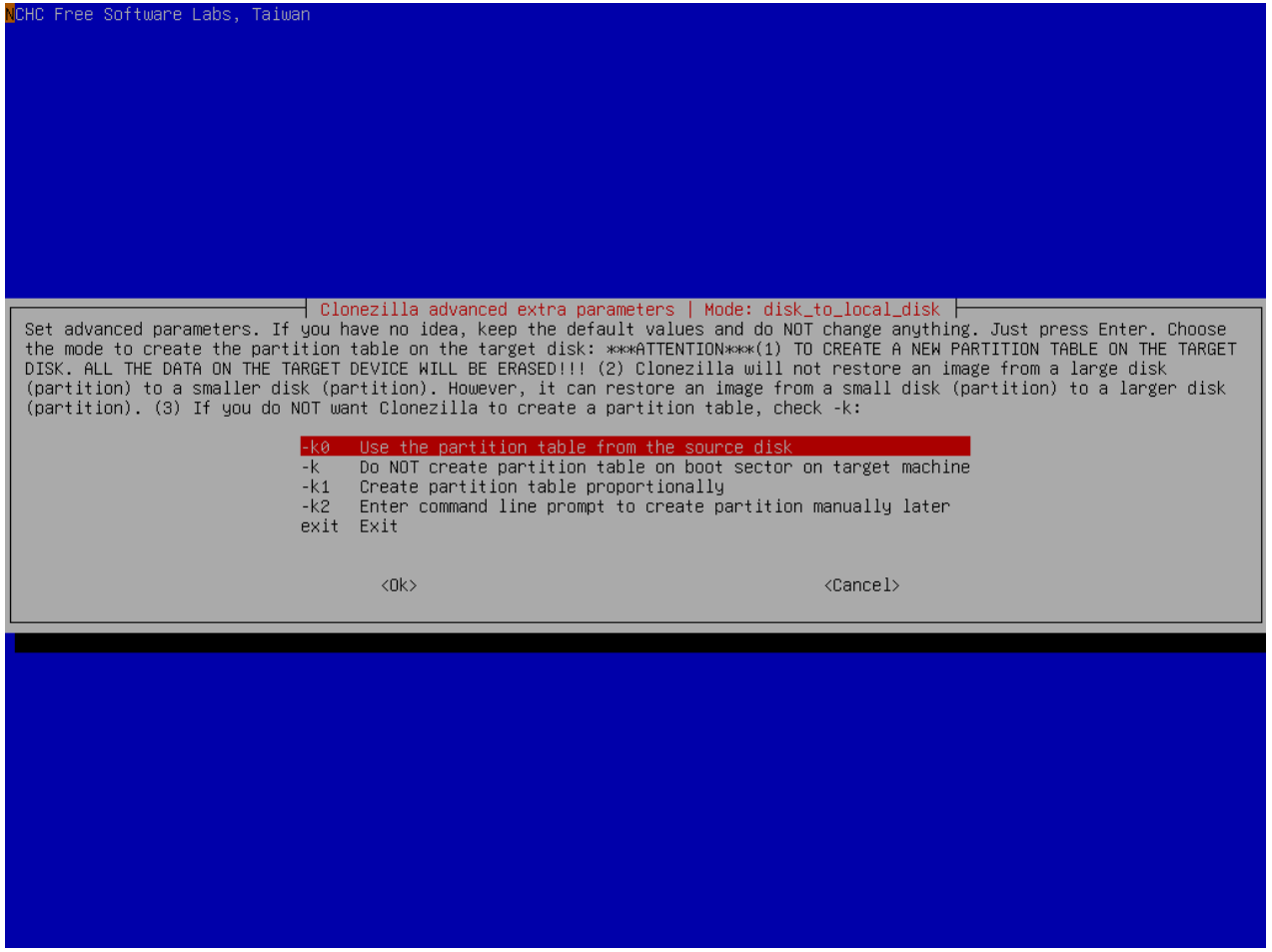
Clone Linux OS

7) Unselect ‘-g auto’ and Select ‘-q1’ then continue



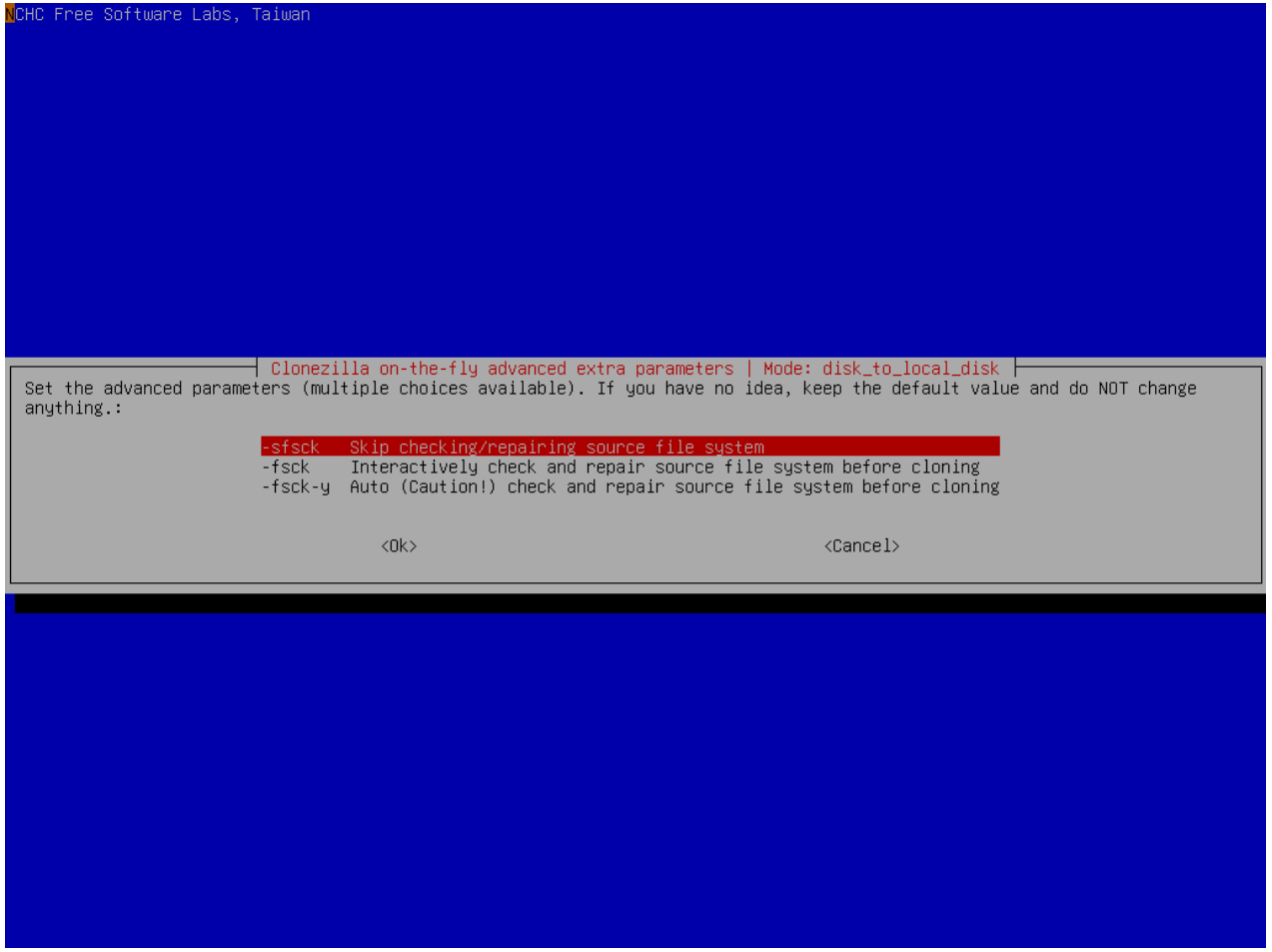
Clone Linux OS

8) Choose '-k0' Use the partition table from the source disk



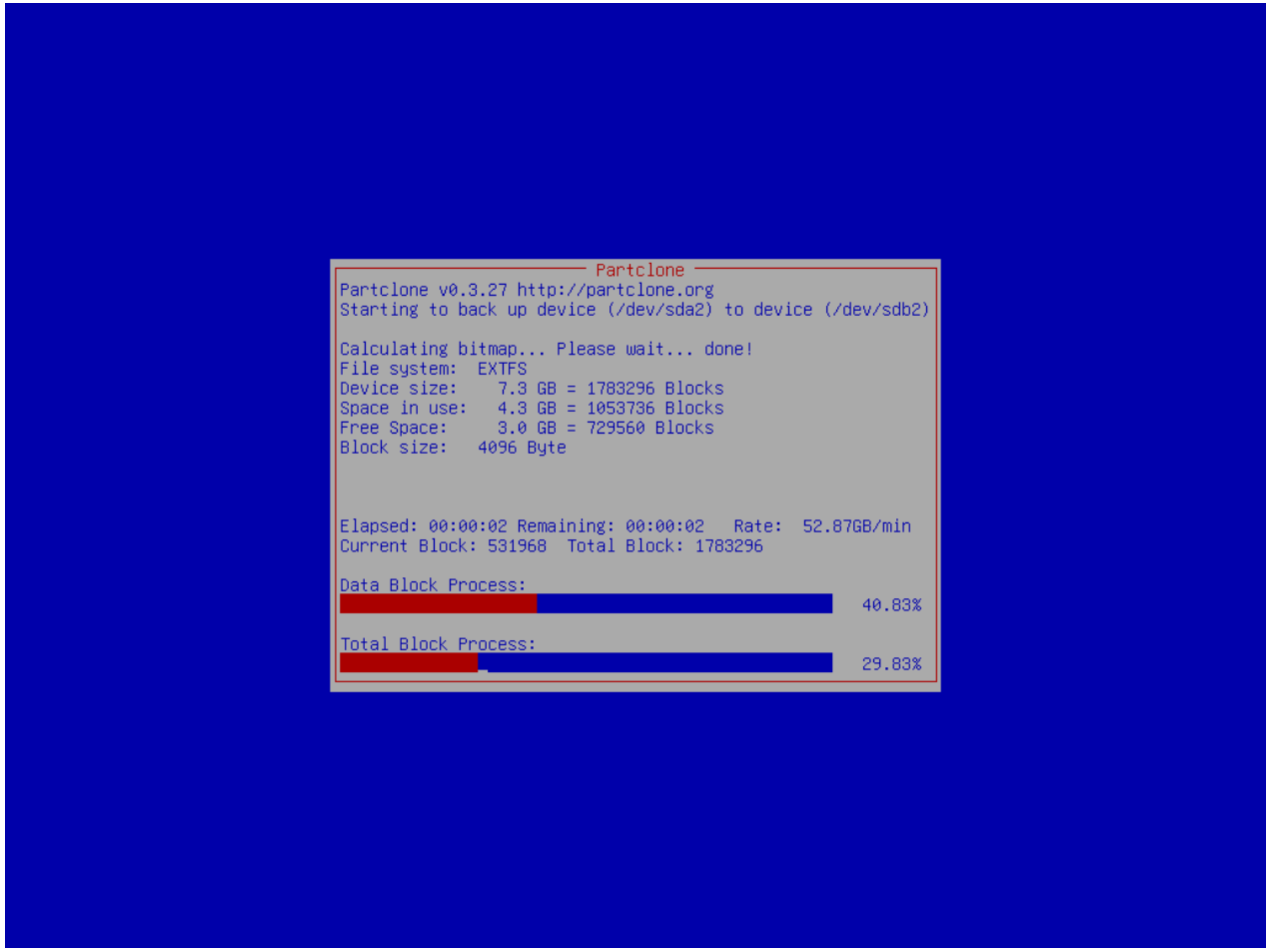
Clone Linux OS

