OPEN VIRTUALIZATION

BUILD AND BOOT GUIDE

Table of Contents

1 Build and booting Open Virtualization in Fastmodel (Single Guest)

1.1 Steps to boot Open Virtualization in Fastmodel (Single Guest)

1.1.1 Install the Fastmodel

Fast Model can be downloaded from the following link by using an ARM user account. (First time users need a registration).

http://www.arm.com/products/tools/models/fast-models.php

The option Download Now on clicking navigates to the page where the list of products available for download will appear.

Select Development Tools-> Fast Models from the products

and download Fast Models Evaluation, Fast Models Third Party IP and the license. Install the Fast Models Evaluation and the third party IP.

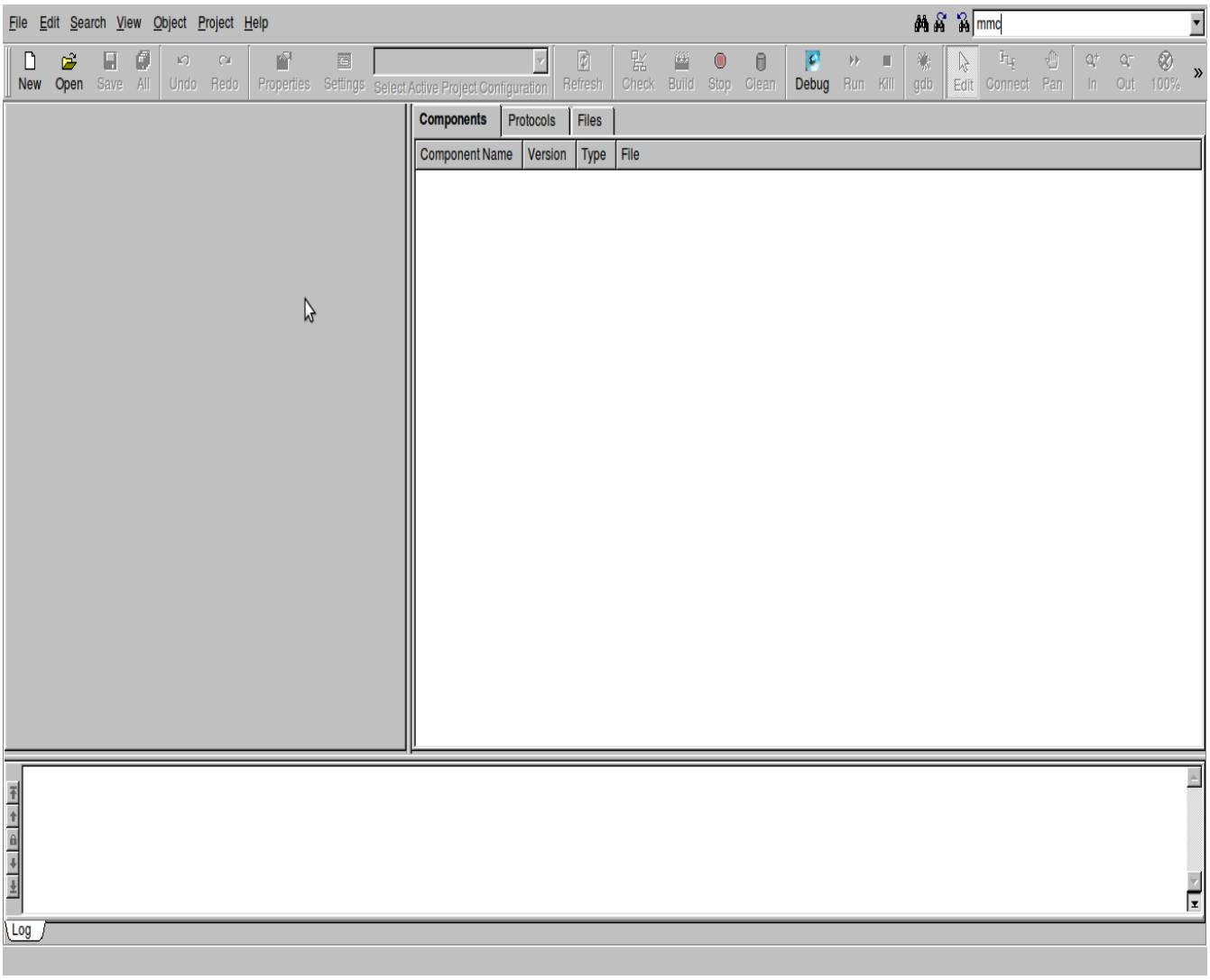
1.1.2 Starting the Fastmodel

Run the following commands in order to launch the fast model.

#source <FastModelInstalled-Dir>/FastModelTools_7.1/source_all.sh

#sgcanvas

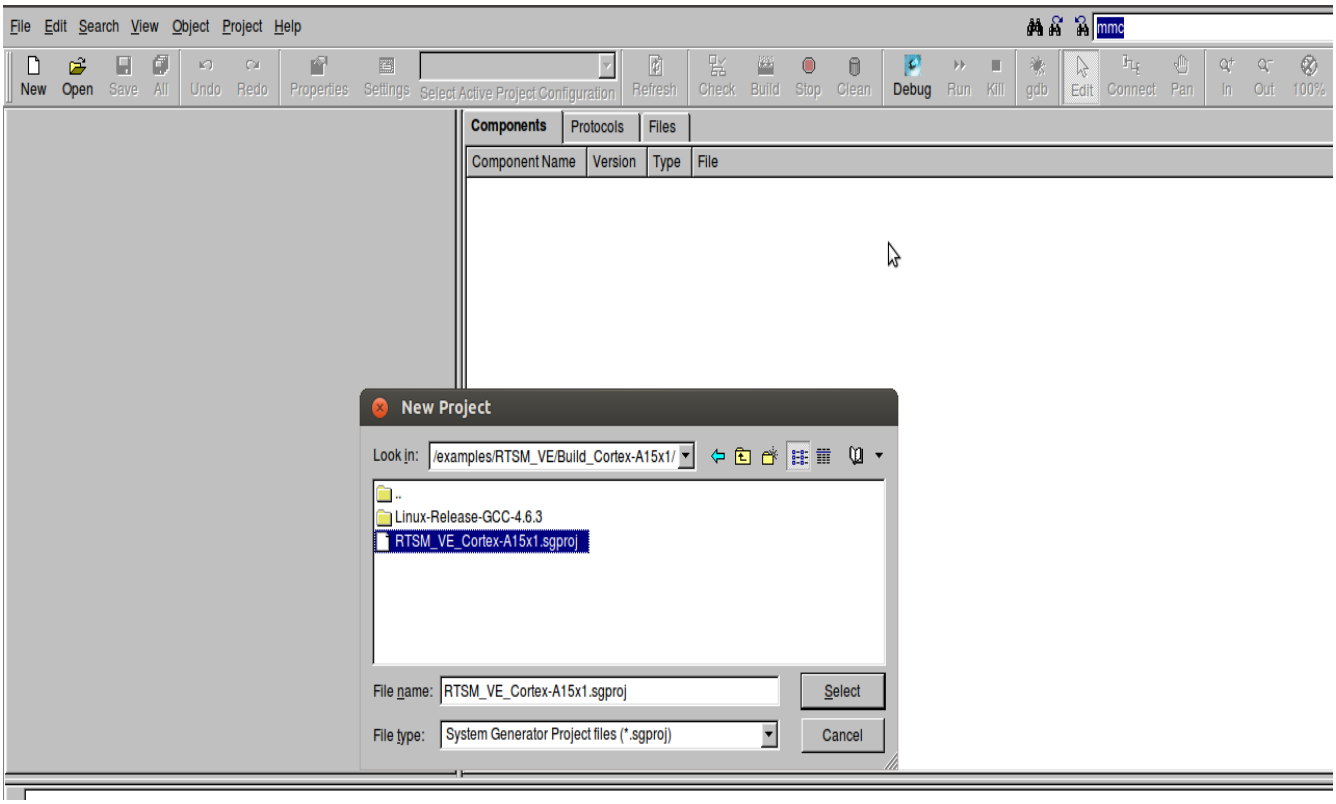
The application gets started and the following screen appears.



