UNISOC

Introduction to Sysdump

WWW.UNISOC.COM

UNISOC (SHANGHAI) TECHNOLOGIES CO., LTD.

BACKBONE OF THE DIGITAL WORLD

Revision History

Version	Date	Notes
V1.0	2020/6/1	Initial release.
V1.1	2020/9/6	1. Adjust the document structure. 2. Delete redundant information and optimize the content. 3. Step-by-step description. 4. Merge similar content and correct errors. 5. Change the document name *UNISOC_sysdump_Brief_Introduction* to *Introduction to Sysdump.*
V1.2	2020/10/21	P18 adds the description of new function on logel_r9.20.1401 _P1 and above version.
V1.3	2020/12/29	1. P33 updates the sample figure of bt command. 2. P13 updates the access path of crash tool.

Keyword

Sysdump, Dump2PC, Fulldump, Minidump.

Contents

01 Sysdump

02 Sysdump Configuration

03 Fulldump

04 Dump2PC

05 Minidump

06 Commands of Crash Tool

07 Exception Handling

01

Sysdump

Introduction

Sysdump, or Dump System Memory, is a dump mechanism that converts valid information such as memory information and register information in case of an exception to files. Sysdump can analyze the problem with the help of analysis tools.

When exceptions such as Kernel crash occurs, after flushing cache and other procedures in Kernel, restart and enter Uboot to save all data. During the saving process, there will be corresponding screen prompts. After completion, restart the phone according to the screen prompts, and export the abnormal data file for analysis.

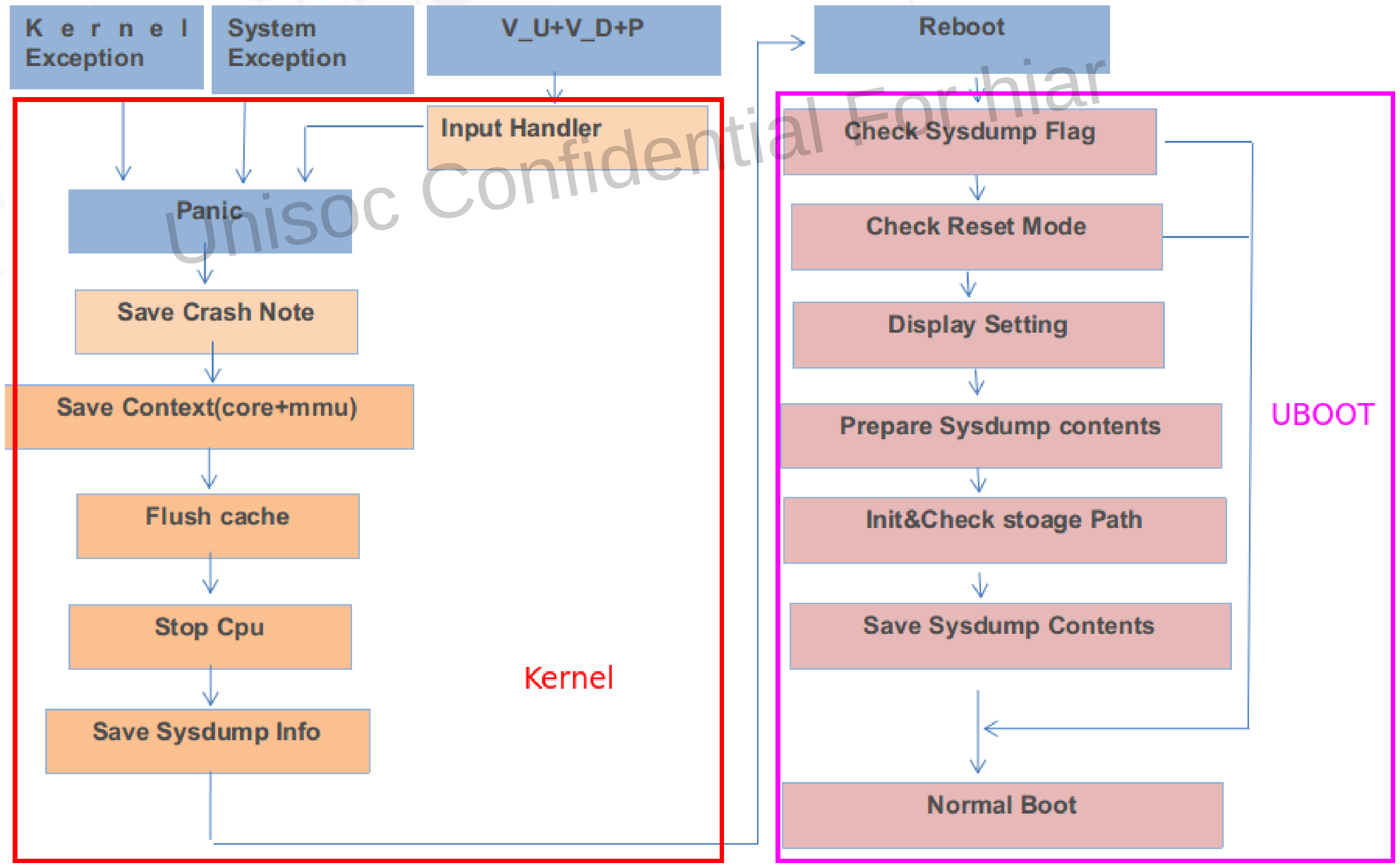
Sysdump has two sub-functions: Fulldump and Minidump, and the two sub-functions are independent of each other.

- Fulldump saves complete DDR information and can support two storage paths, Dump2SD and Dump2PC.
- Minidump saves a small amount of information to a separate SysDumpdb partition. And then the native service program sorts and parses the raw data from the partition and places to the path /SDcard /Minidump.

If the condition allows, Fulldump log is preferred because it can keep a complete snapshot of DRR information, which is more conducive to the in-depth analysis of the problem.

Note: the condition means the device supports Fulldump and has an SD card with more than 4G free space, or it supports Dump2PC.

Design Flow



- Kernel exception, system exception and the system exception triggered by the key combination will all go to the Kernel processing flow, but the operation of 7-second long press will not enter this flow.
- In Kernel stage, Minidump completes the initialization and saves the data of exception handling.
- In UBOOT stage, both Fulldump and Minidump completes the data storage operation.

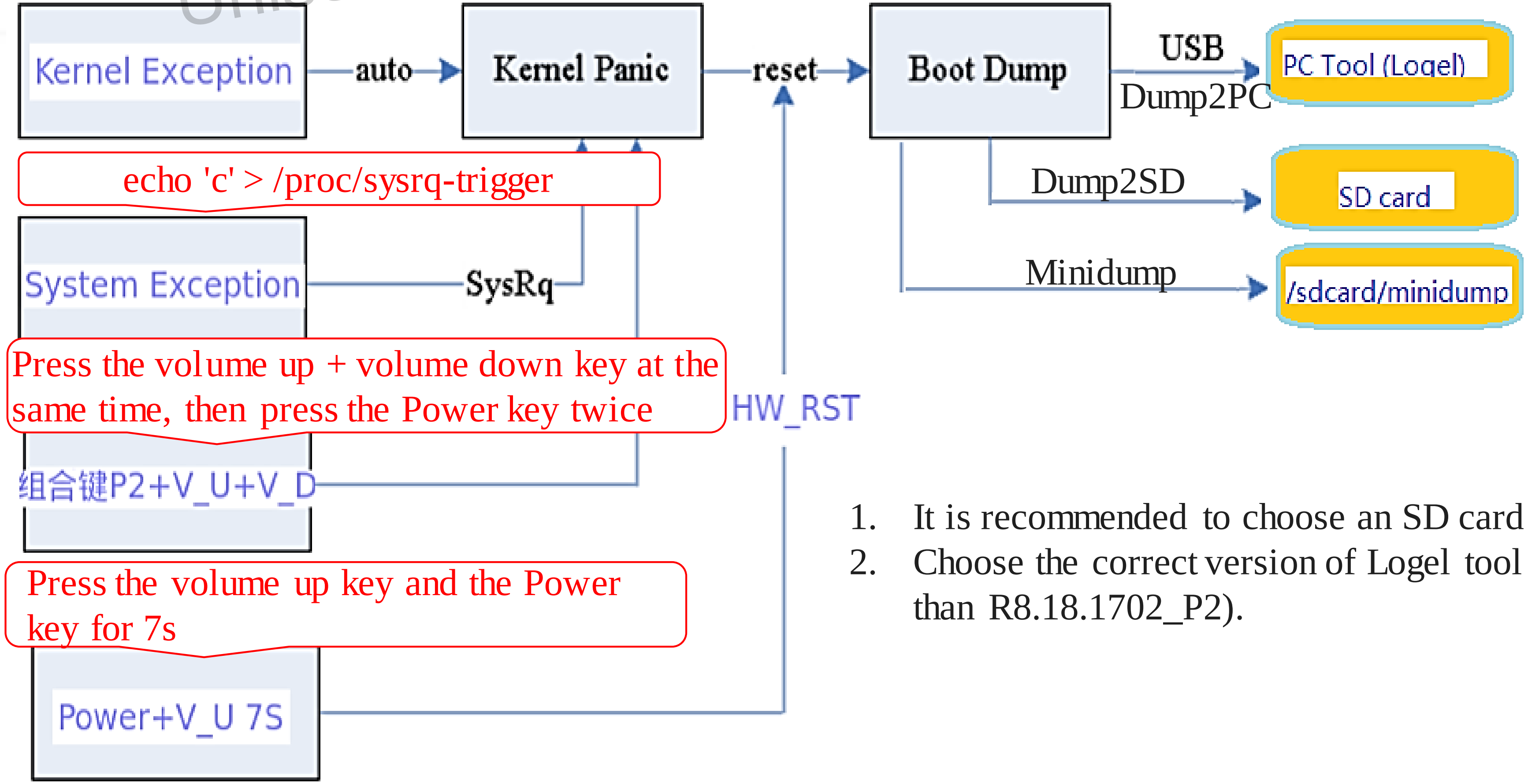
02

Sysdump Configuration

Trigger Modes

Depending on the hardware configuration, Sysdump trigger method is also different, as follows.