9) Choose '-sfsck' Skip checking/repairing source file system



Clone Linux OS

10) Wait for it to complete and enjoy your OS

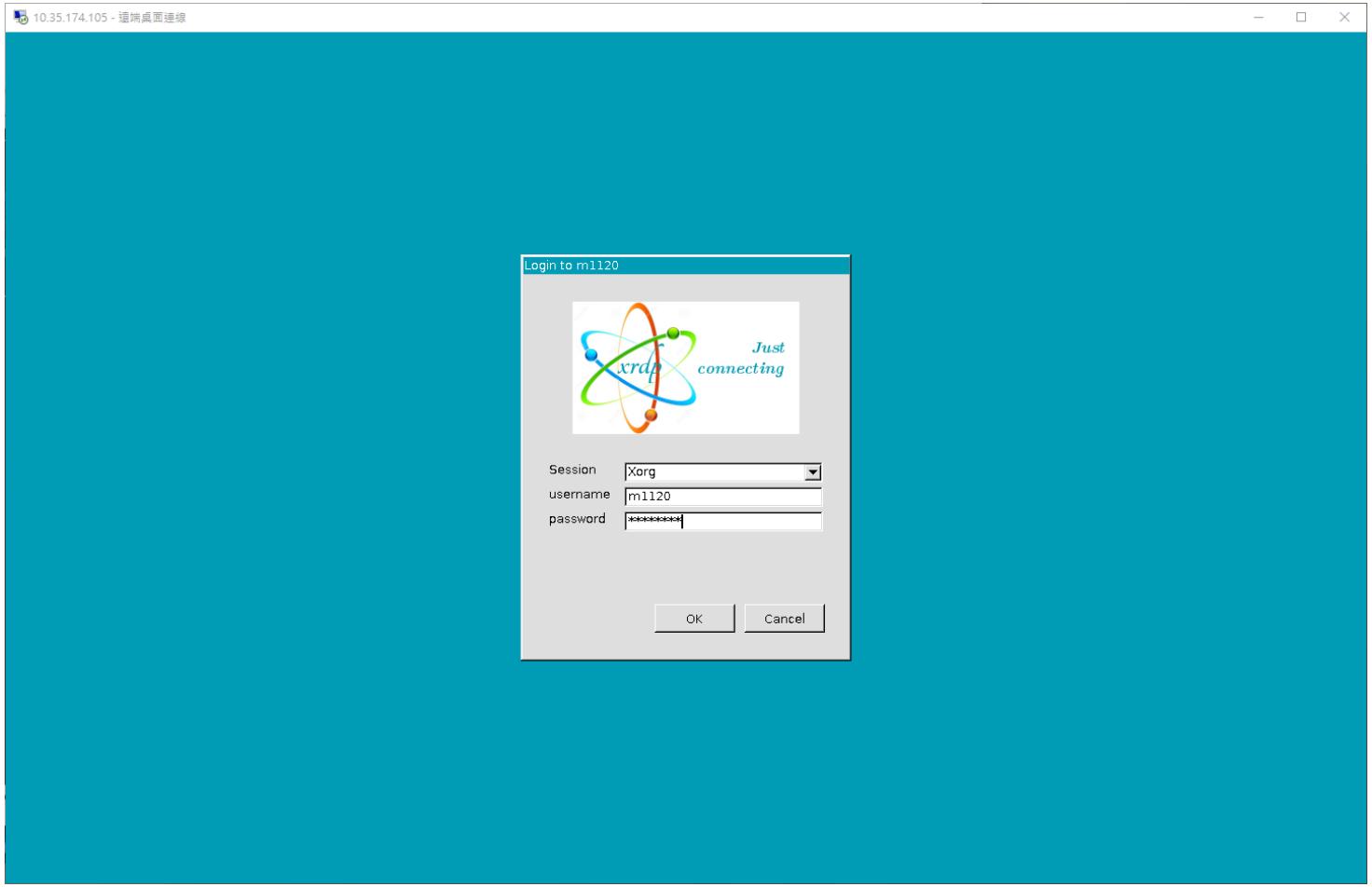


Change Hostname

preparing for remote control

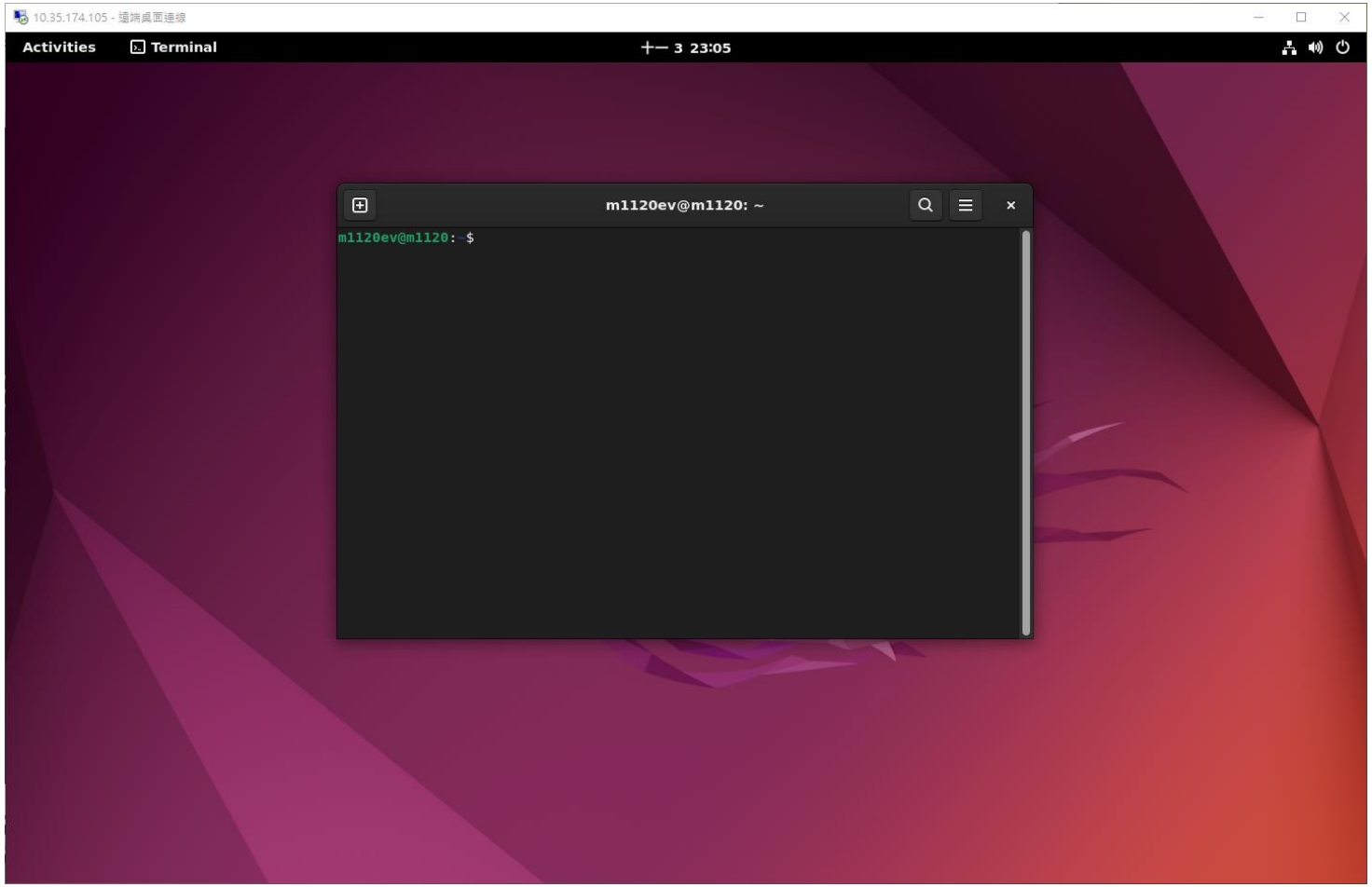
Change Hostname

1) Use remote desktop to login to Ubuntu [m1120/p@ssw0rd]