1.1.3 Load project and debugging

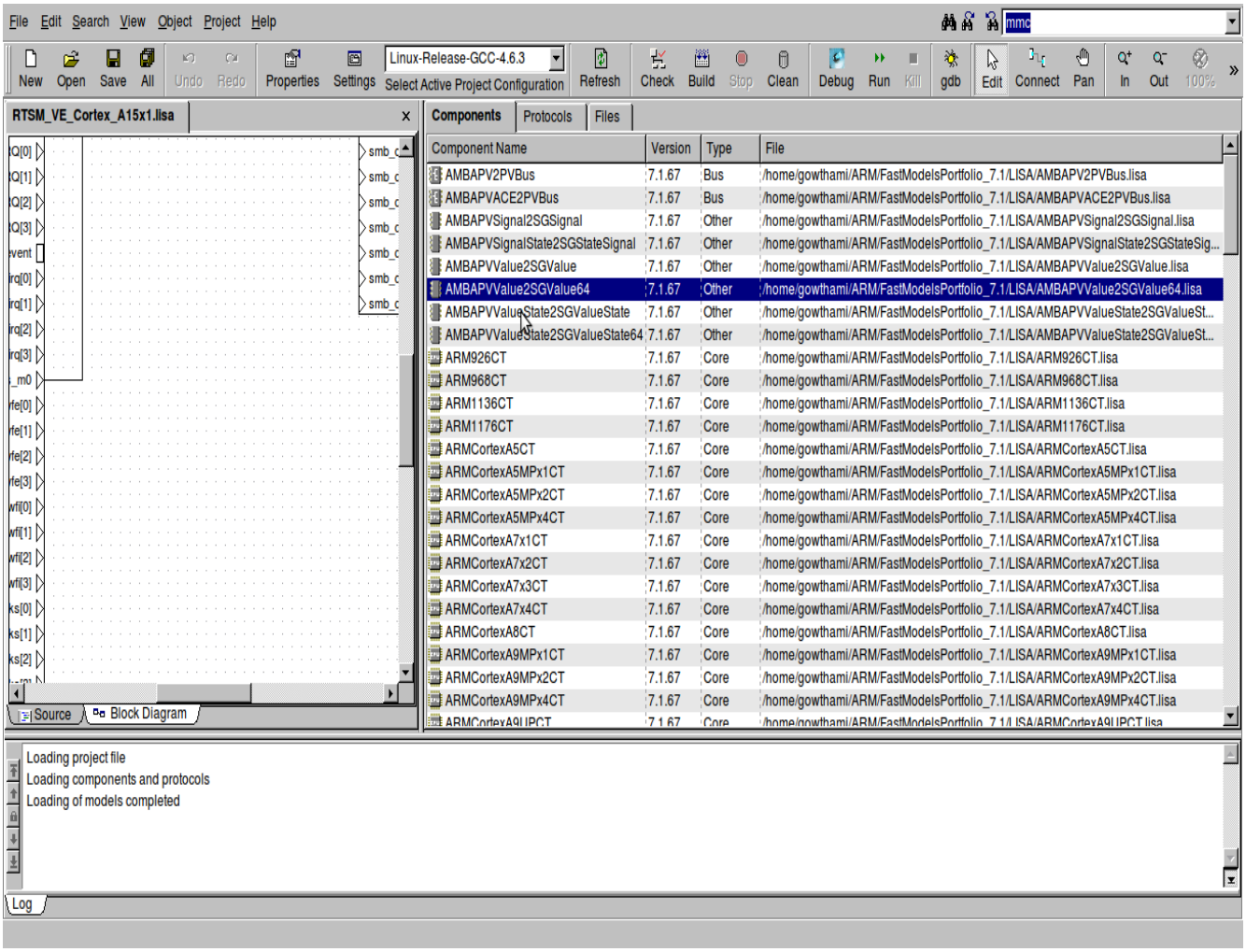
Load the required project by selecting File -> Load Project

For example for using Realview Eval board with cortex A15 select the following.
<FastModel install dir>/
FastModelsPortfolio_7.1/examples/RTSM_VE/Build_Cortex-A15x1/
RTSM_VE_Cortex-A15x1.sgproj



Example projects are available in the Fastmodel installed directory.

The following screen appears on loading the project.



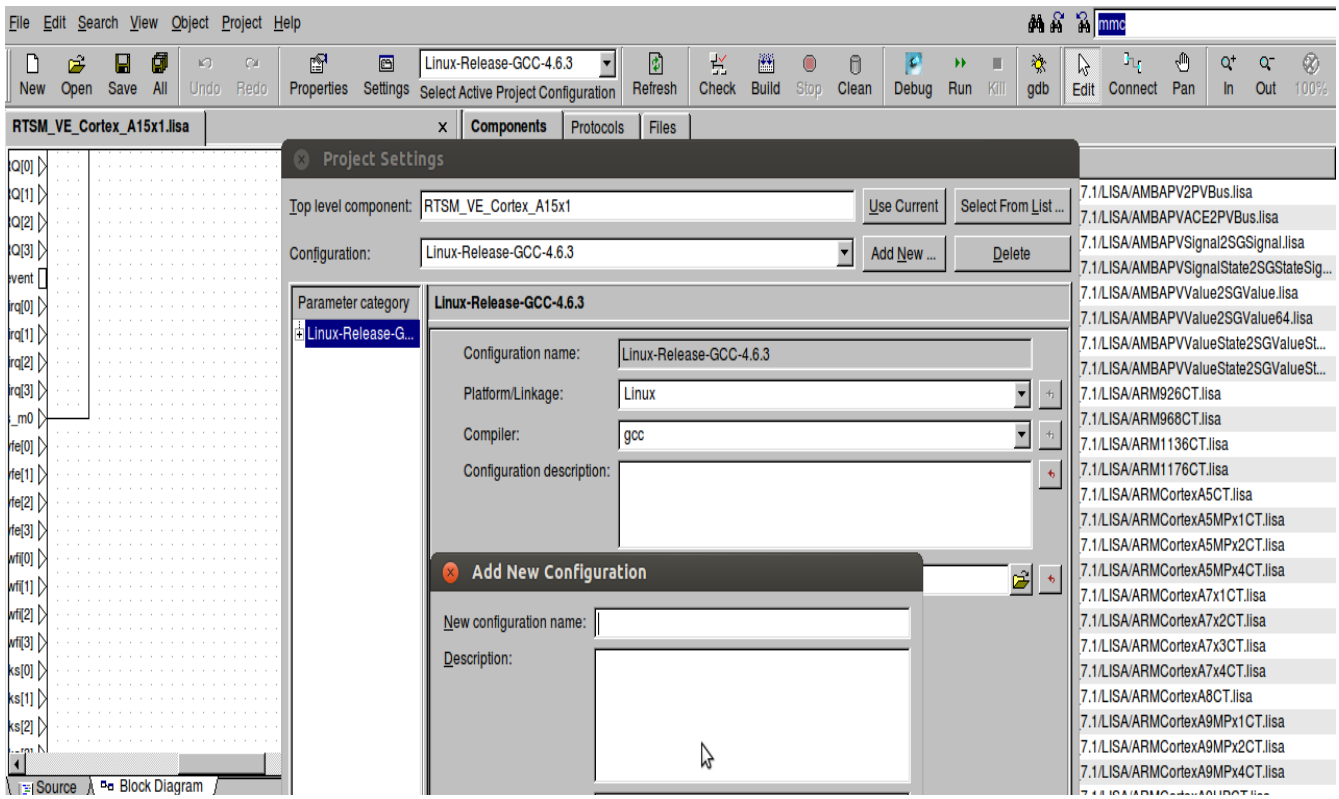
On trying to build (clicking Build option) the project with GCC 4.7.2,few .so files are not created which results in the Model Debugger to throw an error similar to the following.

Cannot load model library '<Fastmodel-install dir>/FastModelsPortfolio_7.1/examples/RTSM_EB/Build_Cortex-A8/Linux-Release-GCC-4.7/cadi_system_Linux-Release-GCC-4.7.so':
Error while loading lib:

<Fastmodel-install dir>/FastModelsPortfolio_7.1/examples/RTSM_EB/Build_Cortex-A8/Linux-Release-GCC-4.7/cadi_system_Linux-Release-GCC-4.7.so:
undefined symbol: _ZN2sg13ConnectorBase5emptyEv

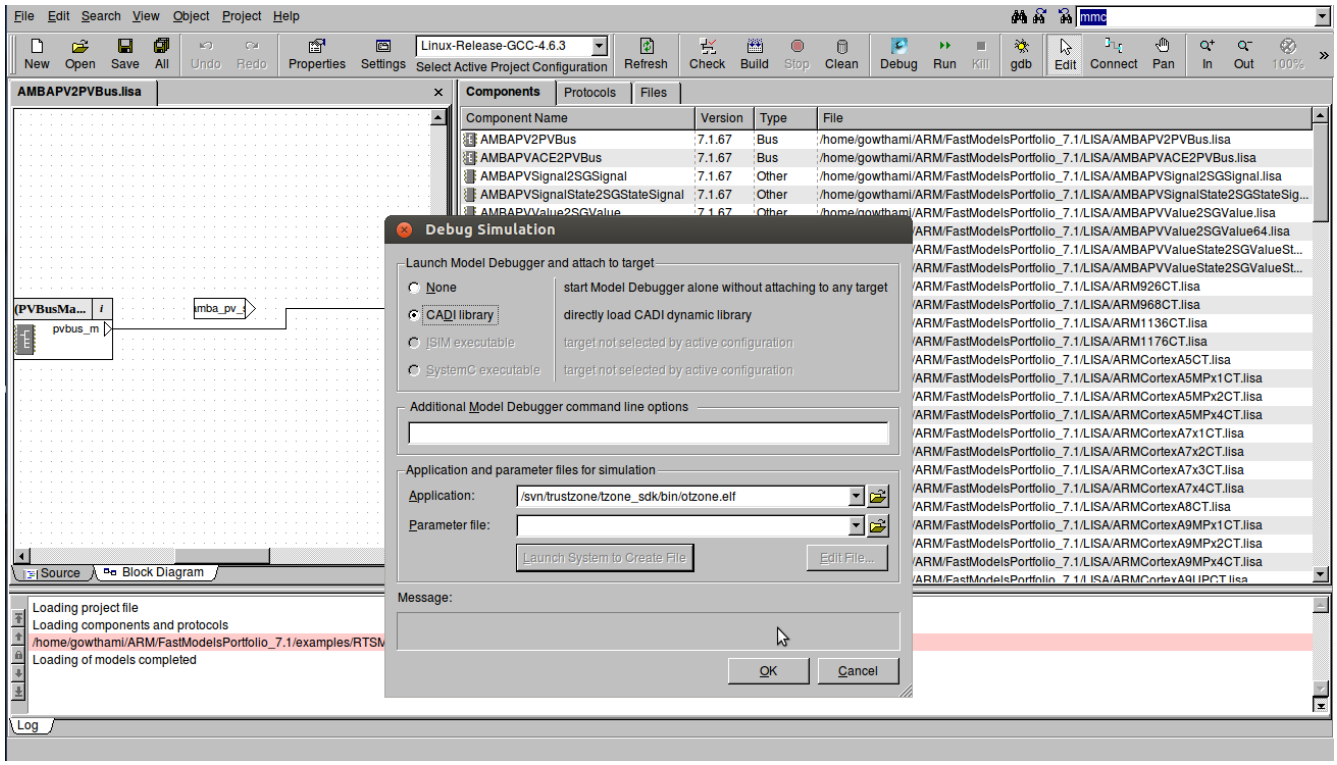
This error can be avoided by using GCC 4.6.3(or earlier versions) to carry out the build.