To complete the subsequent Sysdump process, the Sysdump interface is embedded in the Panic function.



echo 'c' > /proc/sysrq-trigger

Press the volume up + volume down key at the same time, then press the Power key twice

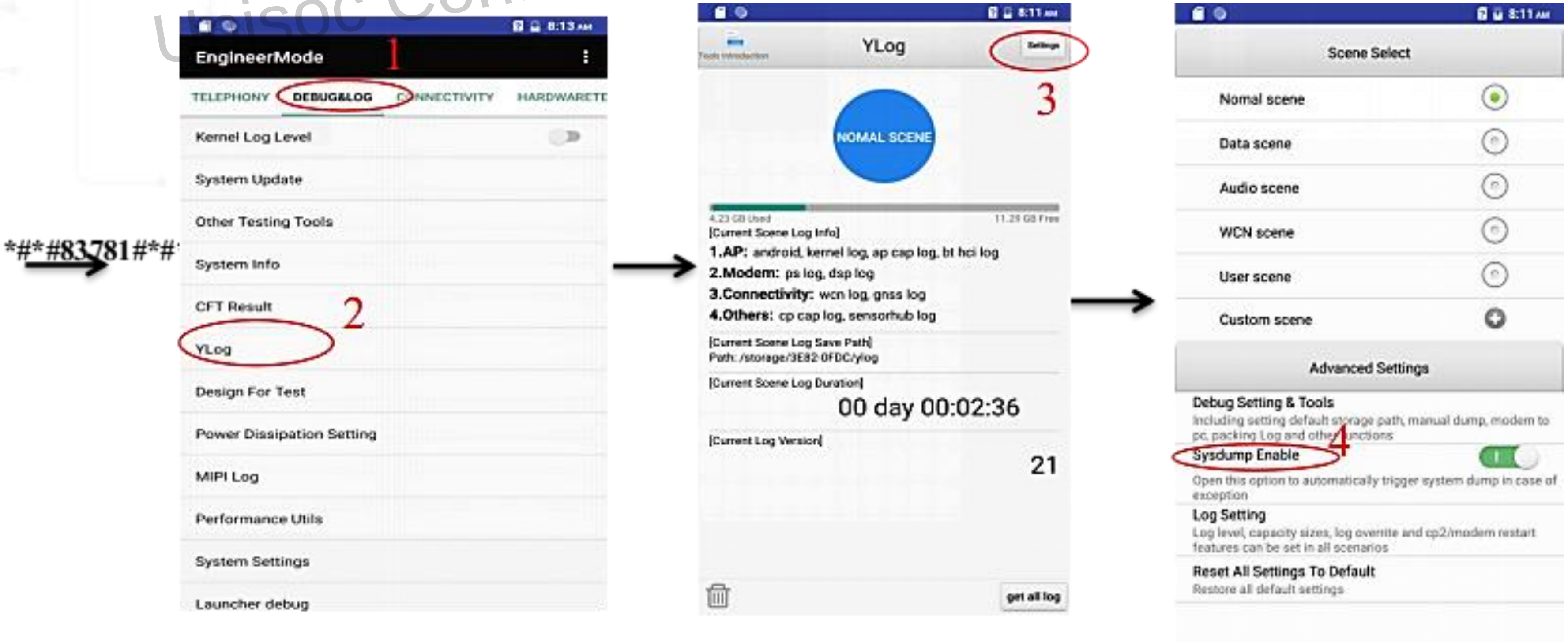
Press the volume up key and the Power key for 7s

1. It is recommended to choose an SD card with a capacity of more than 16G.
2. Choose the correct version of Logel tool (the Logel version must be higher than R8.18.1702_P2).

Functions

- In UserDebug version, Fulldump is enabled by default. In User version, Fulldump is disabled by default.
- FullDump can be enabled or disabled in EngineerMode, and the operation procedure is as follows:

(*#*#83781#*#*) -->DEBUG&LOG-->YLog-->Settings-->Sysdump Enable



EngineerMode

TELEPHONY | DEBUG&LOG | CONNECTIVITY | HARDWARETE

- Kernel Log Level
- System Update
- Other Testing Tools
- System Info
- CFT Result
- YLog
- Design For Test
- Power Dissipation Setting
- MIPI Log
- Performance Utils
- System Settings
- Launcher debug

YLog — Tools Introduction — Settings

NOMAL SCENE

4.23 GB Used — 11.29 GB Free

[Current Scene Log Info]
1.AP: android, kernel log, ap cap log, bt hci log
2.Modem: ps log, dsp log
3.Connectivity: wcn log, gnss log
4.Others: cp cap log, sensorhub log

[Current Scene Log Save Path]
Path: /storage/3E82-0FDC/ylog

[Current Scene Log Duration]
00 day 00:02:36

[Current Log Version]
21

get all log

Scene Select

- Nomal scene ●
- Data scene ○
- Audio scene ○
- WCN scene ○
- User scene ○
- Custom scene

Advanced Settings

Debug Setting & Tools
Including setting default storage path, manual dump, modem to pc, packing Log and other functions

Sysdump Enable ●
Open this option to automatically trigger system dump in case of exception

Log Setting
Log level, capacity sizes, log overrite and cp2/modem restart features can be set in all scenarios

Reset All Settings To Default
Restore all default settings

- ✓ The operation of enabling or disabling Fulldump takes effect immediately, and no need to restart the device.
- ✓ Fulldump will keep the last enabled/disabled state after restart.

- Minidump is enabled by default in both User and UserDebug version.

03

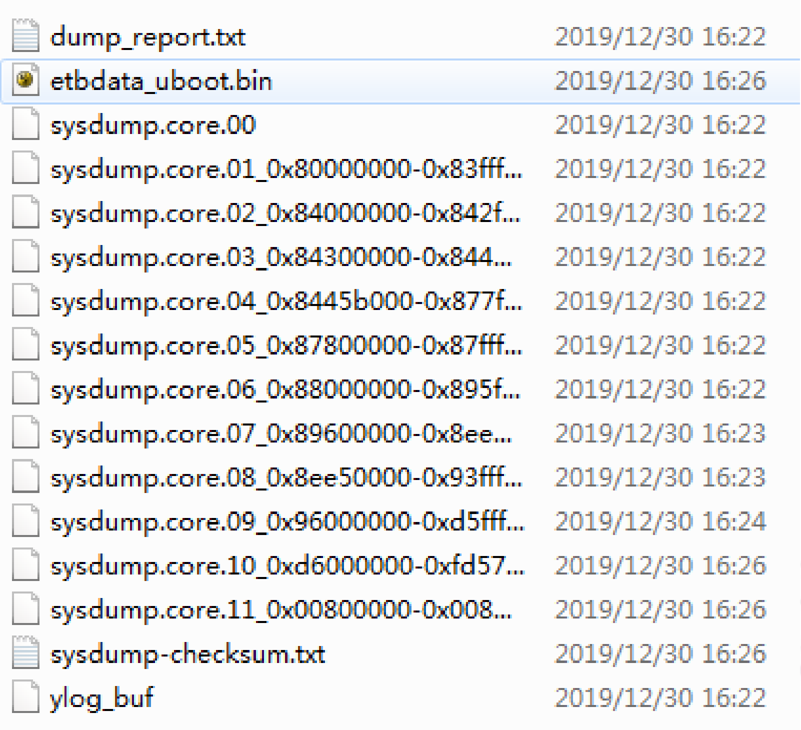
Fulldump

Fulldump (1/2)

- Fulldump files stored in Dump2SD way is generally stored in the SD card, including Sysdump folders. Three historical logs are saved, among which folder 1 is the latest one.
- Check Dump2DC files through Logel tool: File->Open Log Location.

Sysdump files:

- dump_report.txt: Sysdump information files, recording the number of Sysdump files, the reason for the restart, and so on.
- ylog_buf file: record ylog buffer information.
- Sysdump.core*: log files.



dump_report.txt	2019/12/30 16:22
etbdata_uboot.bin	2019/12/30 16:26
sysdump.core.00	2019/12/30 16:22
sysdump.core.01_0x80000000-0x83fff...	2019/12/30 16:22
sysdump.core.02_0x84000000-0x842f...	2019/12/30 16:22
sysdump.core.03_0x84300000-0x844...	2019/12/30 16:22
sysdump.core.04_0x8445b000-0x877f...	2019/12/30 16:22
sysdump.core.05_0x87800000-0x87fff...	2019/12/30 16:22
sysdump.core.06_0x88000000-0x895f...	2019/12/30 16:22
sysdump.core.07_0x89600000-0x8ee...	2019/12/30 16:23
sysdump.core.08_0x8ee50000-0x93fff...	2019/12/30 16:23
sysdump.core.09_0x96000000-0xd5fff...	2019/12/30 16:24
sysdump.core.10_0xd6000000-0xfd57...	2019/12/30 16:26
sysdump.core.11_0x00800000-0x008...	2019/12/30 16:26
sysdump-checksum.txt	2019/12/30 16:26
ylog_buf	2019/12/30 16:22

Without specification, Sysdump refers to Fulldump by default.

Fulldump (2/2)

UNISOC

Fulldump file parsing commands are as follows:

- crash_arm -m phys_base=0x80000000 vmlinux vmcore
- crash_arm64 -m phys_offset=0x80000000 vmlinux vmcore

- Crash tool is obtained in following path:

  vendor/sprd/tools/crash/pycrash/bin/

  vendor/sprd/tools/crash/Usage

  ✓ crash_arm: for 32bit ARM

  ✓ crash_arm64: for 64bit ARM

- vmlinux is the original core file generated when compiling. It is used for kernel debug, the path is out/target/product/xxxx/obj/Kernel/vmlinux
- vmcore is system core dump information collected by Sysdump. It is composed by Sysdump.core*, and the command is as follows.

  ```
  cat Sysdump.core.* > vmcore
  ```

04

Dump2PC

Dump2PC

Dump2PC is used to export the log generated by Sysdump to PC with the help of Logel tool in a no-SD-card condition.

By default:

- If there is an SD card, store log to SD card.
- If there is no SD card or Dump2SD fails, switch to Dump2PC Function automatically, and prompt to connect PC for Dump2PC operation.

It is recommended to use the latest version of Dump2PC tool and SPRD U2S Diag port drivers.