Change Hostname

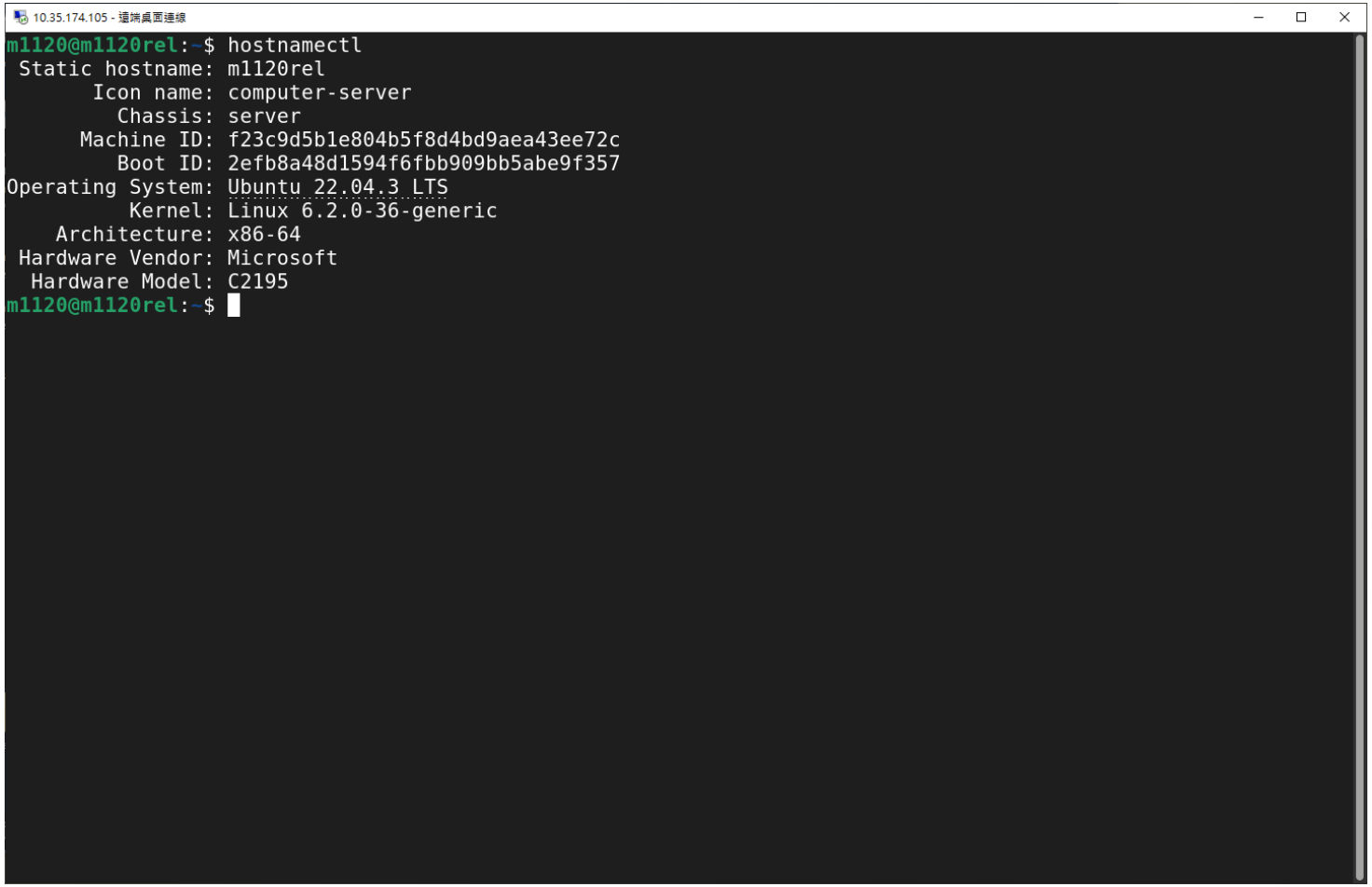
2) Press Ctrl+T to open a terminal window



Tips:
F11: maximize current window
Ctrl/Shift/+: increase text size
Ctrl/-: decrease text size

Change Hostname

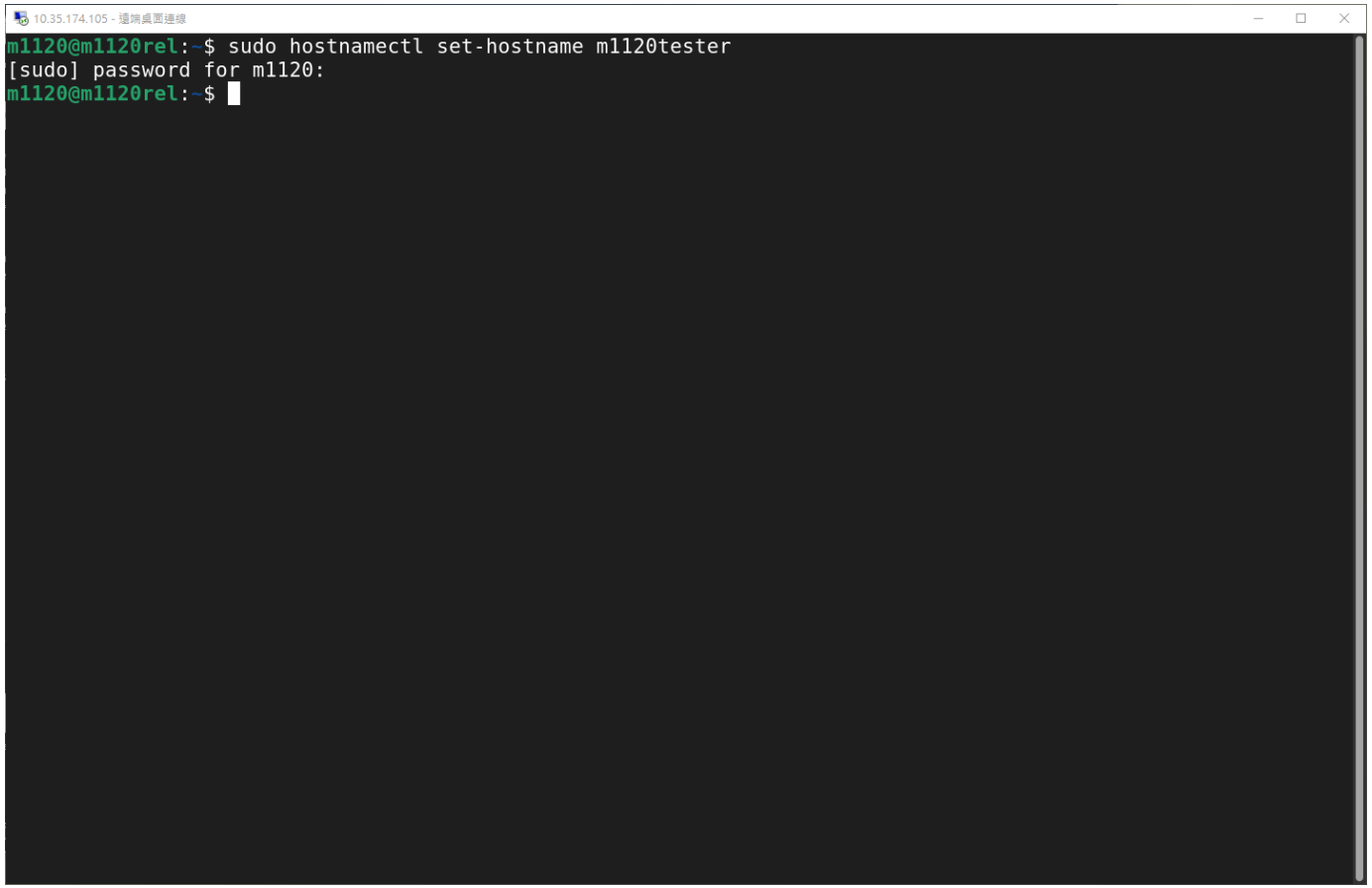
3) Type **hostnamectl** to check the hostname