The GCC version for building the project can be changed by selecting the Settings option and then by adding a new configuration.



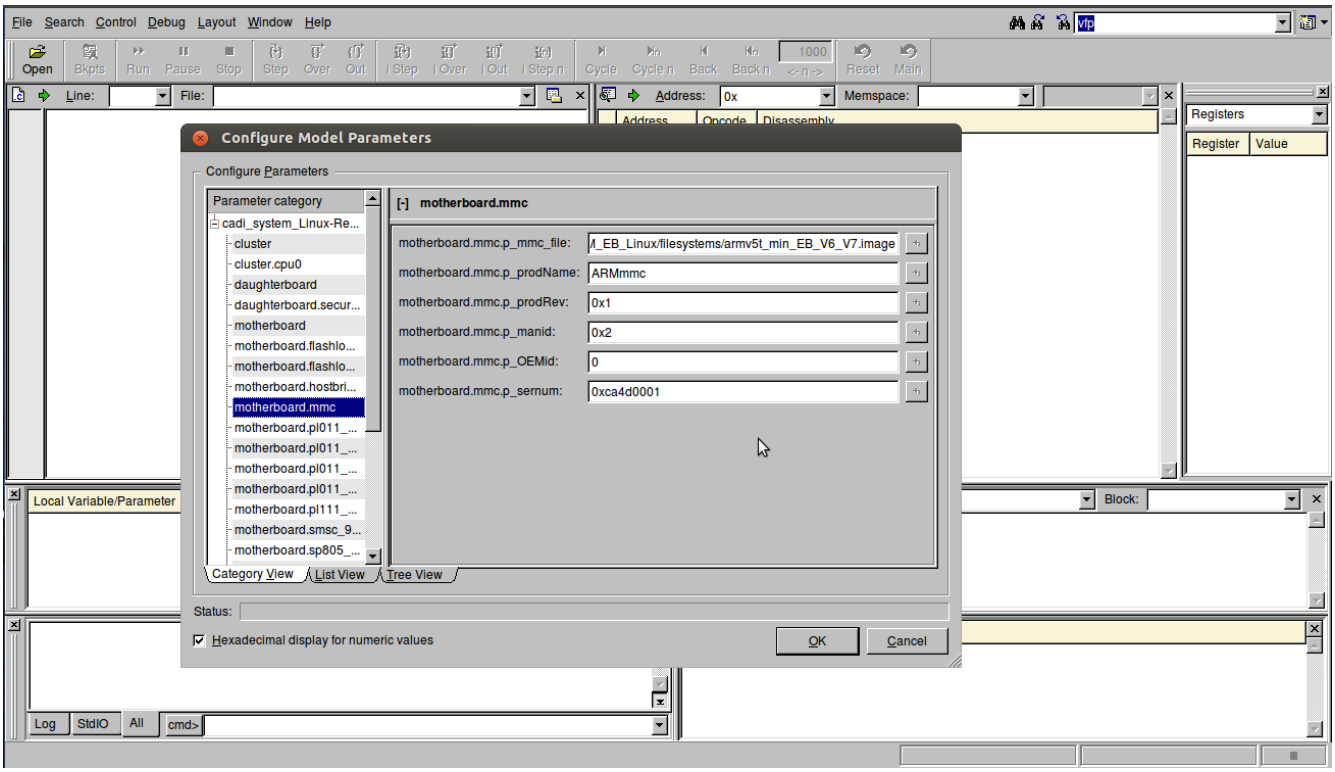
1.1.4 Booting the image and debugging using Fastmodel

On clicking Debug, a Debug Simulation window appears which allows the required application to be loaded. Select otzone.elf and click ok.

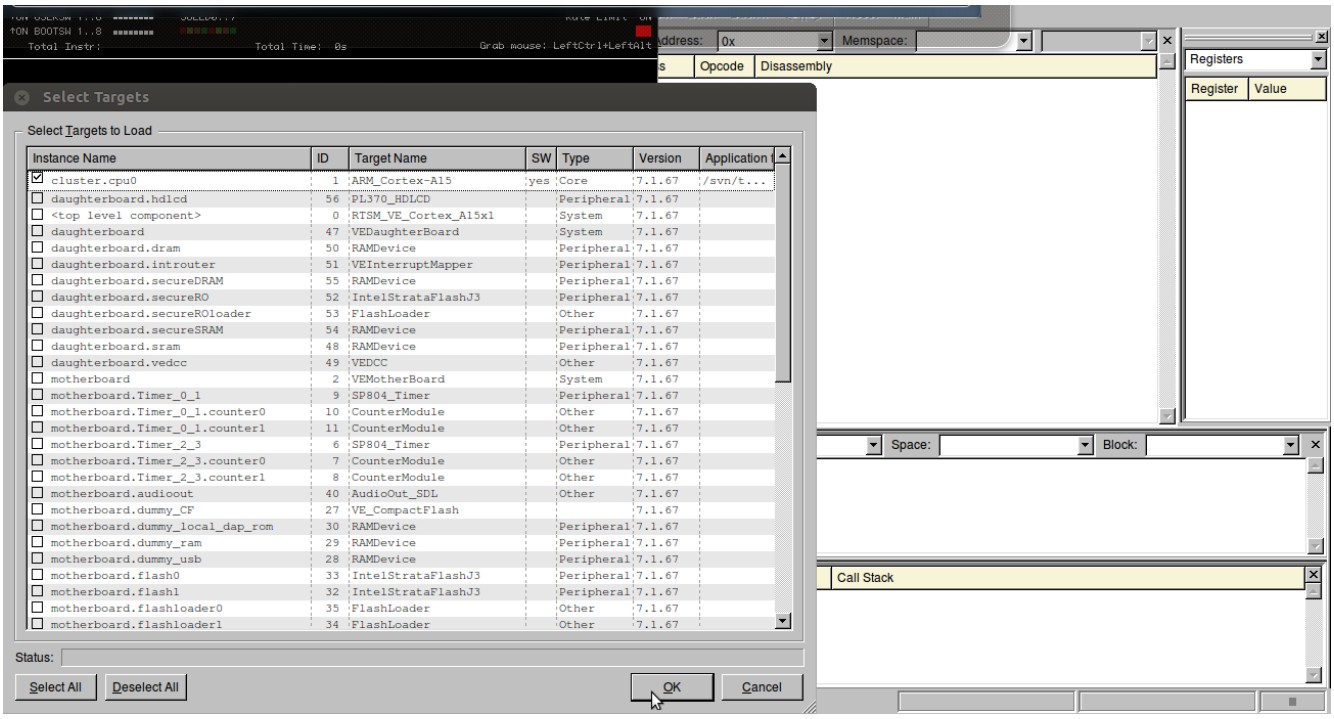


The Configure Model Parameters window gets popped up and change the motherboard.mmc to point to the following root system image.