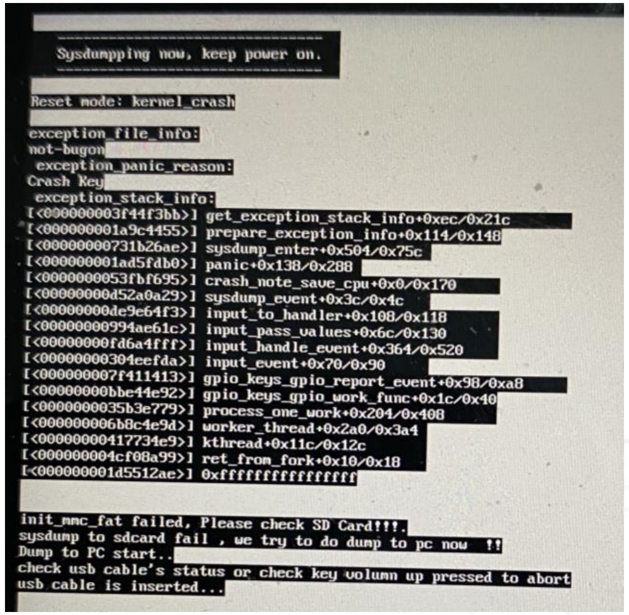
Dump2PC Instructions (1/6)

1. When Sysdump occurs on the device that supports Dump2PC function, it will try Dump2PC automatically after the failure of Dump2SD, and the screen will give the printing information "**check usb cable's status or check key volume up pressed to abort**". The dump has not started at that time, and it will shake hands with PC only after the successful connection with PC.

   Use a USB cable to connect the device to PC. When the device detects a USB cable inserted, the screen will display the printing information "**usb cable is inserted**…".

   It will wait until the battery runs out. If it is connected to a charger before the test, it can be charged in this status to ensure the power will not be exhausted.

   Note: To be detected, there must be a USB insert action.



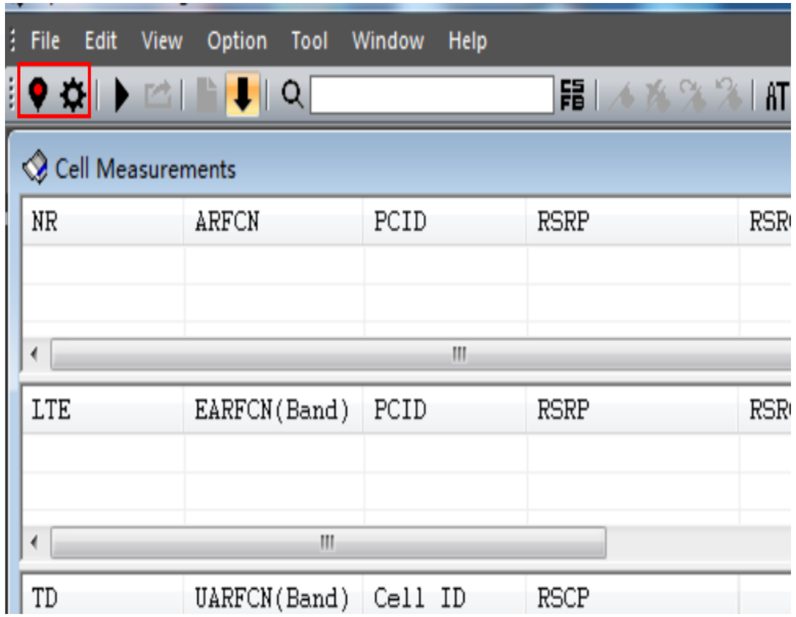
```
 ----------------------------------
  Sysdumpping now, keep power on.
 ----------------------------------

Reset mode: kernel_crash

exception_file_info:
not-bugon
 exception_panic_reason:
Crash Key
 exception_stack_info:
[<000000003f44f3bb>] get_exception_stack_info+0xec/0x21c
[<000000001a9c4455>] prepare_exception_info+0x114/0x148
[<00000000731b26ae>] sysdump_enter+0x504/0x75c
[<000000001ad5fdb0>] panic+0x138/0x288
[<0000000053fbf695>] crash_note_save_cpu+0x0/0x170
[<00000000d52a0a29>] sysdump_event+0x3c/0x4c
[<00000000de9e64f3>] input_to_handler+0x108/0x118
[<00000000994ae61c>] input_pass_values+0x6c/0x130
[<00000000fd6a4fff>] input_handle_event+0x364/0x520
[<00000000304eefda>] input_event+0x70/0x90
[<000000007f411413>] gpio_keys_gpio_report_event+0x98/0xa8
[<00000000bbe44e92>] gpio_keys_gpio_work_func+0x1c/0x40
[<0000000035b3e779>] process_one_work+0x204/0x408
[<000000006b8c4e9d>] worker_thread+0x2a0/0x3a4
[<00000000417734e9>] kthread+0x11c/0x12c
[<000000004cf08a99>] ret_from_fork+0x10/0x18
[<000000001d5512ae>] 0xffffffffffffffff


init_mmc_fat failed, Please check SD Card!!!.
sysdump to sdcard fail , we try to do dump to pc now  !!
Dump to PC start..
check usb cable's status or check key volumn up pressed to abort
usb cable is inserted...
```

Dump2PC Instructions (2/6)

2. Open Logel on PC (the Logel version must be higher than R8.18.1702_P2), interface is shown as Figure 1.

📍 “capture” button, the initial state is red, and it will turn to green when capturing log.

⚙ “capture setting” button, use the default setting, interface is shown as Figure 2.

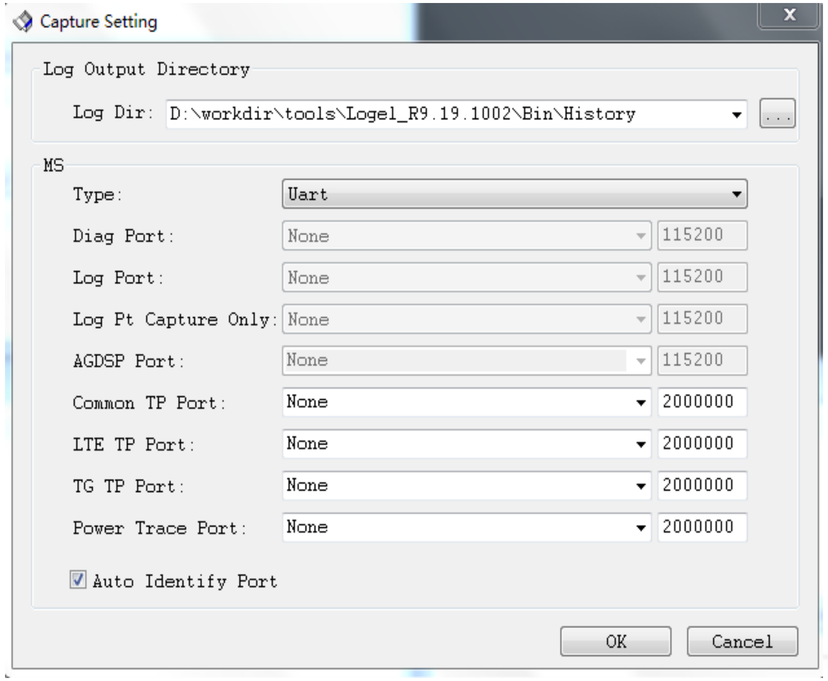


NR	ARFCN	PCID	RSRP	RSR

LTE	EARFCN(Band)	PCID	RSRP	RSR

TD	UARFCN(Band)	Cell ID	RSCP

Figure 1



Log Output Directory

Log Dir: D:\workdir\tools\Logel_R9.19.1002\Bin\History

MS

Type:	Uart	
Diag Port:	None	115200
Log Port:	None	115200
Log Pt Capture Only:	None	115200
AGDSP Port:	None	115200
Common TP Port:	None	2000000
LTE TP Port:	None	2000000
TG TP Port:	None	2000000
Power Trace Port:	None	2000000

☑ Auto Identify Port

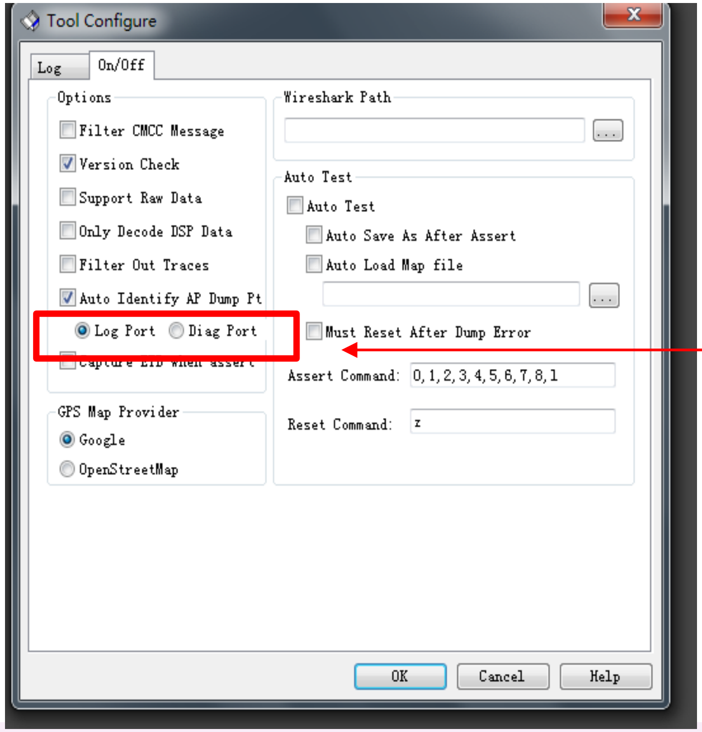
Figure 2

Dump2PC Instructions (3/6)

3. Settings of AP Dump port auto identity

Logel (Logel_R9.19.1002_P1 and above version) disables the automatic recognition function of AP Dump by default. As the port name is same as the download/calibration port name, when a PC uses Pandora/Simba/ResearchDownload at the same time, they will take this port to each other. Therefore, Logel disables this function by default.

You can click on Logel's menu **Option->ToolConfigure**, and check **Auto Identity AP Dump Pt**, as shown in following figure.