Change Hostname

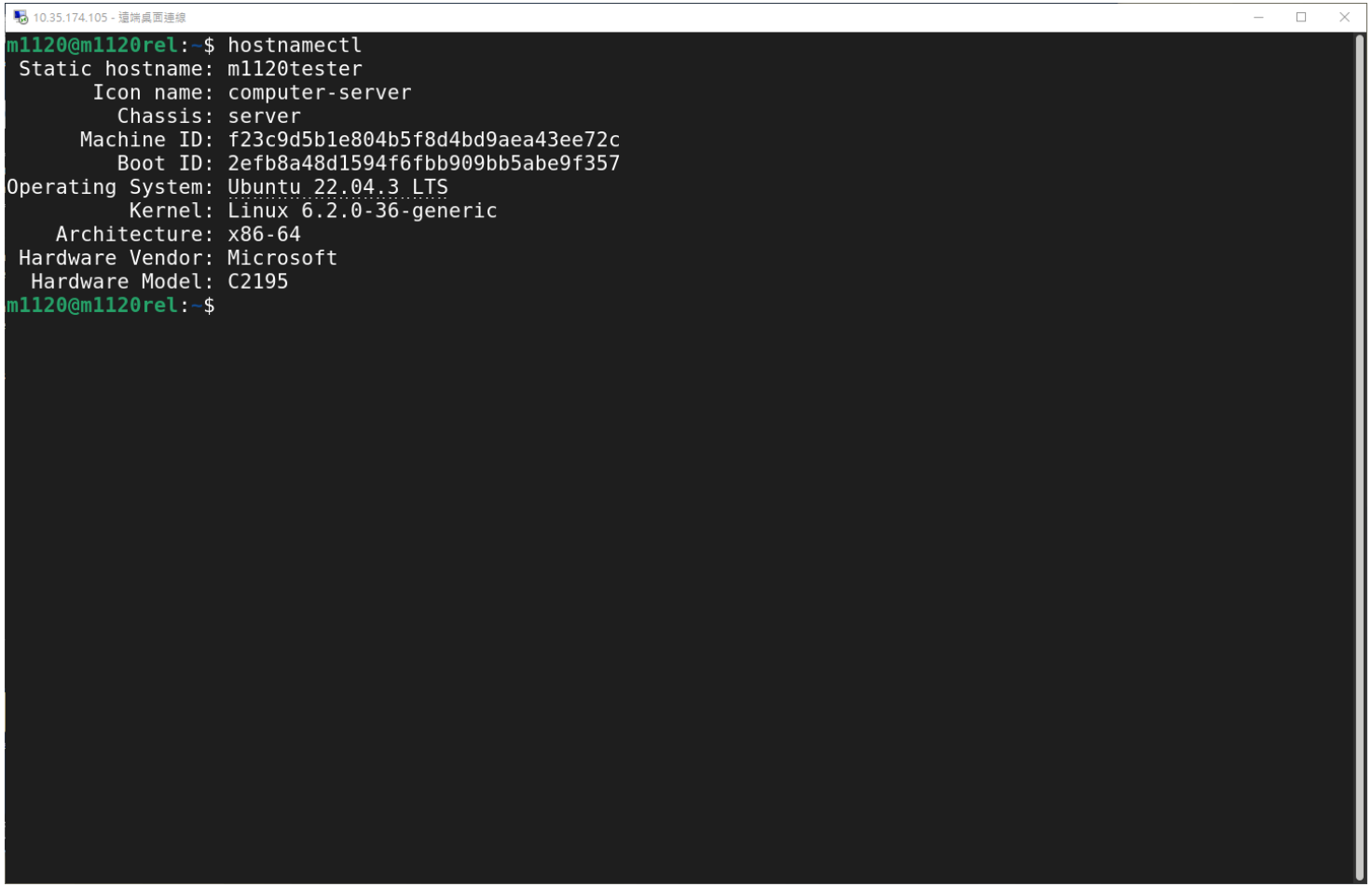
4) Type

sudo hostnamectl set-hostname <desiredname> to change the hostname



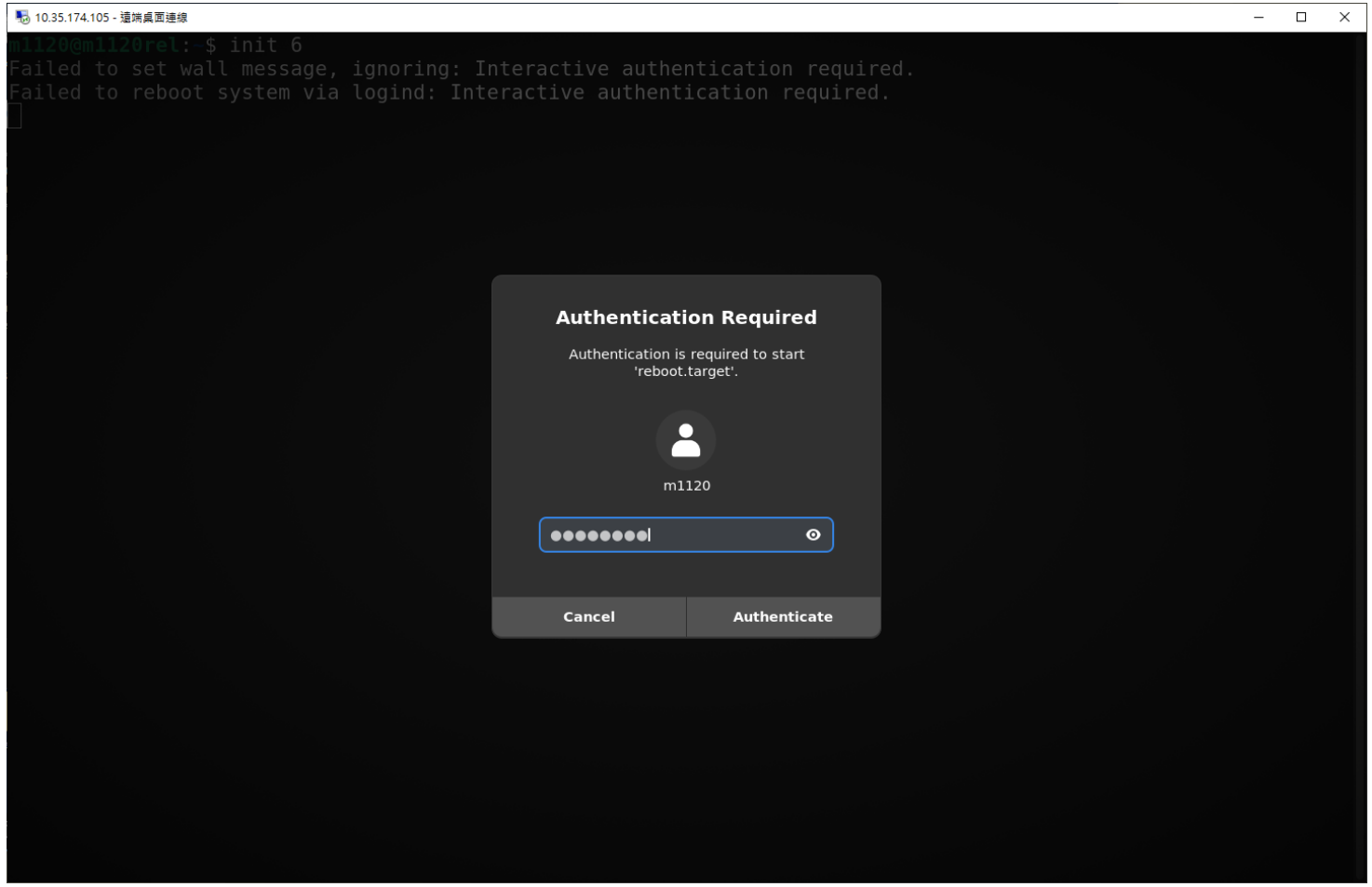
Change Hostname

5) Type **hostnamectl** to check the hostname is changed



Change Hostname

6) Type **init 6** to reboot the system, make sure the new hostname is taking effect



Scripts We Run

Understanding all the scripts

Scripts We Run

In M1120 DV, we use Ubuntu to execute almost all the reliability environmental tests due to the Cerberus stress firmware can only work under Linux.

I created several scripts and already placed in the OS; the scripts will execute by itself if it boot up, but since there were different test condition, we might need to modify the test duration or anything, so we still need to understand it.

All script were placed in /home/m1120/BFT/

And these scripts were controlled by rc-local.service, this is the function like the startup under Windows; and this file was placed in /etc/rc.local.

Scripts We Run

The scripts we used are the followings;

Run_BFT.sh; for collecting firmware and other device info on M1120

Run_FIO.sh; for generating workload for Cerberus (aka Manticore)

Run_SDR.sh; for collecting SDR every min

Run_LionTemp.sh; for collecting Cerberus die temperature

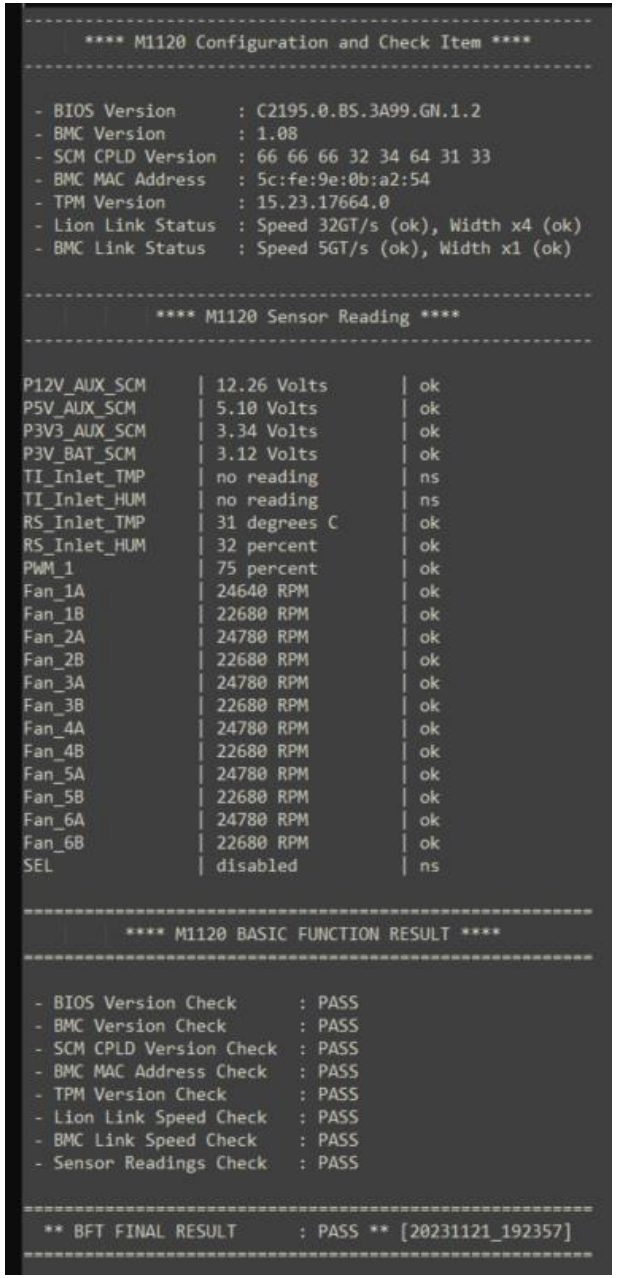
After boot up, the Run_FIO.sh, Run_SDR.sh and Run_LionTemp.sh will be executed by rc-local.service; Run_BFT.sh will be execute every 5min. After target cycles of Run_BFT finished, the rc-local.service will collect all the logs and placed in /home/m1120/BFT/BFT_Logs.

The temporary logs will be in /home/m1120/BFT/BFT_Logs_Temp and /home/m1120/BFT/M1120_DV_BFT_$timestamp.