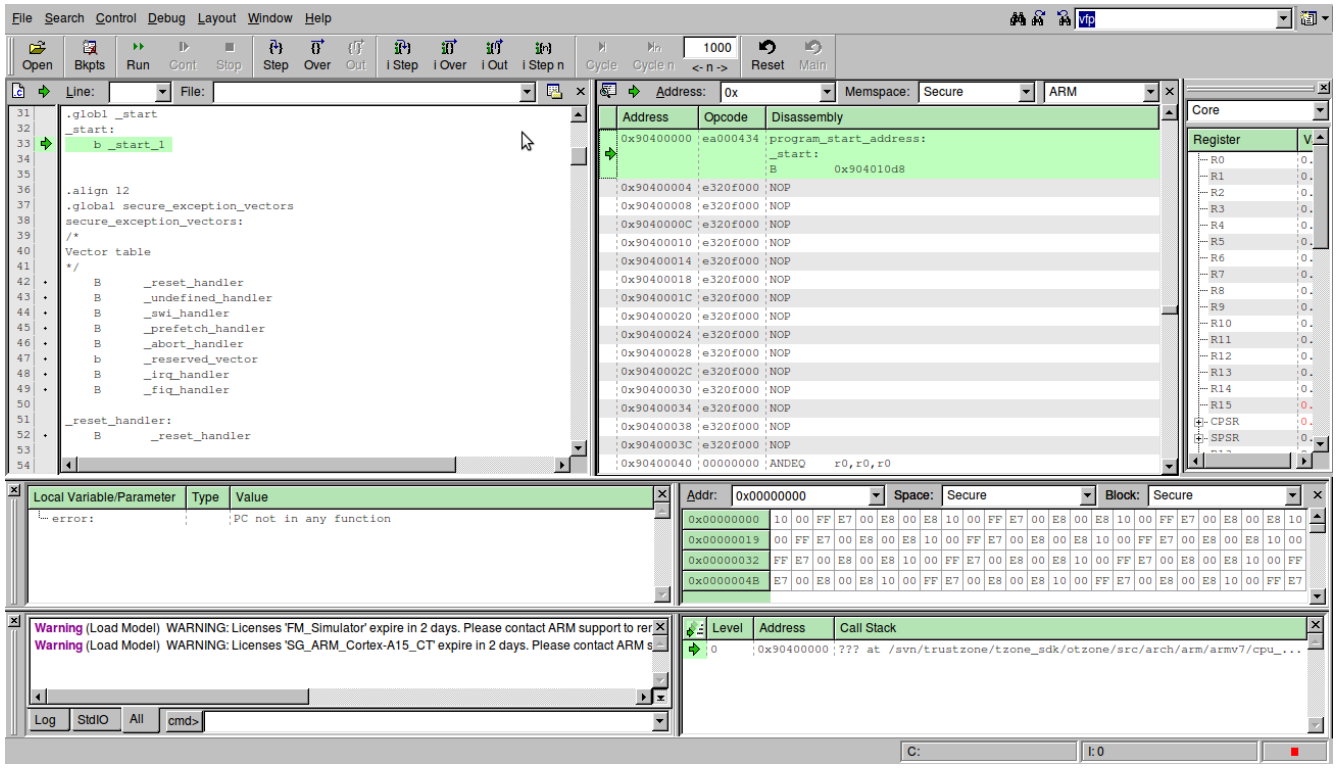
<FastModelInstalldir>/FastModelsPortfolio_7.1/images/RTSM_EB_Linux/filesystems/armv5t_min_EB_V6_V7.image.



Next the Select Targets window will appear which is already configured.



After all the configuration has been done and proceeding forward by clicking ok, the Model Debugger window will be appearing in the screen. In this window, the run option would make the system to boot, reset will make the system to boot from the starting point.



Login as root.

```
RTSM terminal_0
Intel/Sharp Extended Query Table at 0x0031
Using buffer write method
armflash-1: Found 2 x16 devices at 0x0 in 32-bit bank. Manufacturer ID 0x000089
Chip ID 0x000018
Intel/Sharp Extended Query Table at 0x0031
Using buffer write method
armflash: multiple devices found but MTD concat support disabled.
smc91x: not found (-19).
mousedev: PS/2 mouse device common for all mice
mmci-pl18x mb:mmci: mmc0: PL180 manf 41 rev0 at 0x1c050000 irq 41,42 (pio)
TCP cubic registered
NET: Registered protocol family 17
VFP support v0.3: implementor 41 architecture 3 part 30 variant 9 rev 3
Registering SWP/SWPB emulation handler
mmc0: new MMC card at address 0001
mmcblk0: mmc0:0001 ARMmmc 256 MiB
 mmcblk0: unknown partition table
input: AT Raw Set 2 keyboard as /devices/mb:kmi0/serio0/input/input0
input: PS/2 Generic Mouse as /devices/mb:kmi1/serio1/input/input1
VFS: Mounted root (ext2 filesystem) readonly on device 179:0.
Freeing init memory: 160K
init started: BusyBox v1.14.3 (2009-11-12 11:03:55 GMT)
starting pid 37, tty '': '/etc/rc.d/rc.local'
/etc/rc.d/rc.local: line 14: can't create /var/testfile: Read-only file system
warning: can't open /etc/mtab: No such file or directory
S: devpts
S: udev
udevd (77): /proc/77/oom_adj is deprecated, please use /proc/77/oom_score_adj in
stead.
S: sshd
S: dbus id
Thu Jan  1 00:00:08 UTC 1970
S: hald
S: Xorg
R: Xorg
S: dhcdbd
Found no /etc/resolv.conf you need one for e.g. browser to resolve URLs
S: ohmd
sbrshd: Can't get address info
No network interface 'eth0' found
No network interface 'usb0' found
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 f
AEL login: root
login[330]: root login on 'ttyAMA0'


BusyBox v1.14.3 (2009-11-12 11:03:55 GMT) built-in shell (ash)
Enter 'help' for a list of built-in commands.

# 
```

1.1.5 Testing the Open-virtualization Stack

Load the Open-Trust-Zone kernel driver by executing the following instruction.

#insmod /otz_client.ko

Launch the test application.

#otzapp.elf

```
RTSM terminal_0
S: udev
udevd (77): /proc/77/oom_adj is deprecated, please use /proc/77/oom_score_adj in
stead.
S: sshd
S: dbus id
Thu Jan  1 00:00:08 UTC 1970
S: hald
S: Xorg
R: Xorg
S: dhcdbd
Found no /etc/resolv.conf you need one for e.g. browser to resolve URLs
S: ohmd
sbrshd: Can't get address info
No network interface 'eth0' found
No network interface 'usb0' found
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 f
AEL login: root
login[330]: root login on 'ttyAMA0'


BusyBox v1.14.3 (2009-11-12 11:03:55 GMT) built-in shell (ash)
Enter 'help' for a list of built-in commands.

# insmod /otz_client.ko
# otzapp.elf
fd :3
input data test data abcdefgh
out data ipc echo test: Test IPI echo cmd
Attempting to lock the variable
Lock successful. Trying to lock it one more time
Mutex already locked. We cannot lock it anymore !!
Unlock successful. Trying to lock it one more time
Going to sleep
Second time locking successful
=  test data abcdefgh and out data len 0x13
shared res buf addr 0x40167200, out data addr 0x40167200 and value  test_shared_
buffer and out data len 0x13
next decoder data type 0x2
res buf addr 0x4033f000, out data addr 0x4033f000 and value  test_array_space_bu
ffer and out data len 0x18
next decoder data type 0x0
device close successful
Creating task for testing otz mutexes
device close successful
Mutex testing finished
Creating task for testing secure kernel notification feature
echo task handler 0x115
device close successful
Notification testing finished
# 
RTSM terminal_0
```

TEE compliant app can be tested by launching the following application:

#otz_tee_app.elf

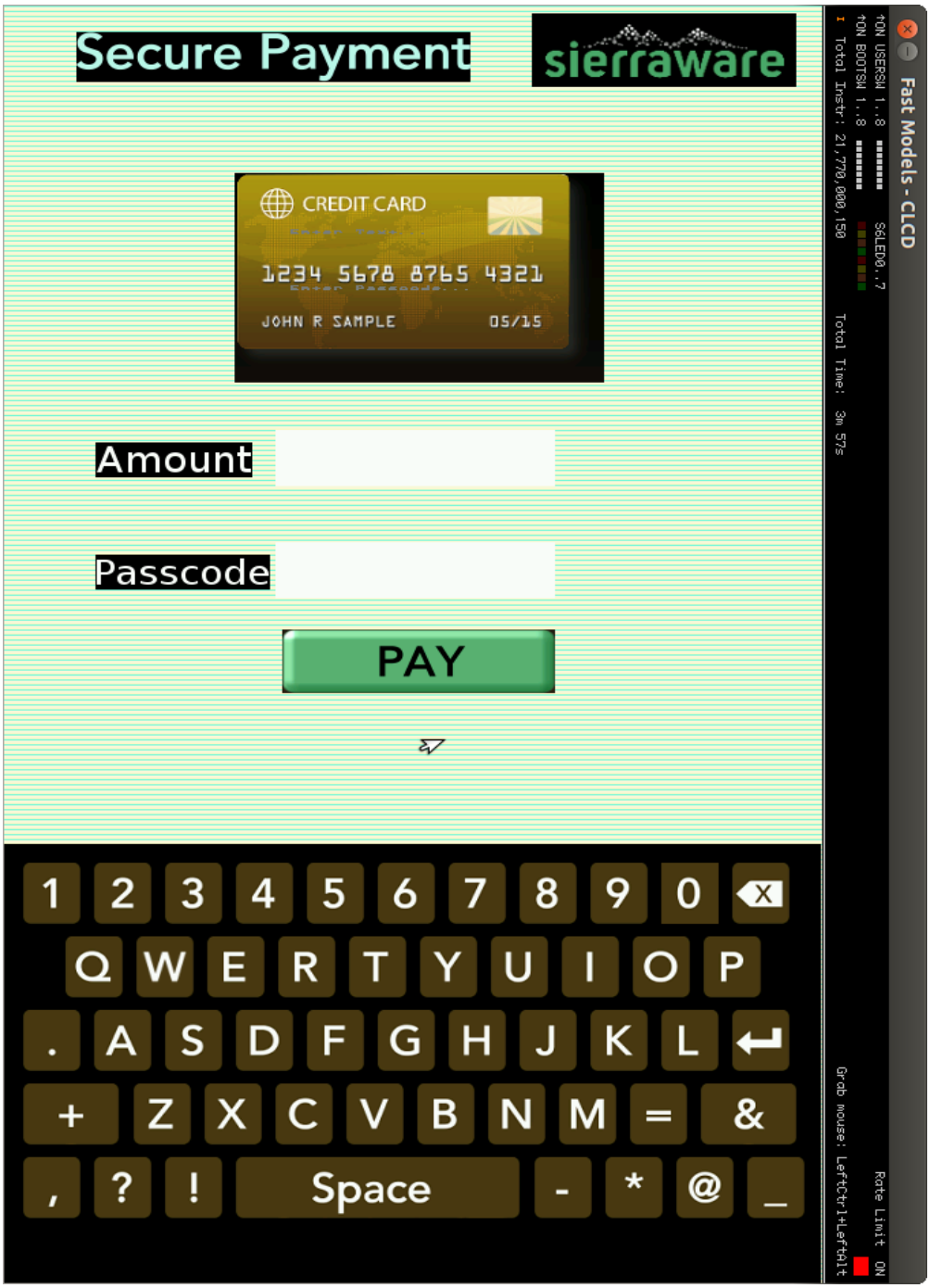
```
RTSM terminal_0
S: Xorg
R: Xorg
S: dhcdbd
Found no /etc/resolv.conf you need one for e.g. browser to resolve URLs
S: ohmd
sbrshd: Can't get address info
No network interface 'eth0' found
No network interface 'usb0' found
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 f
AEL login: root
login[330]: root login on 'ttyAMA0'


BusyBox v1.14.3 (2009-11-12 11:03:55 GMT) built-in shell (ash)
Enter 'help' for a list of built-in commands.

# insmod /otz_client.ko
# otzapp.elf
fd :3
input data test data abcdefgh
out data ipc echo test: Test IPI echo cmd
Attempting to lock the variable
Lock successful. Trying to lock it one more time
Mutex already locked. We cannot lock it anymore !!
Unlock successful. Trying to lock it one more time
Going to sleep
Second time locking successful
=  test data abcdefgh and out data len 0x13
shared res buf addr 0x40167200, out data addr 0x40167200 and value  test_shared_
buffer and out data len 0x13
next decoder data type 0x2
res buf addr 0x4033f000, out data addr 0x4033f000 and value  test_array_space_bu
ffer and out data len 0x18
next decoder data type 0x0
device close successful
Creating task for testing otz mutexes
device close successful
Mutex testing finished
Creating task for testing secure kernel notification feature
echo task handler 0x115
device close successful
Notification testing finished
# otz_tee_app.elf
session id 0x116
TEEC output buffer 0x40174000: test global platform client api: full memory reference  testing
TEEC output buffer test global platform client api: zero copy testing - inout
TEEC output buffer test global platform client api: zero copy testing
TEEC output buffer test global platform client api: non-zero copy
TEEC output buffer test global platform client api: testing temp memory reference
#
```

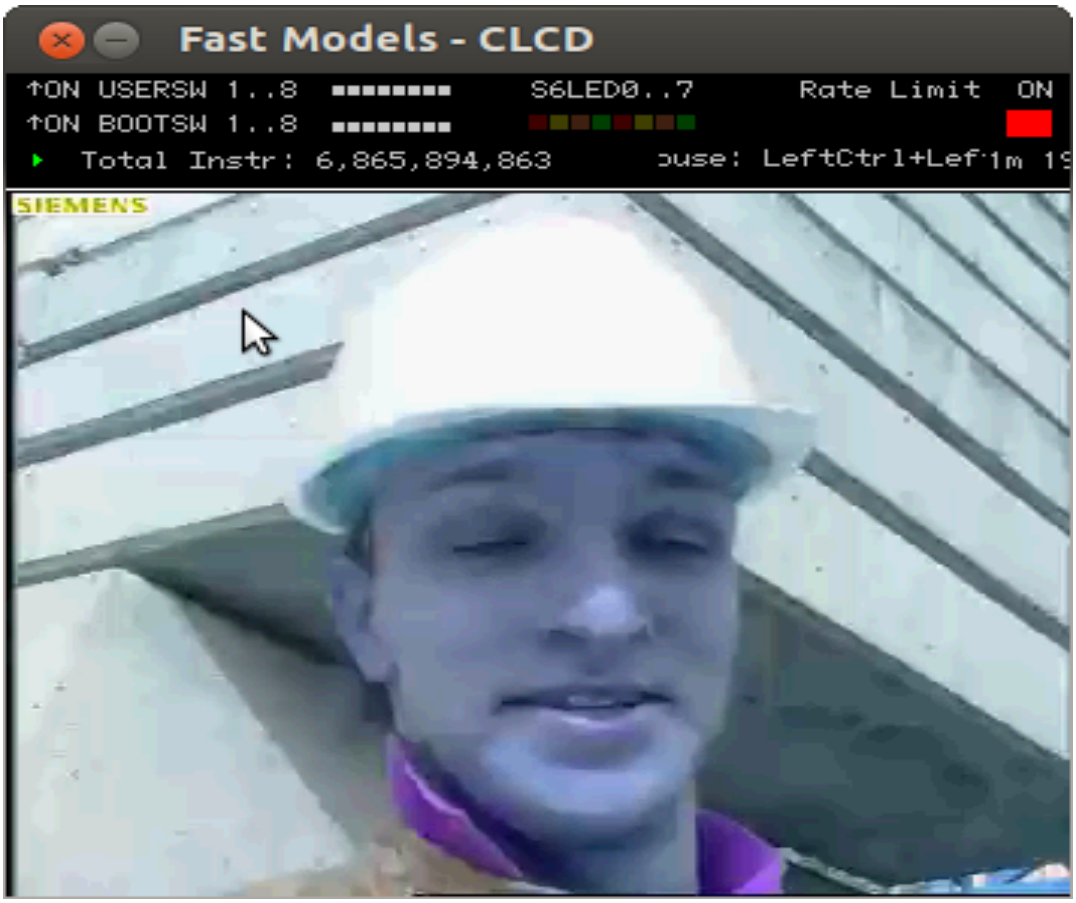
Virtual keyboard can be tested by launching the following application:

otz_virtual_keyboard.elf



Media Player can be tested by launching the following application:

otz_play_media.elf



Neon/VFP support feature can be tested by launching the following application:

```
# otz_neon_app.elf
```

1.2 Steps to Build Open Virtualization SDK:

1.2.1 Steps to build the linux kernel:

i. Download linux-2.6.38.7 from linux kernel source.
ii. Copy the downloaded file to trustzone/otz_linux as linux-2.6.38.7.tar.bz2

iii. Create a soft link or copy the image <FastModelInstalldir>/FastModelsPortfolio_7.1/images/RTSM_EB_Linux/filesystems/armv5t_min_EB_V6_V7.image to trustzone/otz_linux.

iv. Run make command from otz_linux path.

1.2.2 Steps to build the secure kernel :

i. Export the CROSS_COMPILE variable to point to the codesourcery toolchain.

   #export CROSS_COMPILE=<toolchain installed directory>/bin/arm-none-linux-gnueabi-

ii. Export the CROSS_COMPILE_NEWLIB variable to point to the supplied toolchain in case of building crypto application. In case of building *crypto application*,

   #export CROSS_COMPILE_NEWLIB=<supplied toolchain installed directory>/bin/arm-none-eabi-

   Otherwise,

   #export CROSS_COMPILE_NEWLIB=<toolchain installed directory>/bin/arm-none-linux-gnueabi

iii. Run make command from SDK path.

iv. Binaries and library files are available in ***tzone_sdk/bin*** and ***tzone_sdk/lib*** respectively.

v. Linux trustzone api client application (**otzapp.elf**), Linux trustzone client driver (**otz_client.ko**) and trustzone api shared library (**libtzapi.so**) are copied to the root file system.

1.2.3 Steps to follow to resolve the toolchain error in Ubuntu 12.x

While building the secure kernel with crypto enabled in ubuntu 12.04, the tool chain might give some errors due to the missing of the required libraries. Run the following in order to fix this issue.