Options
- □ Filter CMCC Message
- ☑ Version Check
- □ Support Raw Data
- □ Only Decode DSP Data
- □ Filter Out Traces
- ☑ Auto Identify AP Dump Pt
  - ● Log Port ○ Diag Port
- □ Capture LTD when assert

GPS Map Provider
- ● Google
- ○ OpenStreetMap

Wireshark Path

Auto Test
- □ Auto Test
  - □ Auto Save As After Assert
  - □ Auto Load Map file
  - □ Must Reset After Dump Error

Assert Command: 0,1,2,3,4,5,6,7,8,1

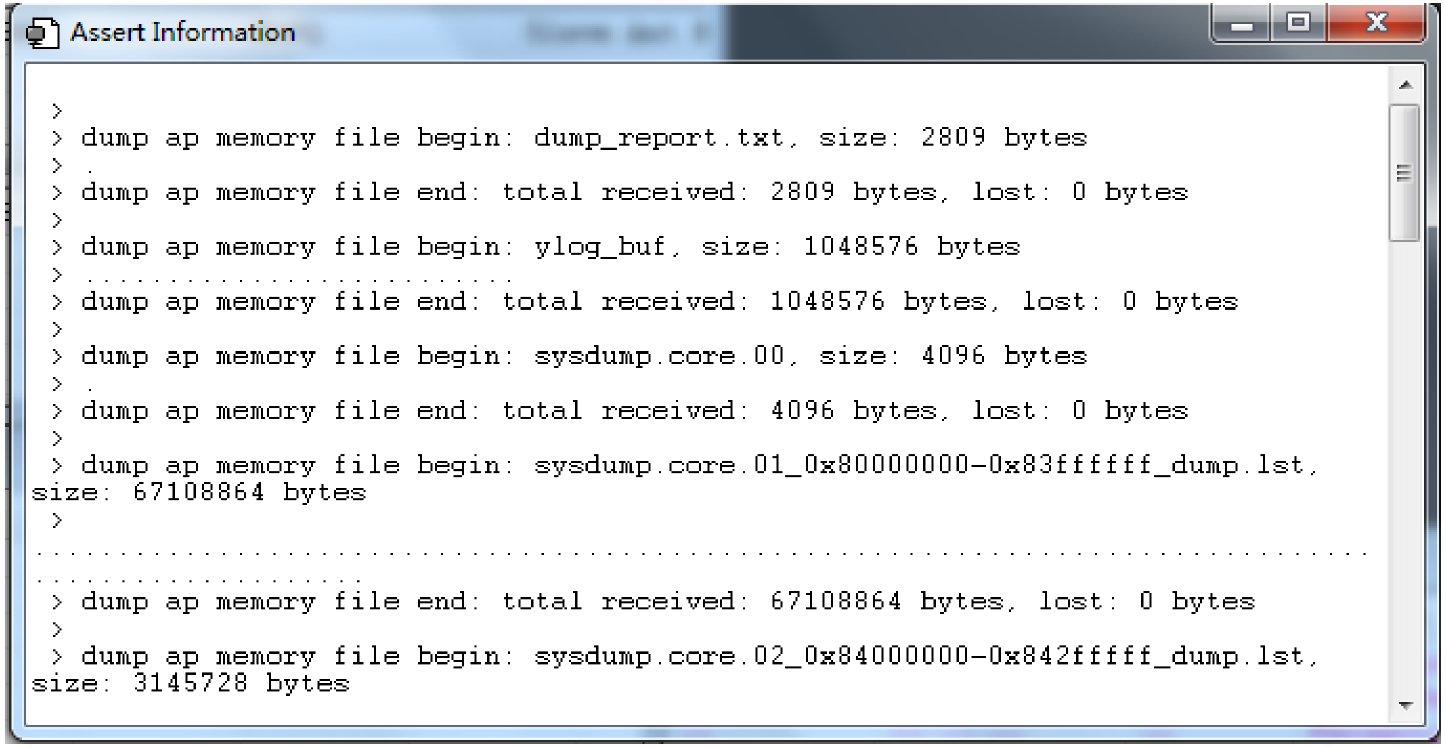
Reset Command: z

- This option is only available in Logel_R9.20.1401_P1 and above version.
- Check **Log Port** when Dump2PC.

Dump2PC Instructions (4/6)

4. When Logel shake hands with phone successfully, the PC tool will automatically pop up the data box and start exporting log, as shown in the following figure.

   Note: It pops up automatically because you have used the same port before. If you have not used the port before, click **capture** button on the PC.

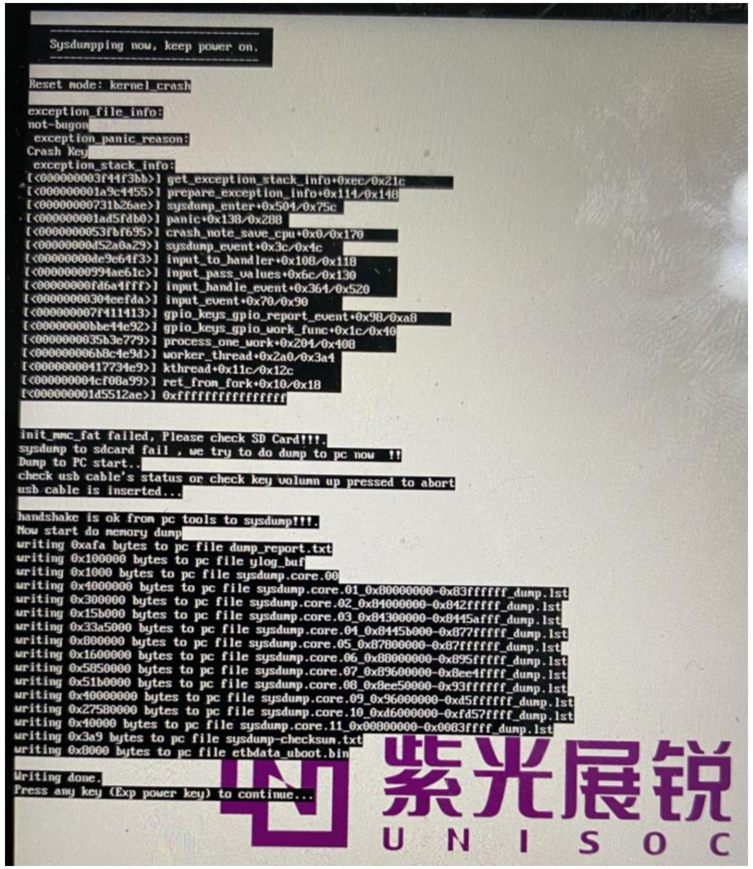


```
 >
 > dump ap memory file begin: dump_report.txt, size: 2809 bytes
 > .
 > dump ap memory file end: total received: 2809 bytes, lost: 0 bytes
 >
 > dump ap memory file begin: ylog_buf, size: 1048576 bytes
 > ..........................
 > dump ap memory file end: total received: 1048576 bytes, lost: 0 bytes
 >
 > dump ap memory file begin: sysdump.core.00, size: 4096 bytes
 > .
 > dump ap memory file end: total received: 4096 bytes, lost: 0 bytes
 >
 > dump ap memory file begin: sysdump.core.01_0x80000000-0x83ffffff_dump.lst,
size: 67108864 bytes
 >
..................................................................................
....................
 > dump ap memory file end: total received: 67108864 bytes, lost: 0 bytes
 >
 > dump ap memory file begin: sysdump.core.02_0x84000000-0x842fffff_dump.lst,
size: 3145728 bytes
```

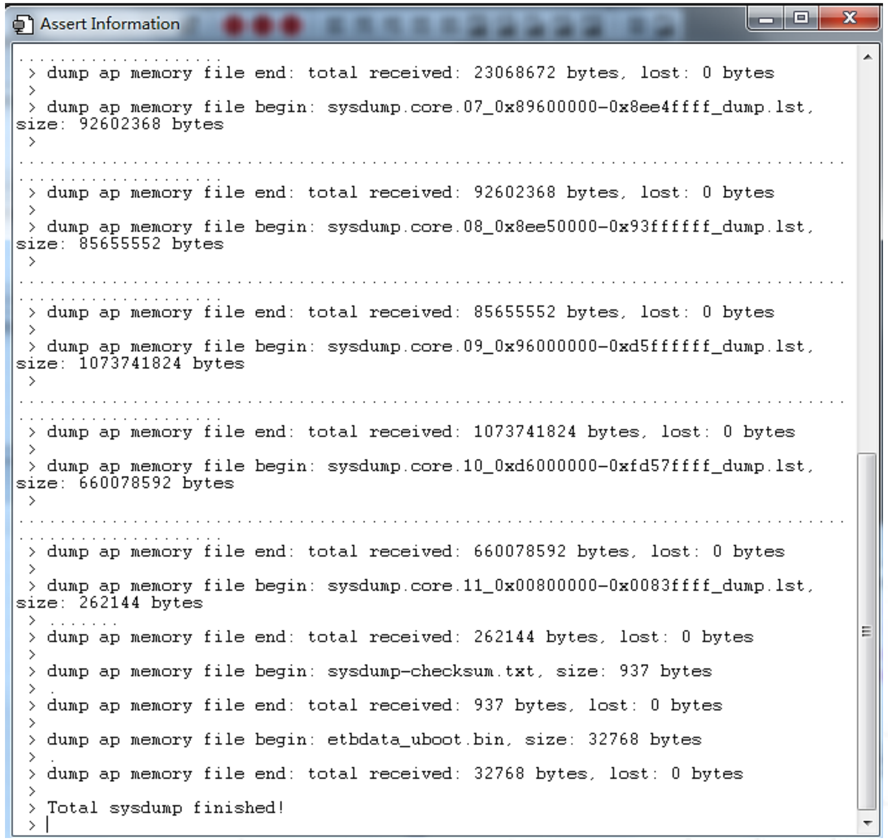
Dump2PC Instructions (5/6)