Scripts We Run

A proper BFT log will look like this →

The Tools I used here -
BIOS : dmidecode
BMC : ipmitool
SCM CPLD: ipmitool
TPM : eltt2
Lion/BMC Link Speed : lspci
Sensor Reading : ipmitool



```
------------------------------------------------------------
     **** M1120 Configuration and Check Item ****
------------------------------------------------------------

 - BIOS Version      : C2195.0.BS.3A99.GN.1.2
 - BMC Version       : 1.08
 - SCM CPLD Version  : 66 66 66 32 34 64 31 33
 - BMC MAC Address   : 5c:fe:9e:0b:a2:54
 - TPM Version       : 15.23.17664.0
 - Lion Link Status  : Speed 32GT/s (ok), Width x4 (ok)
 - BMC Link Status   : Speed 5GT/s (ok), Width x1 (ok)

------------------------------------------------------------
          **** M1120 Sensor Reading ****
------------------------------------------------------------

P12V_AUX_SCM     | 12.26 Volts       | ok
P5V_AUX_SCM      | 5.10 Volts        | ok
P3V3_AUX_SCM     | 3.34 Volts        | ok
P3V_BAT_SCM      | 3.12 Volts        | ok
TI_Inlet_TMP     | no reading        | ns
TI_Inlet_HUM     | no reading        | ns
RS_Inlet_TMP     | 31 degrees C      | ok
RS_Inlet_HUM     | 32 percent        | ok
PWM_1            | 75 percent        | ok
Fan_1A           | 24640 RPM         | ok
Fan_1B           | 22680 RPM         | ok
Fan_2A           | 24780 RPM         | ok
Fan_2B           | 22680 RPM         | ok
Fan_3A           | 24780 RPM         | ok
Fan_3B           | 22680 RPM         | ok
Fan_4A           | 24780 RPM         | ok
Fan_4B           | 22680 RPM         | ok
Fan_5A           | 24780 RPM         | ok
Fan_5B           | 22680 RPM         | ok
Fan_6A           | 24780 RPM         | ok
Fan_6B           | 22680 RPM         | ok
SEL              | disabled          | ns

============================================================
          **** M1120 BASIC FUNCTION RESULT ****
============================================================

 - BIOS Version Check      : PASS
 - BMC Version Check       : PASS
 - SCM CPLD Version Check  : PASS
 - BMC MAC Address Check   : PASS
 - TPM Version Check       : PASS
 - Lion Link Speed Check   : PASS
 - BMC Link Speed Check    : PASS
 - Sensor Readings Check   : PASS

============================================================
  ** BFT FINAL RESULT      : PASS ** [20231121_192357]
============================================================
```

Scripts We Run

An example of FIO log will look like this;

Since this is the ram disk virtualized by Manticore, we don't care about the IOPS, just make sure the test duration is correct and there's no error messages.

```
throughput-test-job: (g=0): rw=read, bs=(R) 128KiB-128KiB, (W) 128KiB-128KiB, (T) 128KiB-128KiB, ioengine=libaio, iodepth=64
...
fio-3.28
Starting 12 processes

throughput-test-job: (groupid=0, jobs=12): err= 0: pid=2760: Wed Nov  8 01:52:56 2023
  read: IOPS=103k, BW=12.6GiB/s (13.5GB/s)(111TiB/9000007msec) Test Duration
    slat (usec): min=5, max=1148.4k, avg=115.52, stdev=2688.69
    clat (usec): min=87, max=4912.6k, avg=7338.79, stdev=63872.62
     lat (usec): min=105, max=5114.8k, avg=7454.39, stdev=64888.13
    clat percentiles (usec):
     |  1.00th=[   1221],  5.00th=[   1827], 10.00th=[   1876],
     | 20.00th=[   2474], 30.00th=[   3097], 40.00th=[   3326],
     | 50.00th=[   3916], 60.00th=[   4490], 70.00th=[   5473],
     | 80.00th=[   6521], 90.00th=[   8160], 95.00th=[   9503],
     | 99.00th=[  13042], 99.50th=[  16581], 99.90th=[1333789],
     | 99.95th=[1753220], 99.99th=[2122318]
   bw (  MiB/s): min=   10, max=57092, per=100.00%, avg=12891.65, stdev=1081.20, samples=215943
   iops        : min=   82, max=456740, avg=103131.16, stdev=8649.53, samples=215943
  lat (usec)   : 100=0.01%, 250=0.01%, 500=0.01%, 750=0.01%, 1000=0.01%
  lat (msec)   : 2=12.51%, 4=37.59%, 10=45.62%, 20=3.94%, 50=0.12%
  lat (msec)   : 100=0.01%, 250=0.01%, 500=0.04%, 750=0.01%, 1000=0.05%
  lat (msec)   : 2000=0.11%, >=2000=0.02%
  cpu          : usr=0.61%, sys=8.15%, ctx=123495463, majf=2, minf=24782
  IO depths    : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%
     submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
     complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.1%, >=64=0.0%
     issued rwts: total=927208912,0,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=64

Run status group 0 (all jobs):
   READ: bw=12.6GiB/s (13.5GB/s), 12.6GiB/s-12.6GiB/s (13.5GB/s-13.5GB/s), io=111TiB (122TB), run=9000007-9000007msec

Disk stats (read/write):
  nvme4n1: ios=927187105/0, merge=0/0, ticks=3266060001/0, in_queue=3266060001, util=100.00%
```

Scripts We Run

A normal pretest SEL Log will look like this -

```
ipmiutil sel version 3.18
-- BMC version 1.05, IPMI version 2.0
SEL Ver 51 Support 02, Size = 4099 records (Used=4, Free=4095)
RecId Date/Time_______ SEV Src_ Evt_Type___ Sens# Evt_detail - Trig [Evt_data]
0001 01/13/70 01:45:25 INF BMC  Management Subsystem Health #ee  Other FW HAL error 6f [a1 00 ff]
0002 01/13/70 01:47:08 INF BMC  OEM(e1) #ef  - 6f [03 00 ff]
0003 01/13/70 01:47:08 INF BMC  OEM(e1) #ef  - 6f [03 10 00]
0004 01/13/70 01:47:08 INF BMC  OEM(e1) #ef  - 6f [03 20 00]
ipmiutil sel, completed successfully
```

0001 : CPLD boot SEL (from CFG0 to CFG1 boot)

0002 to 0004 : System config didn't match with inventory.json

- Above entries are expected logs, can be ignored

```
ipmiutil sel version 3.18
-- BMC version 1.05, IPMI version 2.0
SEL Ver 51 Support 02, Size = 4095 records (Used=0, Free=4095)
RecId Date/Time_______ SEV Src_ Evt_Type___ Sens# Evt_detail - Trig [Evt_data]
Firmware Log (SEL) is empty
ipmiutil sel, completed successfully
```

- Above log means SEL is empty (Known Issue: Bug 1480424)

Scripts We Run

A Lion temperature Log will look like this -

```
 Row 0  0x10 otp idfuse read 0x50a19913 and readback is 0x50a19913
 Row 1 count 0x10 otp idfuse read 0x80ffff09 and readback is 0x80ffff09
Running AES, SHA and UPKA0-15 in continous loop
Channel 1 - PCIe0: TSEN_CH1_DATA = 821 Output Raw = 202 Output Temp = 49
Channel 2 - PCIe1: TSEN_CH2_DATA = 823 Output Raw = 200 Output Temp = 50
Channel 3 - UPKA0 : TSEN_CH3_DATA = 818 Output Raw = 205 Output Temp = 48
Channel 4 - UPKA1 : TSEN_CH4_DATA = 819 Output Raw = 204 Output Temp = 48
Channel 5 - NQM: TSEN_CH5_DATA = 814 Output Raw = 209 Output Temp = 46
Channel 6 - BCP: TSEN_CH6_DATA = 816 Output Raw = 207 Output Temp = 47
Channel 7 - Middle: TSEN_CH7_DATA = 814 Output Raw = 209 Output Temp = 46

 Row 0  0x10 otp idfuse read 0x50a19913 and readback is 0x50a19913
 Row 1 count 0x10 otp idfuse read 0x80ffff09 and readback is 0x80ffff09
Running AES, SHA and UPKA0-15 in continous loop
Channel 1 - PCIe0: TSEN_CH1_DATA = 821 Output Raw = 202 Output Temp = 49
Channel 2 - PCIe1: TSEN_CH2_DATA = 824 Output Raw = 199 Output Temp = 50
Channel 3 - UPKA0 : TSEN_CH3_DATA = 817 Output Raw = 206 Output Temp = 47
Channel 4 - UPKA1 : TSEN_CH4_DATA = 818 Output Raw = 205 Output Temp = 48
Channel 5 - NQM: TSEN_CH5_DATA = 815 Output Raw = 208 Output Temp = 46
Channel 6 - BCP: TSEN_CH6_DATA = 816 Output Raw = 207 Output Temp = 47
Channel 7 - Middle: TSEN_CH7_DATA = 814 Output Raw = 209 Output Temp = 46

 Row 0  0x10 otp idfuse read 0x50a19913 and readback is 0x50a19913
 Row 1 count 0x10 otp idfuse read 0x80ffff09 and readback is 0x80ffff09
Running AES, SHA and UPKA0-15 in continous loop
Channel 1 - PCIe0: TSEN_CH1_DATA = 819 Output Raw = 204 Output Temp = 48
Channel 2 - PCIe1: TSEN_CH2_DATA = 823 Output Raw = 200 Output Temp = 50
Channel 3 - UPKA0 : TSEN_CH3_DATA = 817 Output Raw = 206 Output Temp = 47
Channel 4 - UPKA1 : TSEN_CH4_DATA = 819 Output Raw = 204 Output Temp = 48
Channel 5 - NQM: TSEN_CH5_DATA = 815 Output Raw = 208 Output Temp = 46
Channel 6 - BCP: TSEN_CH6_DATA = 819 Output Raw = 204 Output Temp = 48
Channel 7 - Middle: TSEN_CH7_DATA = 815 Output Raw = 208 Output Temp = 46

 Row 0  0x10 otp idfuse read 0x50a19913 and readback is 0x50a19913
 Row 1 count 0x10 otp idfuse read 0x80ffff09 and readback is 0x80ffff09
Running AES, SHA and UPKA0-15 in continous loop
Channel 1 - PCIe0: TSEN_CH1_DATA = 822 Output Raw = 201 Output Temp = 49
Channel 2 - PCIe1: TSEN_CH2_DATA = 825 Output Raw = 198 Output Temp = 50
Channel 3 - UPKA0 : TSEN_CH3_DATA = 820 Output Raw = 203 Output Temp = 48
Channel 4 - UPKA1 : TSEN_CH4_DATA = 819 Output Raw = 204 Output Temp = 48
Channel 5 - NQM: TSEN_CH5_DATA = 817 Output Raw = 206 Output Temp = 47
Channel 6 - BCP: TSEN_CH6_DATA = 817 Output Raw = 206 Output Temp = 47
Channel 7 - Middle: TSEN_CH7_DATA = 815 Output Raw = 208 Output Temp = 46

 Row 0  0x10 otp idfuse read 0x50a19913 and readback is 0x50a19913
 Row 1 count 0x10 otp idfuse read 0x80ffff09 and readback is 0x80ffff09
Running AES, SHA and UPKA0-15 in continous loop
Channel 1 - PCIe0: TSEN_CH1_DATA = 822 Output Raw = 201 Output Temp = 49
Channel 2 - PCIe1: TSEN_CH2_DATA = 826 Output Raw = 197 Output Temp = 51
Channel 3 - UPKA0 : TSEN_CH3_DATA = 819 Output Raw = 204 Output Temp = 48
Channel 4 - UPKA1 : TSEN_CH4_DATA = 819 Output Raw = 204 Output Temp = 48
Channel 5 - NQM: TSEN_CH5_DATA = 818 Output Raw = 205 Output Temp = 48
Channel 6 - BCP: TSEN_CH6_DATA = 818 Output Raw = 205 Output Temp = 48
Channel 7 - Middle: TSEN_CH7_DATA = 816 Output Raw = 207 Output Temp = 47
```