- apt-get install libmpc2:i386

- apt-get install libmpfr4:i386
- ln -s /usr/lib/i386-linux-gnu/libmpfr.so.4 /usr/lib/i386-linux-gnu/libmpfr.so.1

- apt-get install libgmp10:i386
- ln -s /usr/lib/i386-linux-gnu/libgmp.so.10 /usr/lib/i386-linux-gnu/libgmp.so.3

- apt-get install libelf1:i386
- ln -s /usr/lib/i386-linux-gnu/libelf.so.1 /usr/lib/i386-linux-gnu/libelf.so.0

Note: Supported toolchain version is arm-2010q1.

2 Build and booting Open Virtualization in Fastmodel (Dedicated TEE)

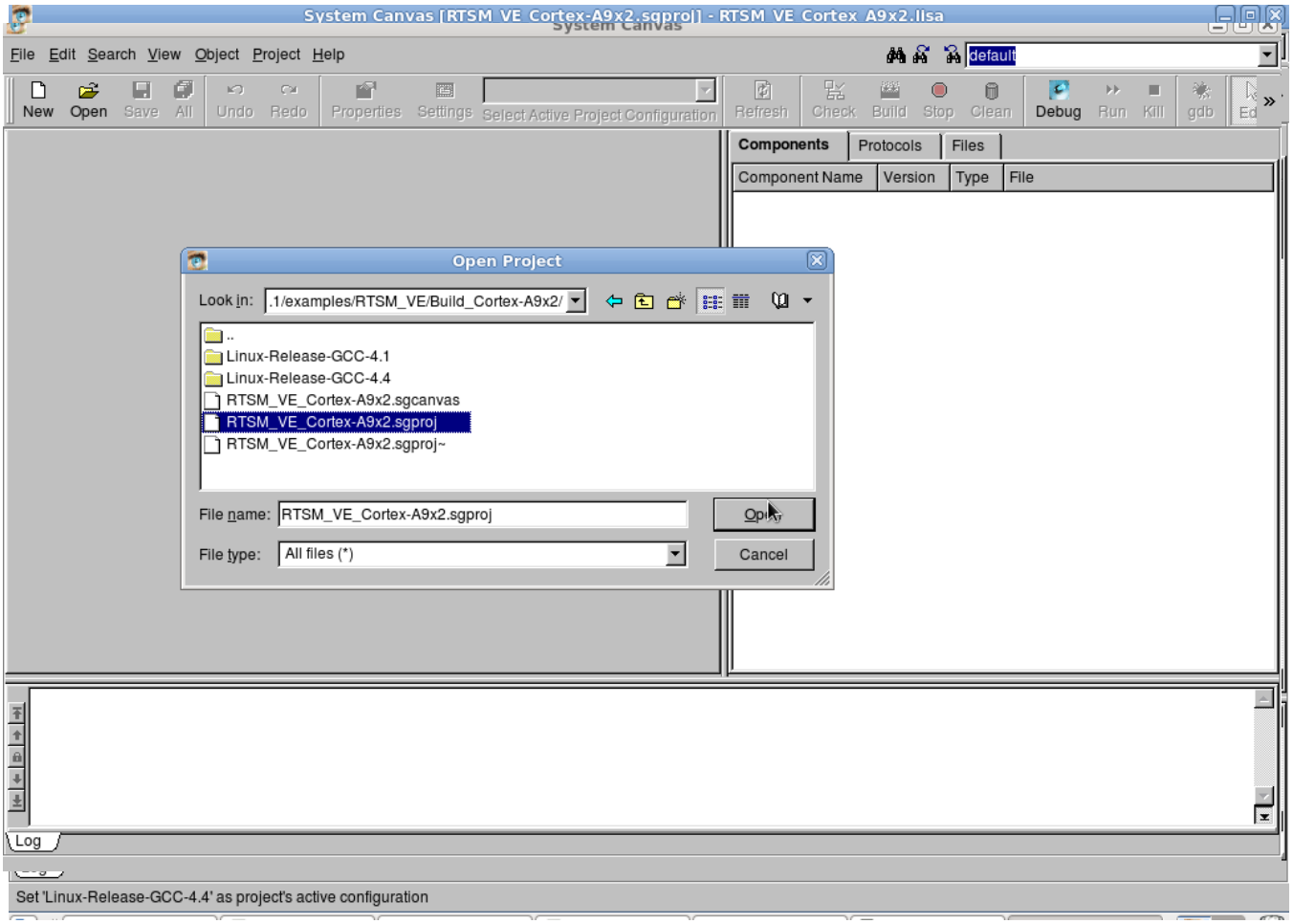
2.1 Steps to boot Open Virtualization in Fastmodel(Dedicated TEE):

The Steps are the same as those described in steps (i) to (v) of Section 1.1

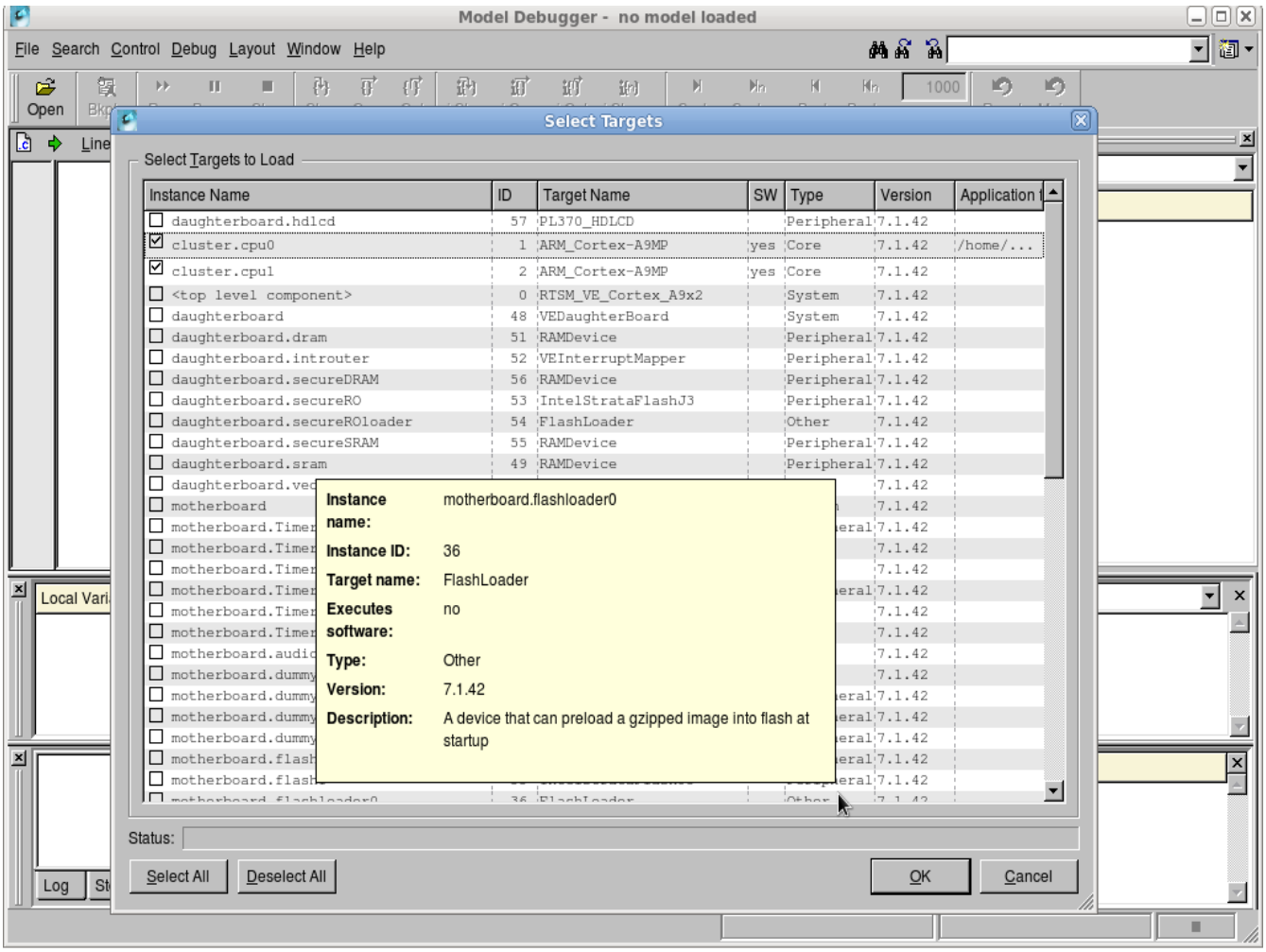
It is supported in Versatile Express Cortex A9 and Cortex A15.