5. Dump2PC exports log and completes confirmation.

When the dump is finished, the phone will display “**Press any key(Exp power key) to continue…**”, and PC will display “**Total sysdump finished!**”. Then press the volume up key as the device screen prompts to restart the phone, and the whole dump process is finished.



```
Sysdumpping now, keep power on.

Reset mode: kernel_crash

exception_file_info:
not-bugon
 exception_panic_reason:
Crash Key
 exception_stack_info:
[<000000003f44f3bb>] get_exception_stack_info+0xec/0x21c
[<000000001a9c4455>] prepare_exception_info+0x114/0x148
[<00000000731b26ae>] sysdump_enter+0x504/0x75c
[<000000001ad5fdb0>] panic+0x138/0x288
[<0000000053fbf695>] crash_note_save_cpu+0x0/0x170
[<00000000d52a0a29>] sysdump_event+0x3c/0x4c
[<00000000de9e64f3>] input_to_handler+0x108/0x118
[<00000000994ae61c>] input_pass_values+0x6c/0x130
[<00000000fd6a4fff>] input_handle_event+0x364/0x520
[<00000000304ecfda>] input_event+0x70/0x90
[<000000007f411413>] gpio_keys_gpio_report_event+0x98/0xa8
[<00000000bbe44e92>] gpio_keys_gpio_work_func+0x1c/0x40
[<0000000035b3e779>] process_one_work+0x204/0x408
[<000000006b8c4e9d>] worker_thread+0x2a0/0x3a4
[<00000000417734e9>] kthread+0x11c/0x12c
[<000000004cf08a99>] ret_from_fork+0x10/0x18
[<000000001d5512ae>] 0xffffffffffffffff


init_mmc_fat failed, Please check SD Card!!!.
sysdump to sdcard fail , we try to do dump to pc now  !!
Dump to PC start..
check usb cable's status or check key volumn up pressed to abort
usb cable is inserted...

handshake is ok from pc tools to sysdump!!!.
Now start do memory dump
writing 0xafa bytes to pc file dump_report.txt
writing 0x100000 bytes to pc file ylog_buf
writing 0x1000 bytes to pc file sysdump.core.00
writing 0x4000000 bytes to pc file sysdump.core.01_0x80000000-0x83ffffff_dump.lst
writing 0x300000 bytes to pc file sysdump.core.02_0x84000000-0x842fffff_dump.lst
writing 0x15b000 bytes to pc file sysdump.core.03_0x84300000-0x8445afff_dump.lst
writing 0x33a5000 bytes to pc file sysdump.core.04_0x8445b000-0x877fffff_dump.lst
writing 0x800000 bytes to pc file sysdump.core.05_0x87800000-0x87ffffff_dump.lst
writing 0x1600000 bytes to pc file sysdump.core.06_0x88000000-0x895fffff_dump.lst
writing 0x5850000 bytes to pc file sysdump.core.07_0x89600000-0x8ee4ffff_dump.lst
writing 0x51b0000 bytes to pc file sysdump.core.08_0x8ee50000-0x93ffffff_dump.lst
writing 0x40000000 bytes to pc file sysdump.core.09_0x96000000-0xd5ffffff_dump.lst
writing 0x27580000 bytes to pc file sysdump.core.10_0xd6000000-0xfd57ffff_dump.lst
writing 0x40000 bytes to pc file sysdump.core.11_0x00800000-0x0083ffff_dump.lst
writing 0x3a9 bytes to pc file sysdump-checksum.txt
writing 0x8000 bytes to pc file etbdata_uboot.bin

Writing done.
Press any key (Exp power key) to continue...
```



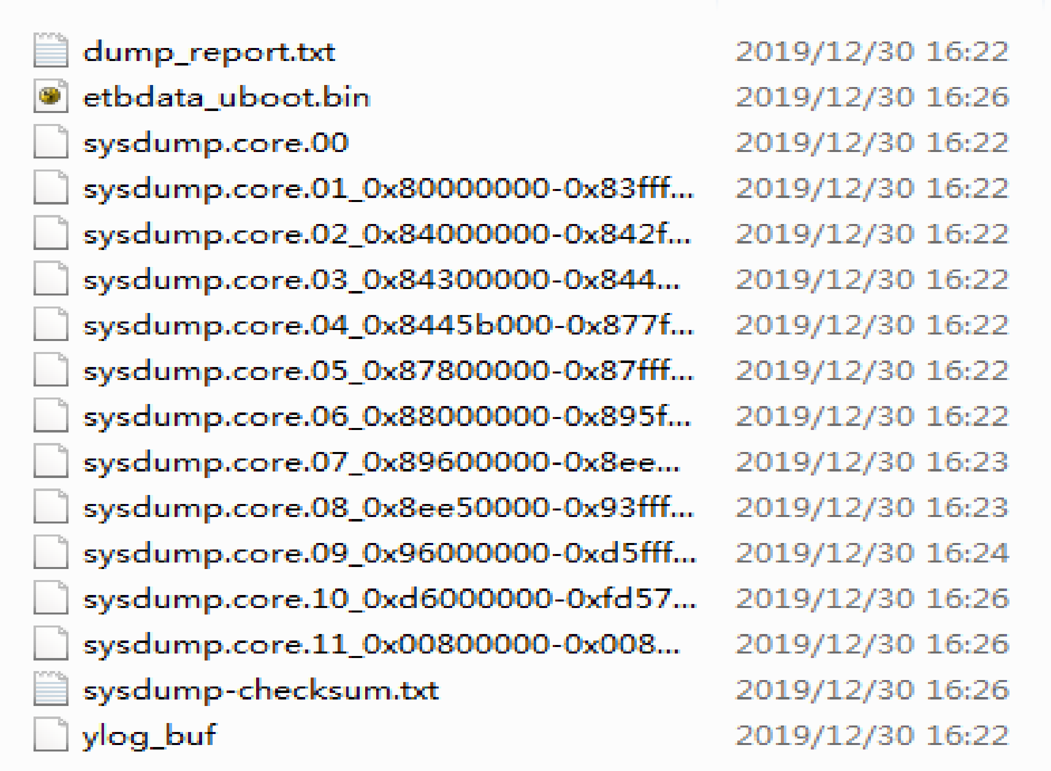
```
 . . . . . . . . . . . . . . . . .
 > dump ap memory file end: total received: 23068672 bytes, lost: 0 bytes
 >
 > dump ap memory file begin: sysdump.core.07_0x89600000-0x8ee4ffff_dump.lst,
size: 92602368 bytes
 >
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . .
 > dump ap memory file end: total received: 92602368 bytes, lost: 0 bytes
 >
 > dump ap memory file begin: sysdump.core.08_0x8ee50000-0x93ffffff_dump.lst,
size: 85655552 bytes
 >
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . .
 > dump ap memory file end: total received: 85655552 bytes, lost: 0 bytes
 >
 > dump ap memory file begin: sysdump.core.09_0x96000000-0xd5ffffff_dump.lst,
size: 1073741824 bytes
 >
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . .
 > dump ap memory file end: total received: 1073741824 bytes, lost: 0 bytes
 >
 > dump ap memory file begin: sysdump.core.10_0xd6000000-0xfd57ffff_dump.lst,
size: 660078592 bytes
 >
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . .
 > dump ap memory file end: total received: 660078592 bytes, lost: 0 bytes
 >
 > dump ap memory file begin: sysdump.core.11_0x00800000-0x0083ffff_dump.lst,
size: 262144 bytes
 > . . . . . .
 > dump ap memory file end: total received: 262144 bytes, lost: 0 bytes
 >
 > dump ap memory file begin: sysdump-checksum.txt, size: 937 bytes
 > .
 > dump ap memory file end: total received: 937 bytes, lost: 0 bytes
 >
 > dump ap memory file begin: etbdata_uboot.bin, size: 32768 bytes
 > .
 > dump ap memory file end: total received: 32768 bytes, lost: 0 bytes
 >
 > Total sysdump finished!
 > |
```

Dump2PC Instructions (6/6)

6. Check Sysdump log files.

Sysdump log files are saved in ap_sys_dump folder of XXX_armlog named after the current time under the /bin/history/directory of PC Logel tool’s unzip root directory.

The interface on PC is shown as follows.



dump_report.txt	2019/12/30 16:22
etbdata_uboot.bin	2019/12/30 16:26
sysdump.core.00	2019/12/30 16:22
sysdump.core.01_0x80000000-0x83fff...	2019/12/30 16:22
sysdump.core.02_0x84000000-0x842f...	2019/12/30 16:22
sysdump.core.03_0x84300000-0x844...	2019/12/30 16:22
sysdump.core.04_0x8445b000-0x877f...	2019/12/30 16:22
sysdump.core.05_0x87800000-0x87fff...	2019/12/30 16:22
sysdump.core.06_0x88000000-0x895f...	2019/12/30 16:22
sysdump.core.07_0x89600000-0x8ee...	2019/12/30 16:23
sysdump.core.08_0x8ee50000-0x93fff...	2019/12/30 16:23
sysdump.core.09_0x96000000-0xd5fff...	2019/12/30 16:24
sysdump.core.10_0xd6000000-0xfd57...	2019/12/30 16:26
sysdump.core.11_0x00800000-0x008...	2019/12/30 16:26
sysdump-checksum.txt	2019/12/30 16:26
ylog_buf	2019/12/30 16:22

05

Minidump

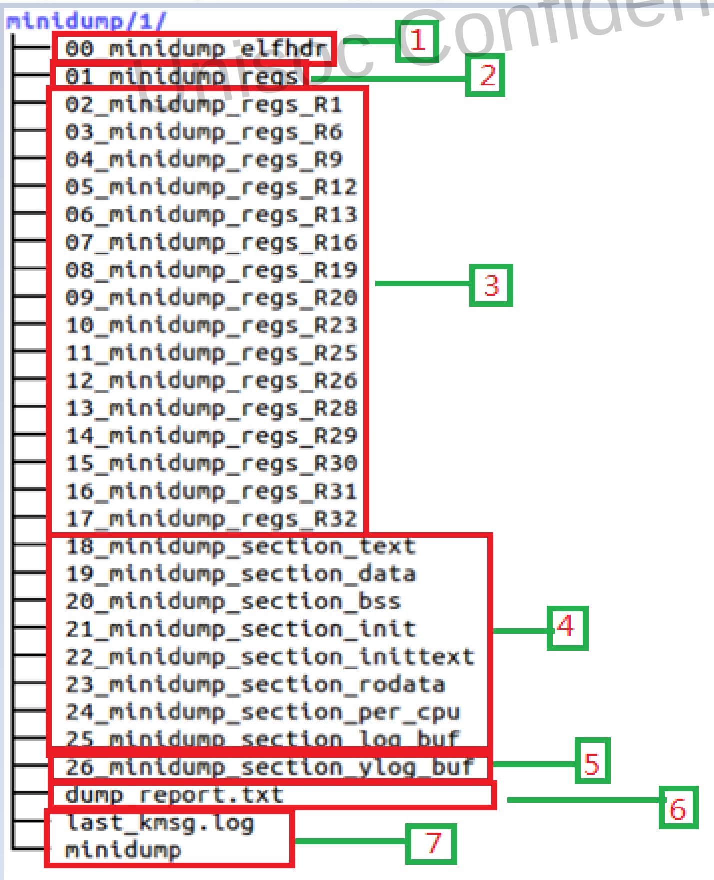
Minidump

- Minidump is usually used in no-SD card condition or Dump2PC is not available. Minidump is enabled by default.
- Minidump storage:
  - ✓ Android 11.0 is saved in /data/Minidump by default.
  - ✓ Android 10.0 is saved in /sdcard/Minidump by default.
- Minidump log is analyzed by TRACE32 ARM SIMULATOR (T32 sim) or Crash tool.
- Currently, Minidump supports the storage of 5 pieces of historical data. There are five file folders, namely 1, 2, 3, 4, 5 under Minidump file folder. They are all used to store compressed data of Minidump. Folder 1 is always the most recently generated data. When Minidump occurs again, name folder 1 as folder 2, and recreate folder 1. when there exists more than 5 folders, the original folder 5 will be discarded, and so on.