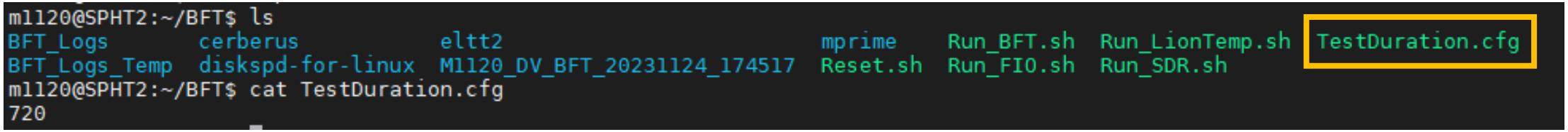
It’s generating automatically by Lion itself, We use header cable/UART to collect the temperature readings from Lion.

The only value we care about is Channel 7; We’ll use the value to check the workload.

Scripts We Run

Long story short, the BFT/SDR/FIO will run for a specific of time/cycle, and LionTemp will generate automatically so we cannot control the time or cycle.

To modify the test duration, all we have to do is to edit the file TestDurarion.cfg under directory /home/m1120/BFT, the unit is minute.



After modified the specific duration, you can use Reset.sh in the same directory to clear all test logs, and then reboot to start the test.

Scripts We Run - Result Collector

After a test is finished, you can use this script to collect the logs from Ubuntu, but make sure your laptop is connected to it though network. **You can modify the Setup area for different number of samples as you desired.**

```
@echo off
SetLocal EnableDelayedExpansion
title=M1120_DV_Log_Collector

:Setup

set /a TotalBlade=2
set SampleName[1]=SPHT1
set SampleName[2]=SPHT2

for /l %%n in (1, 1, %TotalBlade%) do (
    echo [ - Collecting Data from !SampleName[%%n]! ]
    rm -rf .\Test_Logs\!SampleName[%%n]!
    mkdir .\Test_Logs\!SampleName[%%n]!
    pscp.exe -p -r -pw p@ssw0rd m1120@!SampleName[%%n]!:/home/m1120/BFT/BFT_Logs/* .\Linux_Logs\!SampleName[%%n]!\ > nul
    )

pause
```

Download the Scripts : HERE!

Scripts We Run - Result Checker

Following script will help you to check the BFT logs, **it's a simple script for checking all BFT logs you executed and give you an instant result.** And you can modify the setup area for different number of samples too.

```
@echo off
title=M1120_DV_BFT_Log_Check
setlocal EnableDelayedExpansion

set /a TotalBlade=2
set SampleName[1]=SPHT1
set SampleName[2]=SPHT2

echo.
echo ========================================
echo   [Step 1/3] Creating timestamp ...
echo ========================================

:SETUP

::SET_TIME_STAMP_AND_CREATE_NEW_FOLDER
Powershell get-date -format "{yyyyMMdd_HHmmss}" > Timestamp.txt
set /p Timestamp=<.\Timestamp.txt
del Timestamp.txt

timeout /t 2 /nobreak > nul

echo.
echo ========================================
echo   [Step 2/3] Checking test data ...
echo ========================================

:START

echo. >> __M1120_AC_Result_%Timestamp%.log
echo ======================== >> __M1120_AC_Result_%Timestamp%.log
echo   /// TEST RESULTS ///   >> __M1120_AC_Result_%Timestamp%.log
echo ======================== >> __M1120_AC_Result_%Timestamp%.log

for /l %%n in (1, 1, %TotalBlade%) do (

setlocal enabledelayedexpansion

    set A_PASSED=0
    set A_FAILED=0
    set A_CYCLE=0

    for /d %%d in (Test_Logs\!SampleName[%%n]!\M1120_DV_BFT_*) do (

        set /a A_CYCLE+=1

        :: USE_SINGLE_TEST_RESULT_TO_CHECK
        findstr /S /i /c:"** BFT FINAL RESULT      : FAIL **" %%d\* > nul
        if !errorlevel! == 0 set /a A_FAILED+=1
        findstr /S /i /c:"** BFT FINAL RESULT      : FAIL **" %%d\* > nul
        if !errorlevel! == 1 set /a A_PASSED+=1
    )
```

```
    :: TYPE_PASS_AND_FAIL_SUMMARY_OF_ALL_DUTS
    echo  [ !SampleName[%%n]! ] >> __M1120_AC_Result_%Timestamp%.log
    echo  - Cycles:  !A_CYCLE!  >> __M1120_AC_Result_%Timestamp%.log
    echo  - Pass:    !A_PASSED! >> __M1120_AC_Result_%Timestamp%.log
    echo  - Fail:    !A_FAILED! >> __M1120_AC_Result_%Timestamp%.log
    echo ======================== >> __M1120_AC_Result_%Timestamp%.log

    endlocal

)

echo.
echo ========================================
echo   [Step 3/3] Collecing failed tests ...
echo ========================================
echo.

echo. >> __M1120_AC_Result_%Timestamp%.log
echo ======================== >> __M1120_AC_Result_%Timestamp%.log
echo   /// FAILED TESTS ///   >> __M1120_AC_Result_%Timestamp%.log
echo ======================== >> __M1120_AC_Result_%Timestamp%.log

for /l %%n in (1, 1, %TotalBlade%) do (

setlocal enabledelayedexpansion

    echo [ !SampleName[%%n]! ] >> __M1120_AC_Result_%Timestamp%.log

    for /d %%d in (Test_Logs\!SampleName[%%n]!\M1120_DV_BFT_*) do (

        set /a A_CYCLE+=1

        :: USE_SINGLE_TEST_RESULT_TO_CHECK
        findstr /S /i /c:"** BFT FINAL RESULT      : FAIL **" %%d\* > nul
        if !errorlevel! == 0 (
        echo %%~nxd | awk -F_ "{print $4,$5}" >> __M1120_AC_Result_%Timestamp%.log
        )

    )

    echo ======================== >> __M1120_AC_Result_%Timestamp%.log

    endlocal

)

echo ================================
echo   Data collection is completed.
echo ================================
echo.
```

```
========================
  /// TEST RESULTS ///
========================
 [ SPHT1 ]
 - Cycles:  9
 - Pass:    7
 - Fail:    2
========================
 [ SPHT2 ]
 - Cycles:  7
 - Pass:    7
 - Fail:    0
========================


========================
  /// FAILED TESTS ///
========================
[ SPHT1 ]
20231123 111511
20231123 112816
========================
[ SPHT2 ]
========================
```

Download the Scripts : HERE!

Scripts We Run - Known Issue

Since the stress firmware on Manticore is a non-official version, there are some known issues that we confirmed with Microsoft, so it can be ignored.

1) **Ramdisk or Cerberus controller will disappear from time to time**
   a) Use command **$ lsblk | grep 32M** to check the ram disk
   b) Use command **$ lspci -d 1b4b:c003** to check the controller
   c) If above command didn't print anything, it means those item were lost
2) **FIO test will fail from time to time and cannot execute again**
   a) Check the FIO.log under directory /home/m1120/BFT/BFT_Logs after test
   b) Sometimes the ram disk size became 0M from 32M after FIO failed

These 2 issues were not resolved, the only thing we can do is reboot/AC cycle to bring them back; AC power cycle is recommended for higher success rate.

Swap the M1120

how to un/install the test sample

Swap the M1120 - Preparation

Because we do not have enough C2195 (as a fixture), in some circumstances we might have to change the M1120 (as test sample) and continue the validation.