<Fastmodel-install-dir>/FastModelsPortfolio_7.1/examples/RTSM_VE/Build_Cortex-A9x2/RTSM_VE_Cortex-A9x2.sgproj

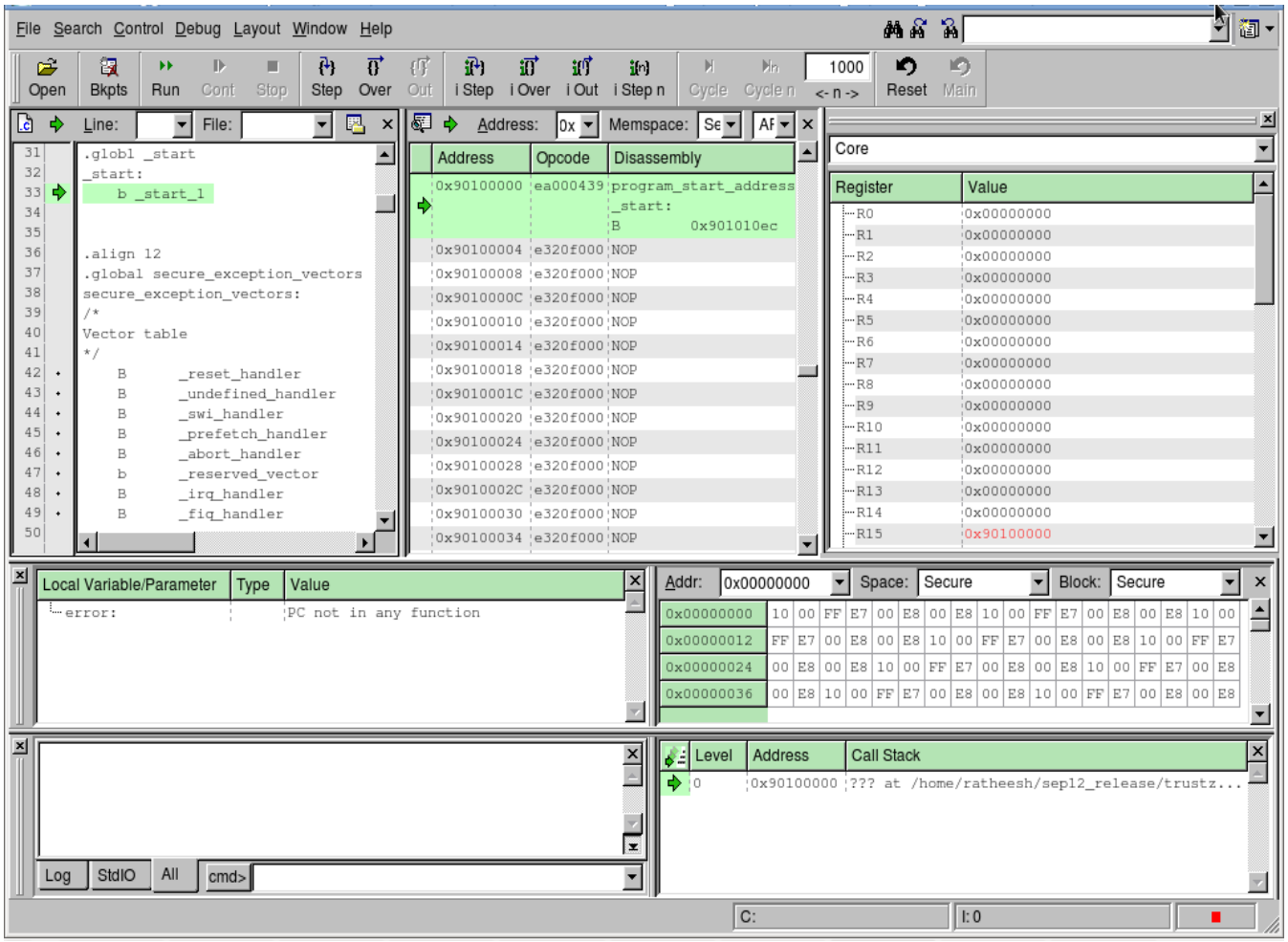
After loading the required project models, a window appears as below.

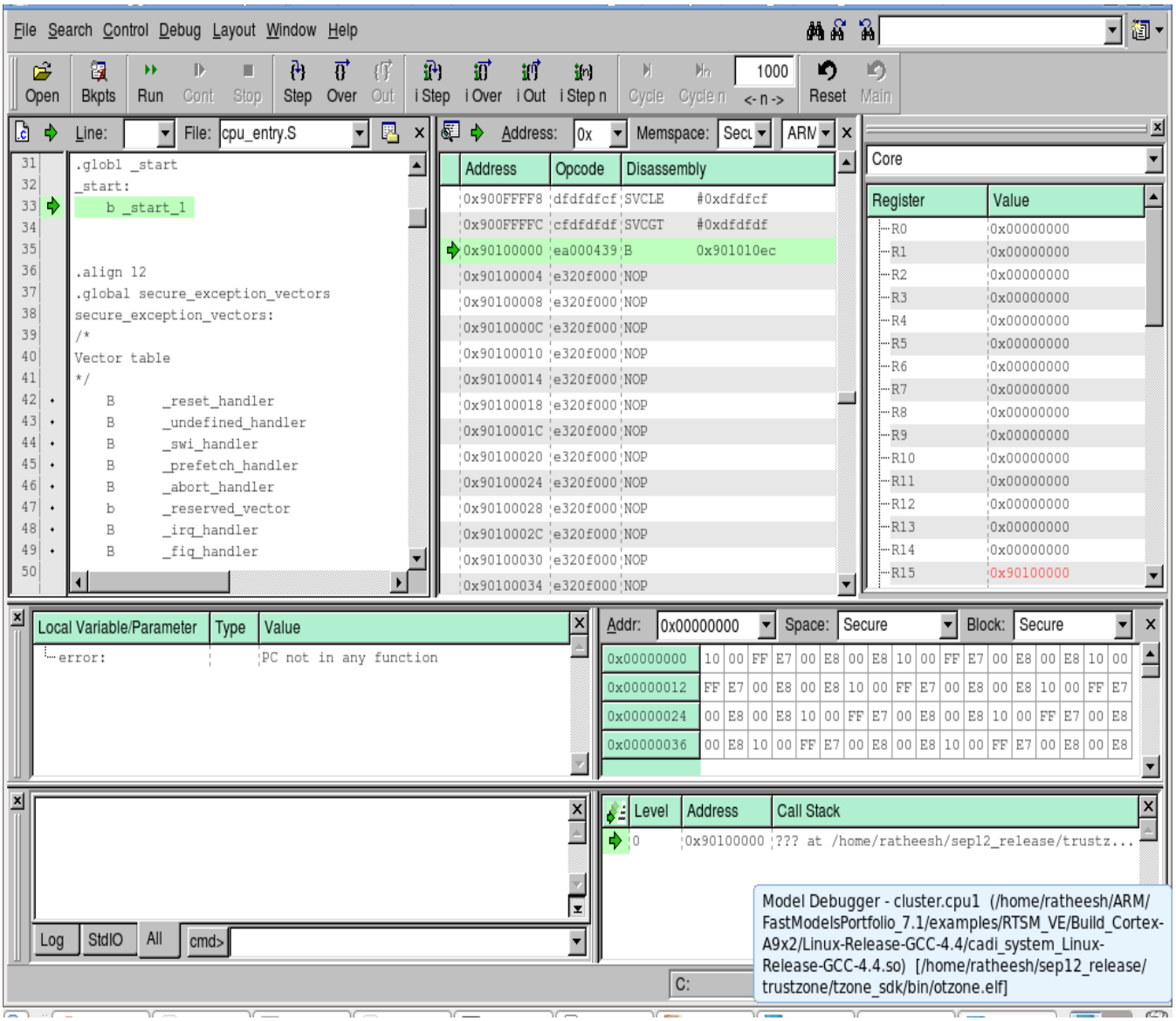

System Canvas [RTSM_VE_Cortex-A9x2.sgproj] - RTSM_VE_Cortex_A9x2.lisa
File Edit Search View Object Project Help
default
New Open Save All Undo Redo Properties Settings Select Active Project Configuration Refresh Check Build Stop Clean Debug Run Kill gdb
Components Protocols Files
Component Name Version Type File
Open Project
Look in: ./examples/RTSM_VE/Build_Cortex-A9x2/
..
Linux-Release-GCC-4.1
Linux-Release-GCC-4.4
RTSM_VE_Cortex-A9x2.sgcanvas
RTSM_VE_Cortex-A9x2.sgproj
RTSM_VE_Cortex-A9x2.sgproj~
File name: RTSM_VE_Cortex-A9x2.sgproj
File type: All files (*)
Open
Cancel
Log
Set 'Linux-Release-GCC-4.4' as project's active configuration

Configure the required parameters in the Configure Model Parameters window and click OK which makes the "select targets" window to appear. Click OK. The file gets loaded and the following window appears.



Two windows will be appearing as shown below.