Minidump log can be exported directly by adb pull command, and the new version of log export tool has added the path /data/ Minidump by default (the path is /sdcard/ Minidump in Android 10.0).

Minidump Log Files (1/2)

Log files generated by Minidump are shown as below:



```
minidump/1/
├── 00_minidump_elfhdr
├── 01_minidump_regs
├── 02_minidump_regs_R1
├── 03_minidump_regs_R6
├── 04_minidump_regs_R9
├── 05_minidump_regs_R12
├── 06_minidump_regs_R13
├── 07_minidump_regs_R16
├── 08_minidump_regs_R19
├── 09_minidump_regs_R20
├── 10_minidump_regs_R23
├── 11_minidump_regs_R25
├── 12_minidump_regs_R26
├── 13_minidump_regs_R28
├── 14_minidump_regs_R29
├── 15_minidump_regs_R30
├── 16_minidump_regs_R31
├── 17_minidump_regs_R32
├── 18_minidump_section_text
├── 19_minidump_section_data
├── 20_minidump_section_bss
├── 21_minidump_section_init
├── 22_minidump_section_inittext
├── 23_minidump_section_rodata
├── 24_minidump_section_per_cpu
├── 25_minidump_section_log_buf
├── 26_minidump_section_ylog_buf
├── dump_report.txt
├── last_kmsg.log
└── minidump
```

1. 00_Minidump_elfhdr： elf file header.
2. 01_Minidump_regs： CPU registers.
3. 02~17： 256 bytes of registers.
4. 18~25： each section’s information.
5. 26_Minidump_section_ylog_buf： Ylog buf in panic.
6. dump_report： dump report information.
7. last_kmsg.log, Minidump： generated by script, and keeps the last kernel log information before the system restart.

Minidump Log Files (2/2)

Files exported by Minidump are in compressed format, use the tool **unisoc_parse_dumplog.py** to unzip them.

The test environment supported by this tool is as follows:

- Python 2.7.6 and 3.7.1 test pass on Ubuntu.
- Python 2.7.1 and 3.7.1 test pass on Windows.

Main functions of this tool:

- Unzip the exported zip files.
- Parse the generated file last_kmsg.log.
- Collect Minidump files for T32 sim or crash tool analysis.

How to use this tool in Linux or windows command:

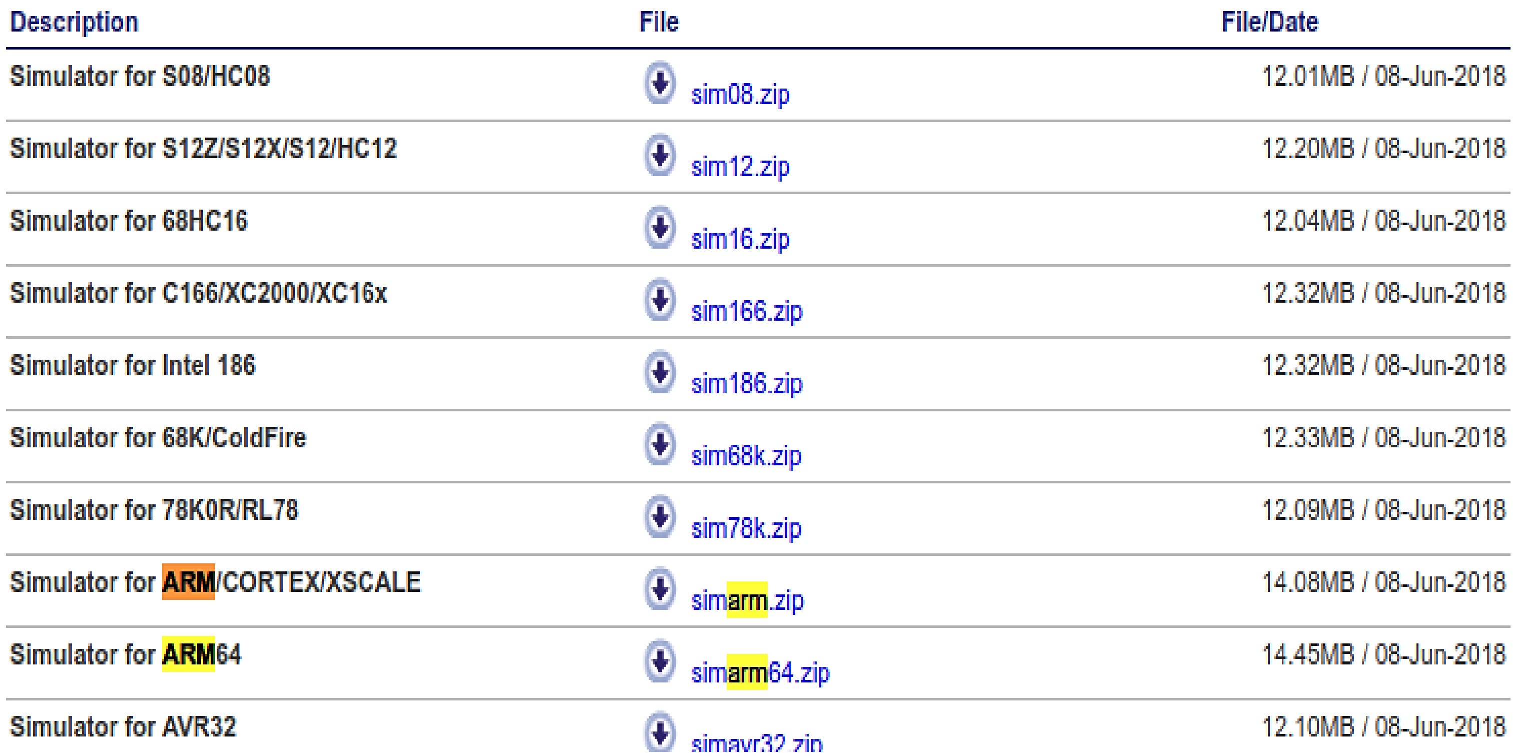
- Command: python unisoc_parse_dumplog.py *xxxx/Minidump/1*

  xxxx/Minidump/1 is the directory of Minidump compressed file.
- Copy the tool directly to the same directory where the compressed file resides and execute directly.

Minidump Log Parse_T32 sim (1/2)

Use T32 sim to analyze the log stored by Minidump.

T32 sim official website: https://www.lauterbach.com/frames.html?Home.html



Description	File	File/Date
Simulator for S08/HC08	sim08.zip	12.01MB / 08-Jun-2018
Simulator for S12Z/S12X/S12/HC12	sim12.zip	12.20MB / 08-Jun-2018
Simulator for 68HC16	sim16.zip	12.04MB / 08-Jun-2018
Simulator for C166/XC2000/XC16x	sim166.zip	12.32MB / 08-Jun-2018
Simulator for Intel 186	sim186.zip	12.32MB / 08-Jun-2018
Simulator for 68K/ColdFire	sim68k.zip	12.33MB / 08-Jun-2018
Simulator for 78K0R/RL78	sim78k.zip	12.09MB / 08-Jun-2018
Simulator for ARM/CORTEX/XSCALE	simarm.zip	14.08MB / 08-Jun-2018
Simulator for ARM64	simarm64.zip	14.45MB / 08-Jun-2018
Simulator for AVR32	simavr32.zip	12.10MB / 08-Jun-2018

Minidump Log Parse_T32 sim(2/2)

Tools and files needed:

- unisoc_parse_dumplog.py
- simarm64 or simarm
- unisoc_Debug.cmm script

Analyze procedure:

Step 1: Use tool unisoc_parse_dumplog.py to collect Minidump files.

Step 2: Download simarm.zip (simarm64.zip in ARM64) and unzip.

Step 3: Add TRACE32 script and the composed files in step 1 to the unzipped folder of step 2.

Step 4: Execute t32marm.exe in the unzipped folder.

Windows: double-click to execute.

Linux: install wine and execute the command wine t32marm.exe.

Step 5: Start to the default interface, click **File ->Run Script** , choose the script unisoc_Debug.cmm and execute.

Step 6: Analyze the execution results of unisoc_Debug.cmm script.

Minidump Log Parse_Crash Tool

Crash tool“minimal” mode

Currently, the exported Minidump files support crash tool “minimal” mode analysis. Commands are as follows:

- ARM: crash_arm Minidump vmlinux –minimal
- ARM64: crash_arm64 -m kimage_voffset=0xffffff7f88000000 Minidump vmlinux –minimal

In this mode, use command “log” to parse Kernel Logbuf completely .