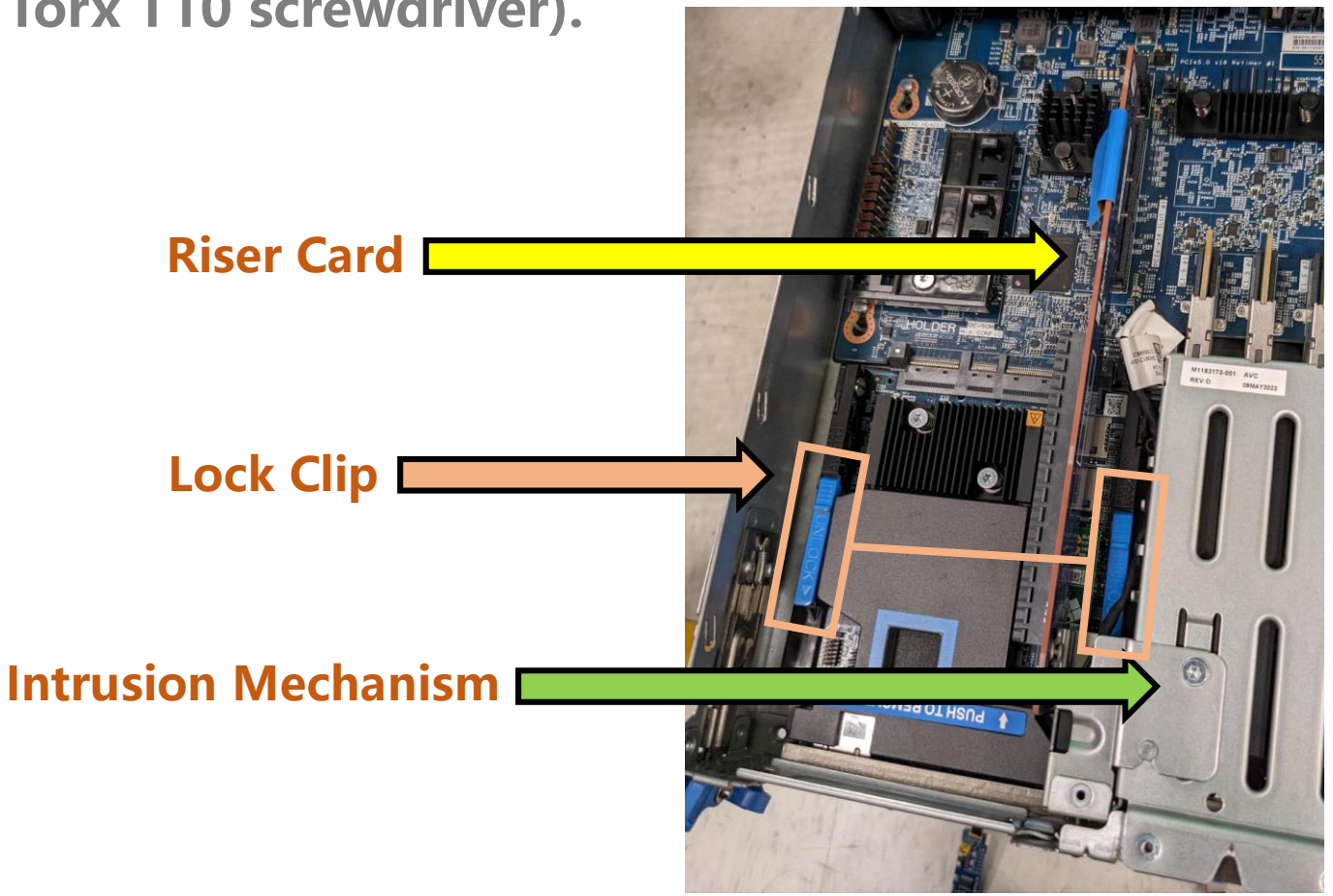
As previous mentioned, the Lion temperature must be recorded by UART cable, therefore, instead of changing the M1120 itself, we also have to connect the UART cable for logging the Lion temperature.

In following pages, I will show you how to install/uninstall the M1120 into the C2195 and connect/disconnect the UART cable when you have to do it.

We will need to use a **Torx T10 screwdriver** to proceed.

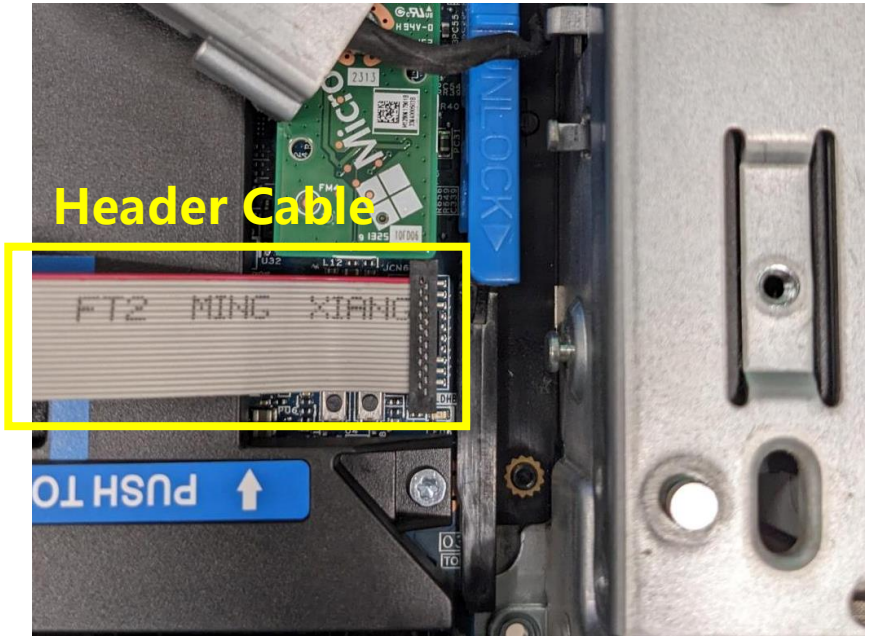
Swap the M1120 - Disassembly

After removing the top cover, you will see the M1120 on the front-left side of C2195 as it showed below, we need to remove the riser card and the intrusion mechanism first (use the Torx T10 screwdriver).



Swap the M1120 - Disassembly

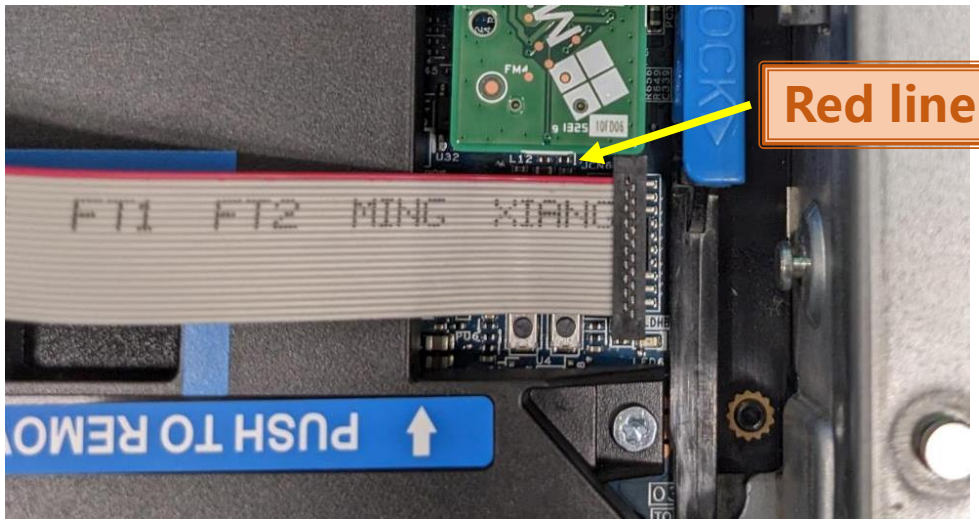
After removing the intrusion mechanism, we need to remove the header cable from JCN6 (on M1120 side). Be careful, the header cable is very fragile, **I would recommend to use a flat head driver to remove it (like I showed below).**



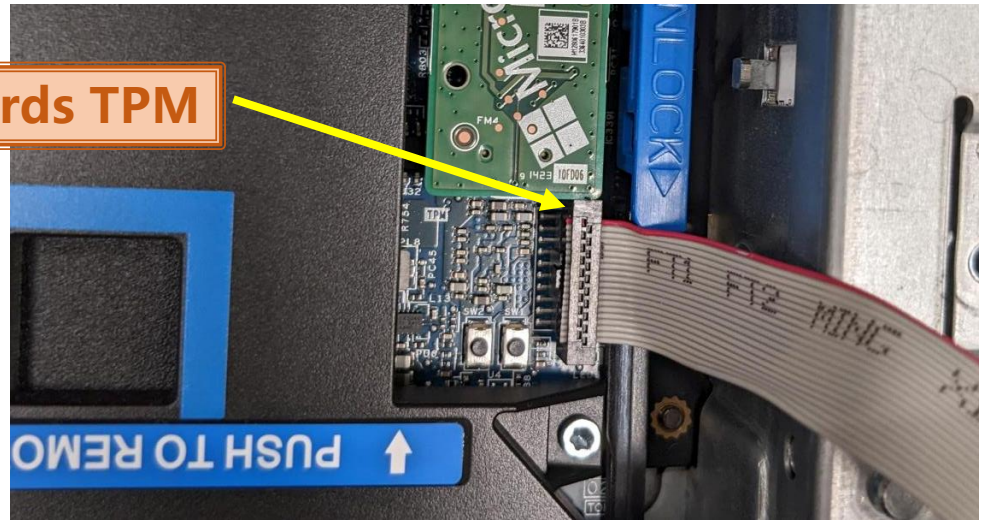
Swap the M1120 - Assembly

When the header cable is removed, you can just unlock the clip and push the M1120 out of sample. For installing the M1120, just reverse these steps.

Be aware, there are 2 types of header cables, no matter which one you get, just keep the red line toward to TPM card and install back in the JCN6.