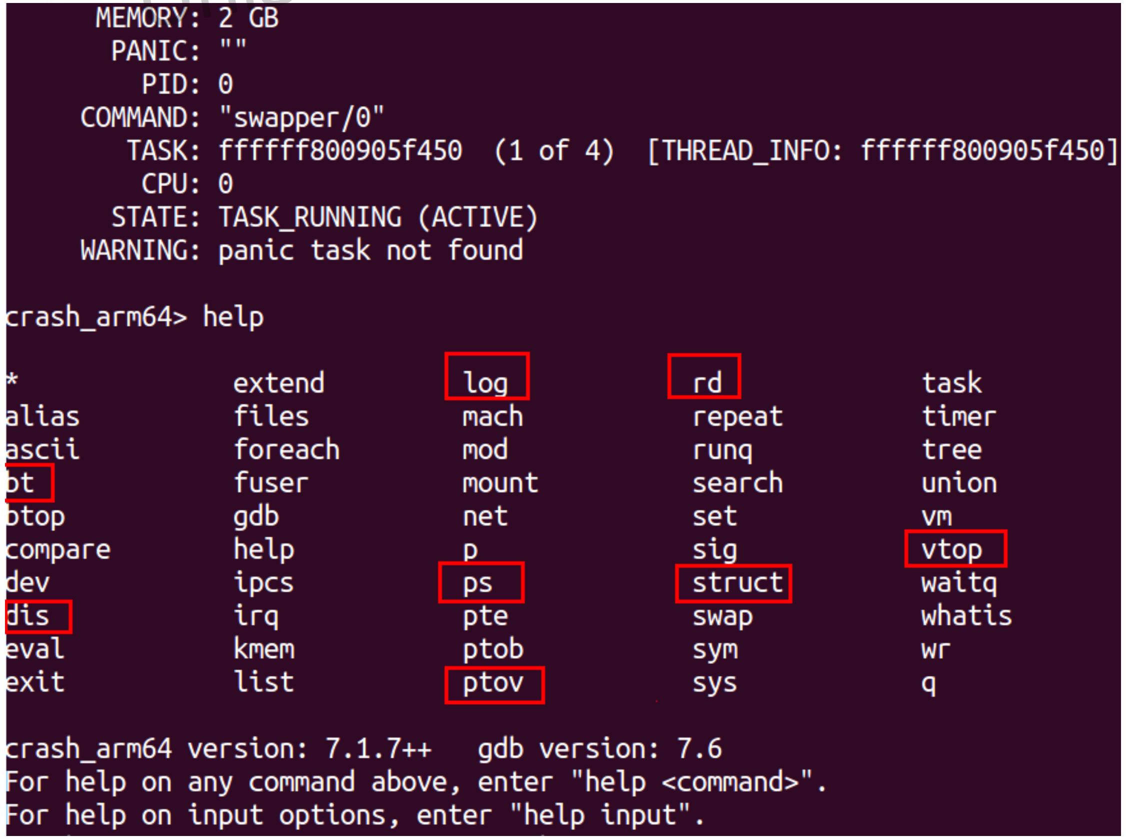
In addition, crash tool also supports other commands such as sym, rd to make a simple analysis of Minidump files.

06

Commands of Crash Tool

help

Enter the crash tool interface, input command “help” to view all supported commands of crash tool. Input command “help cmd” to view each command usage, such as command “help ps”.

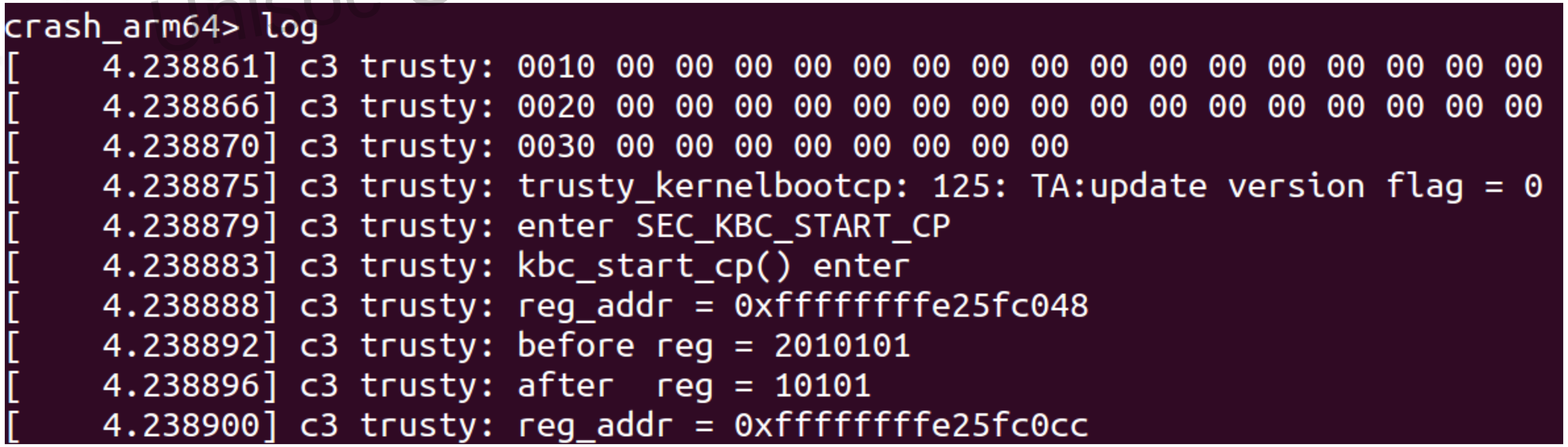


```
      MEMORY: 2 GB
       PANIC: ""
         PID: 0
     COMMAND: "swapper/0"
        TASK: ffffff800905f450  (1 of 4)  [THREAD_INFO: ffffff800905f450]
         CPU: 0
       STATE: TASK_RUNNING (ACTIVE)
     WARNING: panic task not found

crash_arm64> help

*              extend         log            rd             task
alias          files          mach           repeat         timer
ascii          foreach        mod            runq           tree
bt             fuser          mount          search         union
btop           gdb            net            set            vm
compare        help           p              sig            vtop
dev            ipcs           ps             struct         waitq
dis            irq            pte            swap           whatis
eval           kmem           ptob           sym            wr
exit           list           ptov           sys            q

crash_arm64 version: 7.1.7++   gdb version: 7.6
For help on any command above, enter "help <command>".
For help on input options, enter "help input".
```

log

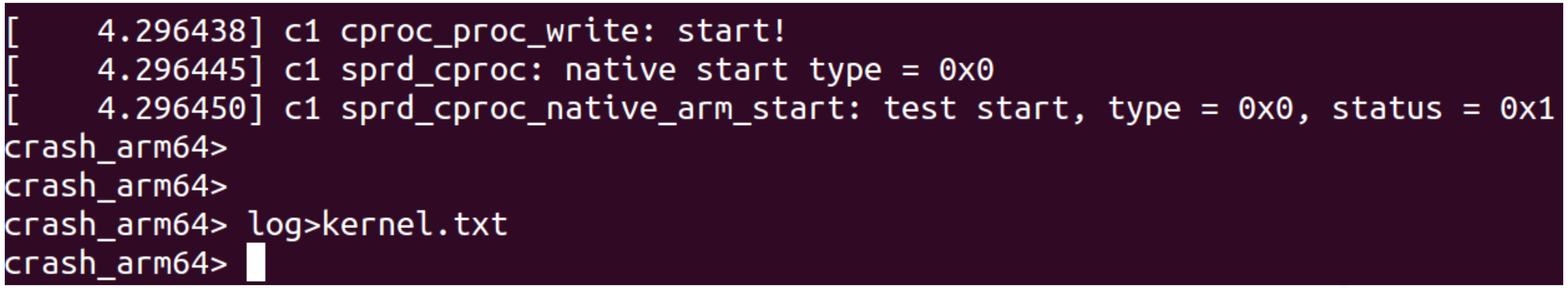
Command “log”

- It can dump kernel log of __log_buf



```
crash_arm64> log
[    4.238861] c3 trusty: 0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[    4.238866] c3 trusty: 0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[    4.238870] c3 trusty: 0030 00 00 00 00 00 00 00 00
[    4.238875] c3 trusty: trusty_kernelbootcp: 125: TA:update version flag = 0
[    4.238879] c3 trusty: enter SEC_KBC_START_CP
[    4.238883] c3 trusty: kbc_start_cp() enter
[    4.238888] c3 trusty: reg_addr = 0xffffffffe25fc048
[    4.238892] c3 trusty: before reg = 2010101
[    4.238896] c3 trusty: after  reg = 10101
[    4.238900] c3 trusty: reg_addr = 0xffffffffe25fc0cc
```

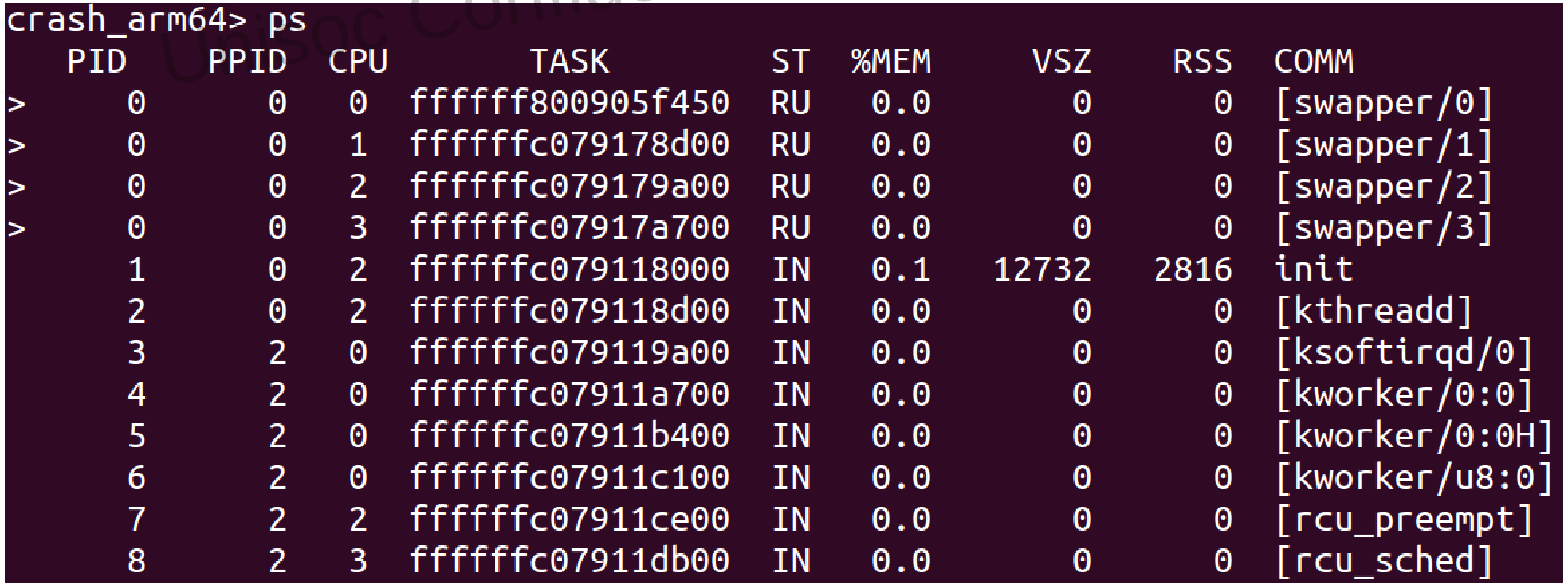
- It can redirect the output content of log to one file for later view and analysis.



```
[    4.296438] c1 cproc_proc_write: start!
[    4.296445] c1 sprd_cproc: native start type = 0x0
[    4.296450] c1 sprd_cproc_native_arm_start: test start, type = 0x0, status = 0x1
crash_arm64>
crash_arm64>
crash_arm64> log>kernel.txt
crash_arm64> 
```

ps

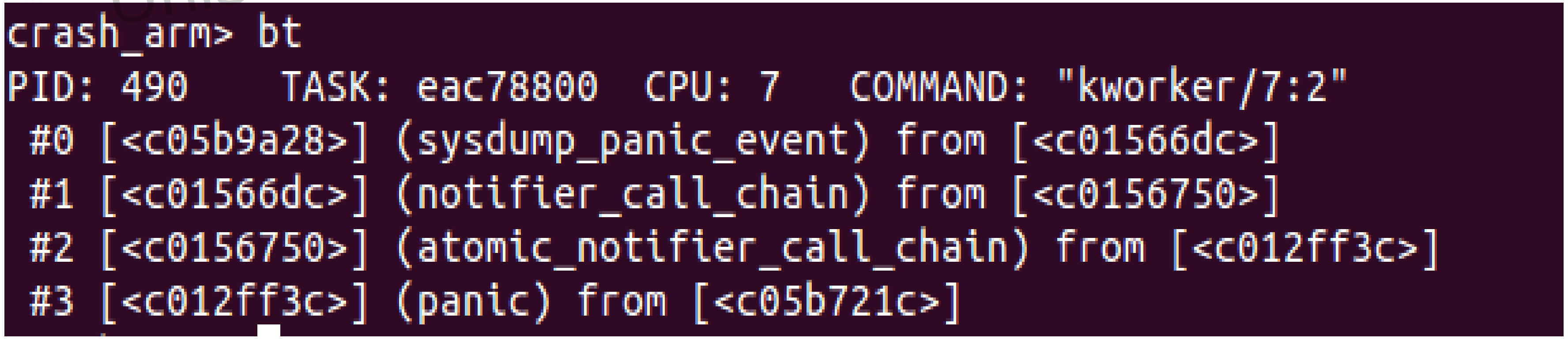
Command "ps" can list all threads and their states, etc.



```
crash_arm64> ps
   PID    PPID  CPU       TASK        ST  %MEM     VSZ    RSS  COMM
>     0      0   0  ffffff800905f450  RU   0.0       0      0  [swapper/0]
>     0      0   1  ffffffc079178d00  RU   0.0       0      0  [swapper/1]
>     0      0   2  ffffffc079179a00  RU   0.0       0      0  [swapper/2]
>     0      0   3  ffffffc07917a700  RU   0.0       0      0  [swapper/3]
      1      0   2  ffffffc079118000  IN   0.1   12732   2816  init
      2      0   2  ffffffc079118d00  IN   0.0       0      0  [kthreadd]
      3      2   0  ffffffc079119a00  IN   0.0       0      0  [ksoftirqd/0]
      4      2   0  ffffffc07911a700  IN   0.0       0      0  [kworker/0:0]
      5      2   0  ffffffc07911b400  IN   0.0       0      0  [kworker/0:0H]
      6      2   0  ffffffc07911c100  IN   0.0       0      0  [kworker/u8:0]
      7      2   2  ffffffc07911ce00  IN   0.0       0      0  [rcu_preempt]
      8      2   3  ffffffc07911db00  IN   0.0       0      0  [rcu_sched]
```

bt

Command “bt” displays a task’s Kernel-stack back trace.



```
crash_arm> bt
PID: 490    TASK: eac78800  CPU: 7   COMMAND: "kworker/7:2"
 #0 [<c05b9a28>] (sysdump_panic_event) from [<c01566dc>]
 #1 [<c01566dc>] (notifier_call_chain) from [<c0156750>]
 #2 [<c0156750>] (atomic_notifier_call_chain) from [<c012ff3c>]
 #3 [<c012ff3c>] (panic) from [<c05b721c>]
```

07

Exception Handling

Exception Handling

- Sysdump process failed

  The Sysdump process is displayed on the screen with a prompt message.

  Most of Sysdump exceptions are SD card exceptions, try to replace the SD card or use Dump2PC mode.

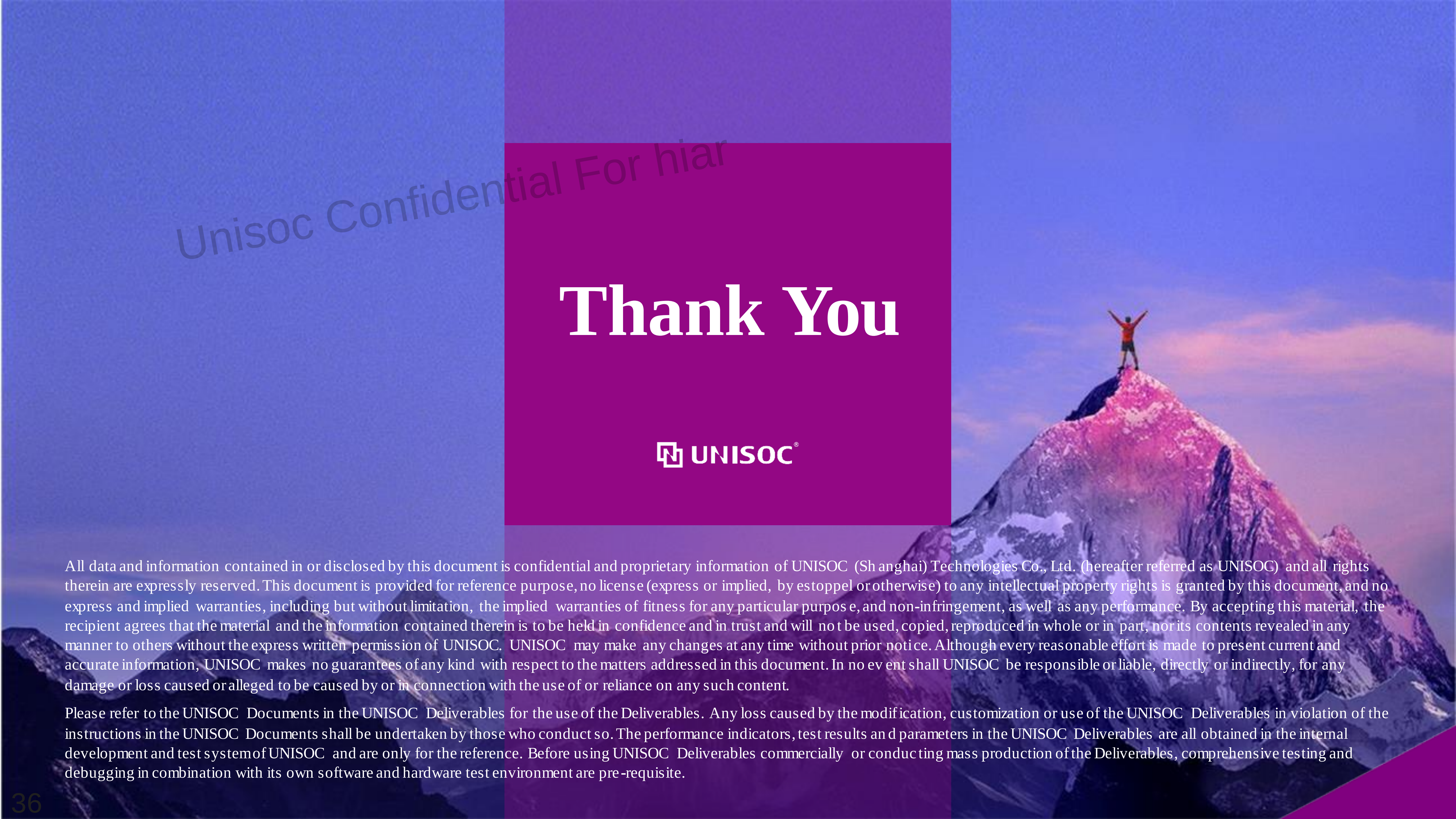
- Fulldump parsing failed

  The screen displays **“crash: vmlinux and vmcore do not match!”**

  Sysdump files and vmlinux do not match. Use the following two commands to see if the fetch time is consistent.

  – strings vmcore |grep "Linux version"

  – strings vmlinux |grep "Linux version"



Thank You

UNISOC®

All data and information contained in or disclosed by this document is confidential and proprietary information of UNISOC (Shanghai) Technologies Co., Ltd. (hereafter referred as UNISOC) and all rights therein are expressly reserved. This document is provided for reference purpose, no license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document, and no express and implied warranties, including but without limitation, the implied warranties of fitness for any particular purpose, and non-infringement, as well as any performance. By accepting this material, the recipient agrees that the material and the information contained therein is to be held in confidence and in trust and will not be used, copied, reproduced in whole or in part, nor its contents revealed in any manner to others without the express written permission of UNISOC. UNISOC may make any changes at any time without prior notice. Although every reasonable effort is made to present current and accurate information, UNISOC makes no guarantees of any kind with respect to the matters addressed in this document. In no event shall UNISOC be responsible or liable, directly or indirectly, for any damage or loss caused or alleged to be caused by or in connection with the use of or reliance on any such content.

Please refer to the UNISOC Documents in the UNISOC Deliverables for the use of the Deliverables. Any loss caused by the modification, customization or use of the UNISOC Deliverables in violation of the instructions in the UNISOC Documents shall be undertaken by those who conduct so. The performance indicators, test results and parameters in the UNISOC Deliverables are all obtained in the internal development and test systemof UNISOC and are only for the reference. Before using UNISOC Deliverables commercially or conducting mass production of the Deliverables, comprehensive testing and debugging in combination with its own software and hardware test environment are pre-requisite.