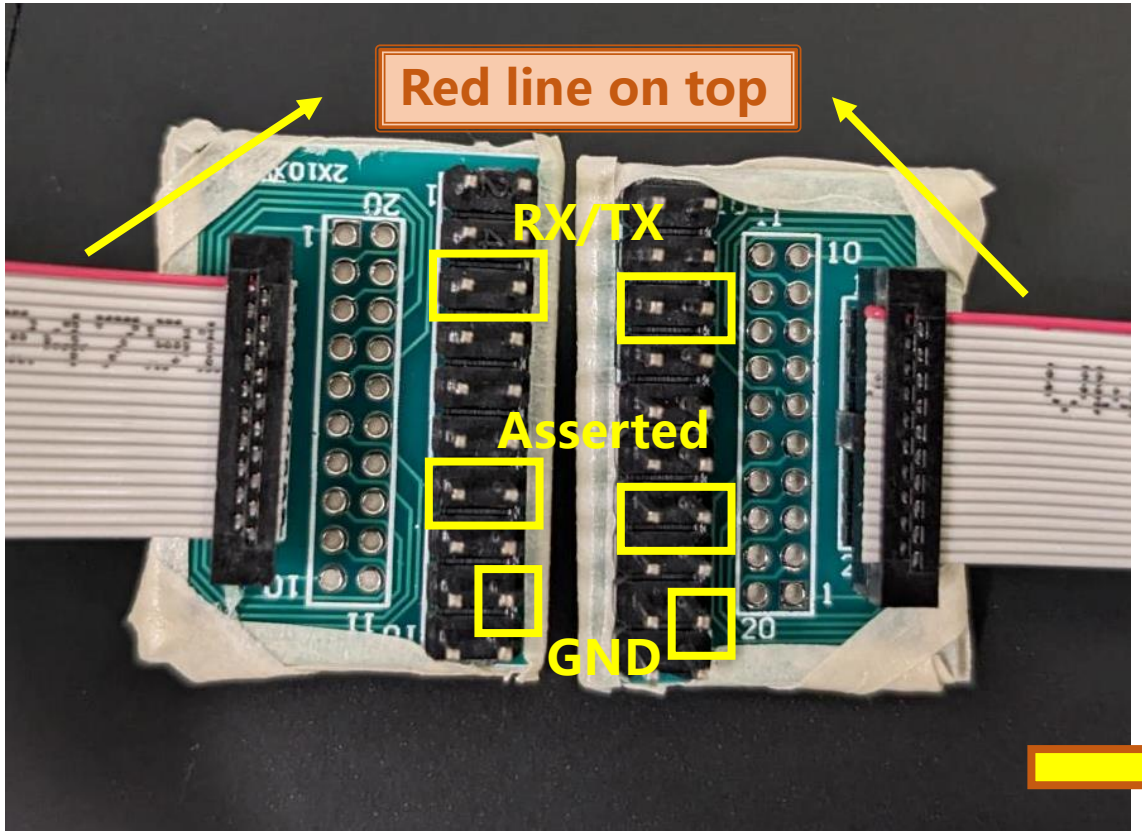
Type A



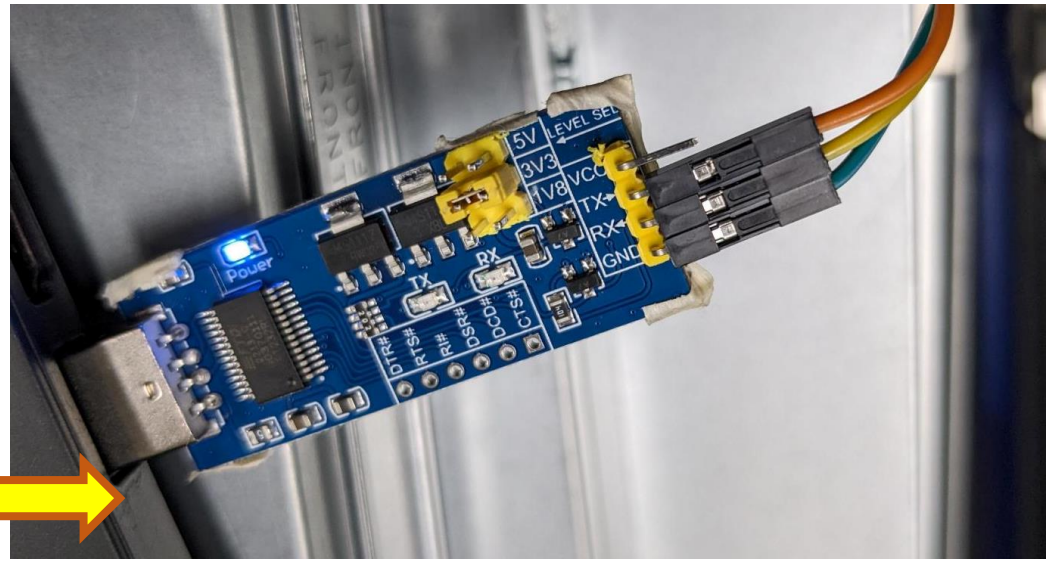
Type B

Swap the M1120 - Assembly

The header cable should be connected to USB UART cable, but sometimes it fall apart; following the below instruction to assemble with female header wires.



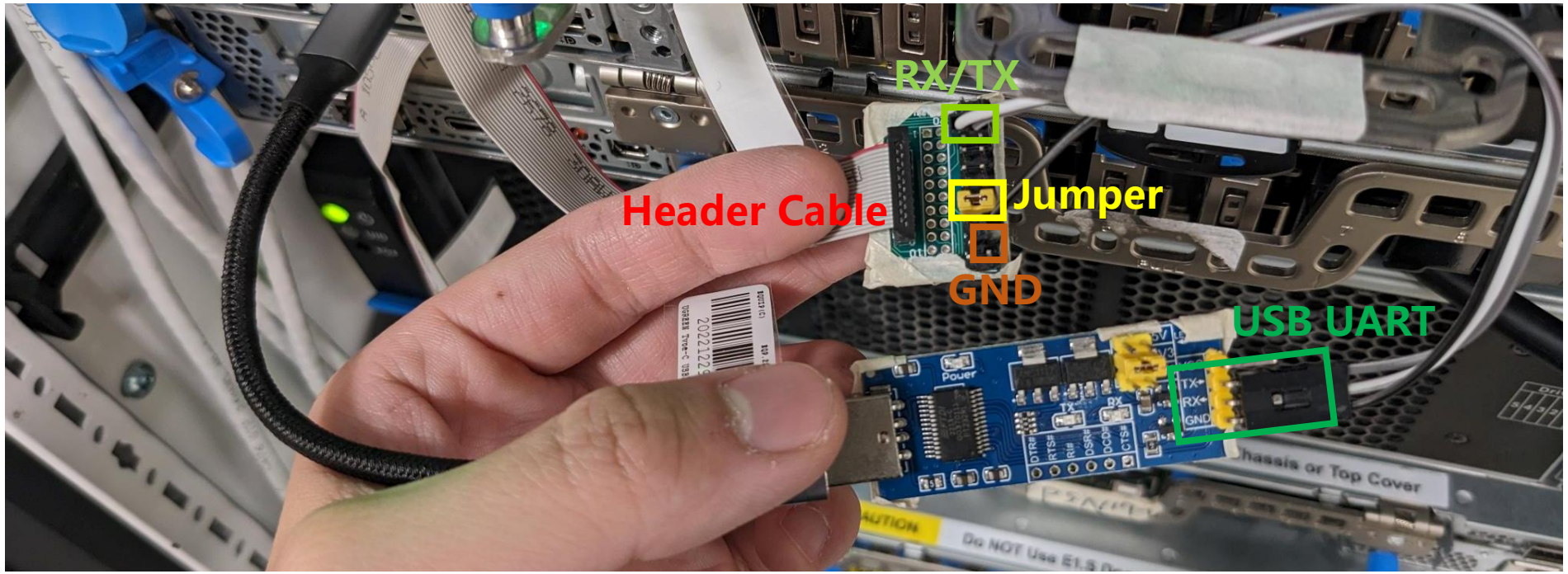
5 - RX
6 - TX
13/14 - Asserted with Jumper
Pin 18 - GND



Connect with USB UART

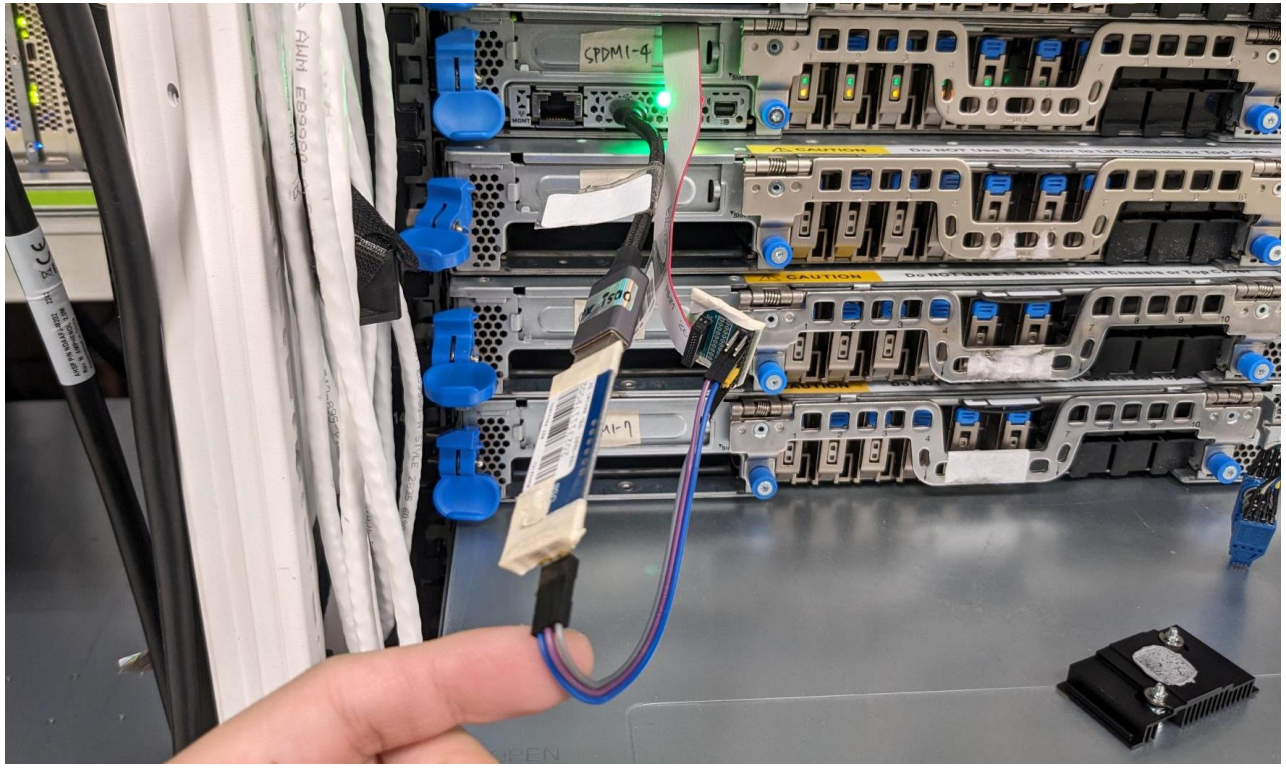
Swap the M1120

The following photo is the overview of header cable and USB UART cable, **pin 13/14 must be asserted by jumper for keeping the stress firmware active.** If you power on the C2195 without pin 13/14 of JCN6 asserted, the stress firmware will not retain and will need to flash again with Dediprog.



Swap the M1120

Remember to connect the USB UART to a Type C-to-A converter cable and connect in back to sample itself, so that the C2195 can capture the Lion temperature logs though UART cable.



thanks mate