File Search Control Debug Layout Window Help
Open Bkpts Run Cont Stop Step Over Out i Step i Over i Out i Step n Cycle Cycle n 1000 <- n -> Reset Main
Line: File: cpu_entry.S
31 .globl _start
32 _start:
33 b _start_1
34
35
36 .align 12
37 .global secure_exception_vectors
38 secure_exception_vectors:
39 /*
40 Vector table
41 */
42 B _reset_handler
43 B _undefined_handler
44 B _swi_handler
45 B _prefetch_handler
46 B _abort_handler
47 b _reserved_vector
48 B _irq_handler
49 B _fiq_handler
50
Address: 0x Memspace: Secu ARM
Address Opcode Disassembly
0x900FFFF8 dfdfdfcf SVCLE #0xdfdfcf
0x900FFFFC cfdfdfdf SVCGT #0xdfdfdf
0x90100000 ea000439 B 0x901010ec
0x90100004 e320f000 NOP
0x90100008 e320f000 NOP
0x9010000C e320f000 NOP
0x90100010 e320f000 NOP
0x90100014 e320f000 NOP
0x90100018 e320f000 NOP
0x9010001C e320f000 NOP
0x90100020 e320f000 NOP
0x90100024 e320f000 NOP
0x90100028 e320f000 NOP
0x9010002C e320f000 NOP
0x90100030 e320f000 NOP
0x90100034 e320f000 NOP
Core
Register Value
R0 0x00000000
R1 0x00000000
R2 0x00000000
R3 0x00000000
R4 0x00000000
R5 0x00000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 0x00000000
R14 0x00000000
R15 0x90100000
Local Variable/Parameter Type Value
error: PC not in any function
Addr: 0x00000000 Space: Secure Block: Secure
0x00000000 10 00 FF E7 00 E8 00 E8 10 00 FF E7 00 E8 00 E8 10 00
0x00000012 FF E7 00 E8 00 E8 10 00 FF E7 00 E8 00 E8 10 00 FF E7
0x00000024 00 E8 00 E8 10 00 FF E7 00 E8 00 E8 10 00 FF E7 00 E8
0x00000036 00 E8 10 00 FF E7 00 E8 00 E8 10 00 FF E7 00 E8 00 E8
Level Address Call Stack
0 0x90100000 ??? at /home/ratheesh/sep12_release/trustz...
Log StdIO All cmd>
C:
Model Debugger - cluster.cpu1 (/home/ratheesh/ARM/FastModelsPortfolio_7.1/examples/RTSM_VE/Build_Cortex-A9x2/Linux-Release-GCC-4.4/cadi_system_Linux-Release-GCC-4.4.so) [/home/ratheesh/sep12_release/trustzone/tzone_sdk/bin/otzone.elf]

Clicking Run will boot the system.

```
RTSM terminal_0
usbcore: registered new interface driver usb-storage
USB Mass Storage support registered.
mousedev: PS/2 mouse device common for all mice
rtc-pl031 dev:rtc: rtc core: registered pl031 as rtc0
mmci-pl18x fpga:mmc0: mmc0: PL180 rev0 at 0x10005000 irq 49,50
usbcore: registered new interface driver usbhid
usbhid: USB HID core driver
mmc0: new MMC card at address 0001
aaci-pl041 fpga:aaci: ARM AC'97 Interface at 0x0000000010004000, irq 51, fifo 512
mmcblk0: mmc0:0001 ARMmm256 MiB
ALSA device list:
  #0: ARM AC'97 Interface at 0x0000000010004000, irq 51
TCP cubic registered
NET: Registered protocol family 17
 mmcblk0: unknown partition table
VFP support v0.3: implementor 41 architecture 3 part 30 variant c rev 2
ThumbEE CPU extension supported.
rtc-pl031 dev:rtc: setting system clock to 1970-01-01 00:00:00 UTC (0)
input: AT Raw Set 2 keyboard as /devices/fpga:kmi0/serio0/input/input0
VFS: Mounted root (ext2 filesystem) readonly on device 179:0.
Freeing init memory: 144K
init started: BusyBox v1.14.3 (2009-11-12 11:03:55 GMT)
starting pid 465, tty '': '/etc/rc.d/rc.local'
/etc/rc.d/rc.local: line 14: can't create /var/testfile: Read-only file system
input: PS/2 Generic Mouse as /devices/fpga:kmi1/serio1/input/input1
warning: can't open /etc/mtab: No such file or directory
S: devpts
S: udev
udevd (508): /proc/508/oom_adj is deprecated, please use /proc/508/oom_score_adj instead.
S: sshd
S: dbus id
Thu Jan  1 00:00:03 UTC 1970
S: hald
S: Xorg
R: Xorg
S: dhcdbd
Found no /etc/resolv.conf you need one for e.g. browser to resolve URLs
S: ohwd
No network interface 'eth0' found
No network interface 'usb0' found
sbrshd: Can't get address info
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

/proc/asound/cards file present
 0 [Interface     ]: aaci-pl041 - ARM AC'97 Interface

AEL login:
```

For rebooting with the same image and root file system just click

stop ---> Reset ---> Run

Login as root and follow the successive steps in Section 1.1.

2.2 Steps to build Open Virtualization SDK(Dedicated TEE):

2.2.1 Steps to build the linux kernel for dedicated tee:

1. Enable OTZONE_AMP_SUPPORT by
   making the following change in ***otz-linux/Makefile***.

   export OTZONE_AMP_SUPPORT := y

2. Run Make which will build the linux kernel image.

   #make

2.2.2 Steps to build the secure kernel :

Refer steps 1 to 3 in Section 1.2.2 and enable CONFIG_SW_DEDICATED_TEE in the VE config file present in the following path ***tzone_sdk/otzone/config/config.ve*** before running make from SDK path.

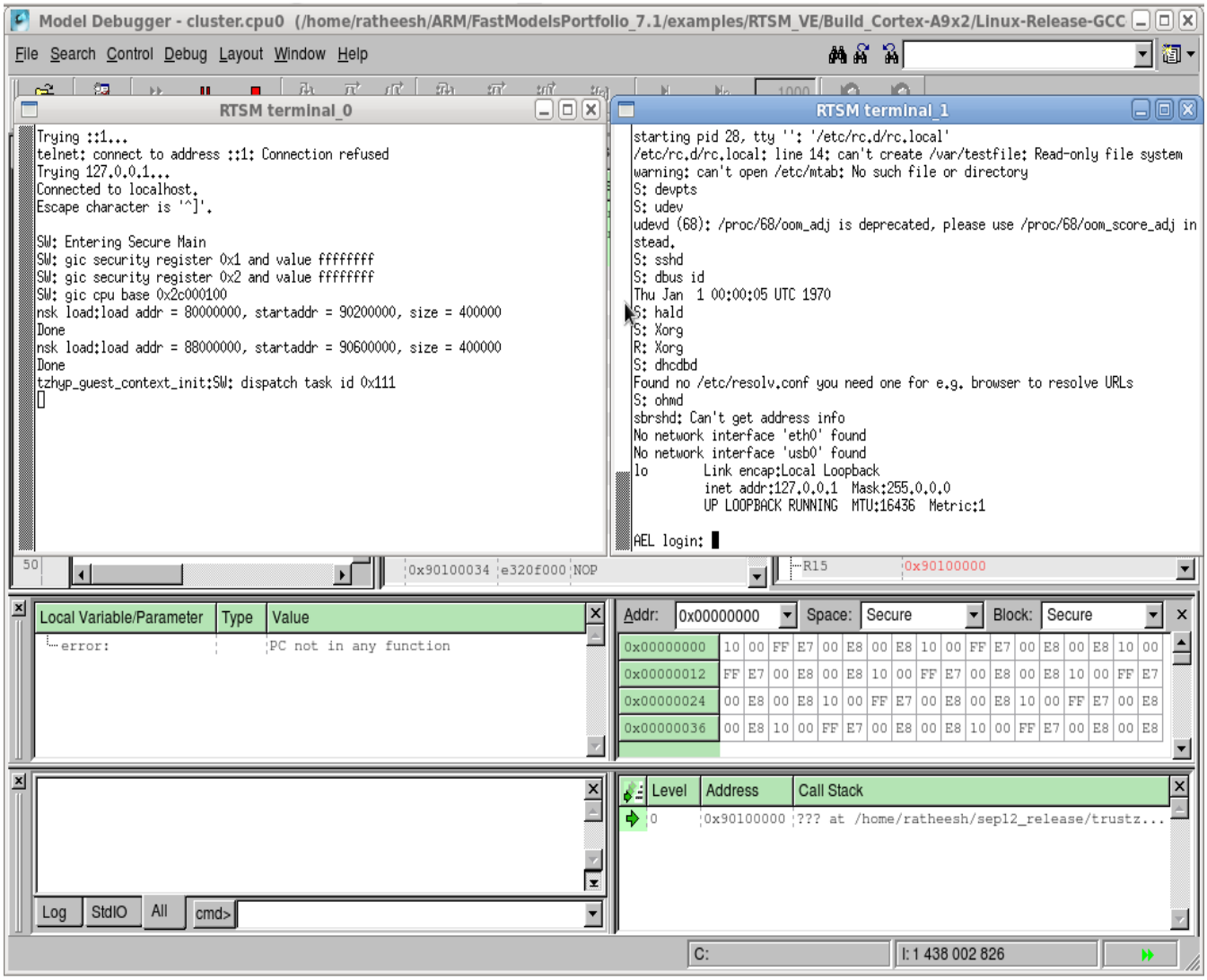
-DCONFIG_SW_DEDICATED_TEE=1 should be enabled in both TARGET_ASMFLAGS and TARGET_CCFLAGS.

3 Build and booting Open Virtualization in Fastmodel (Hypervisor)

3.1 Steps to boot Open Virtualization in Fastmodel(Multiple guests)

Follow the same steps as in Section 1.1.The only exception is that the way in which the testing for the Open Virtualization Stack is done.

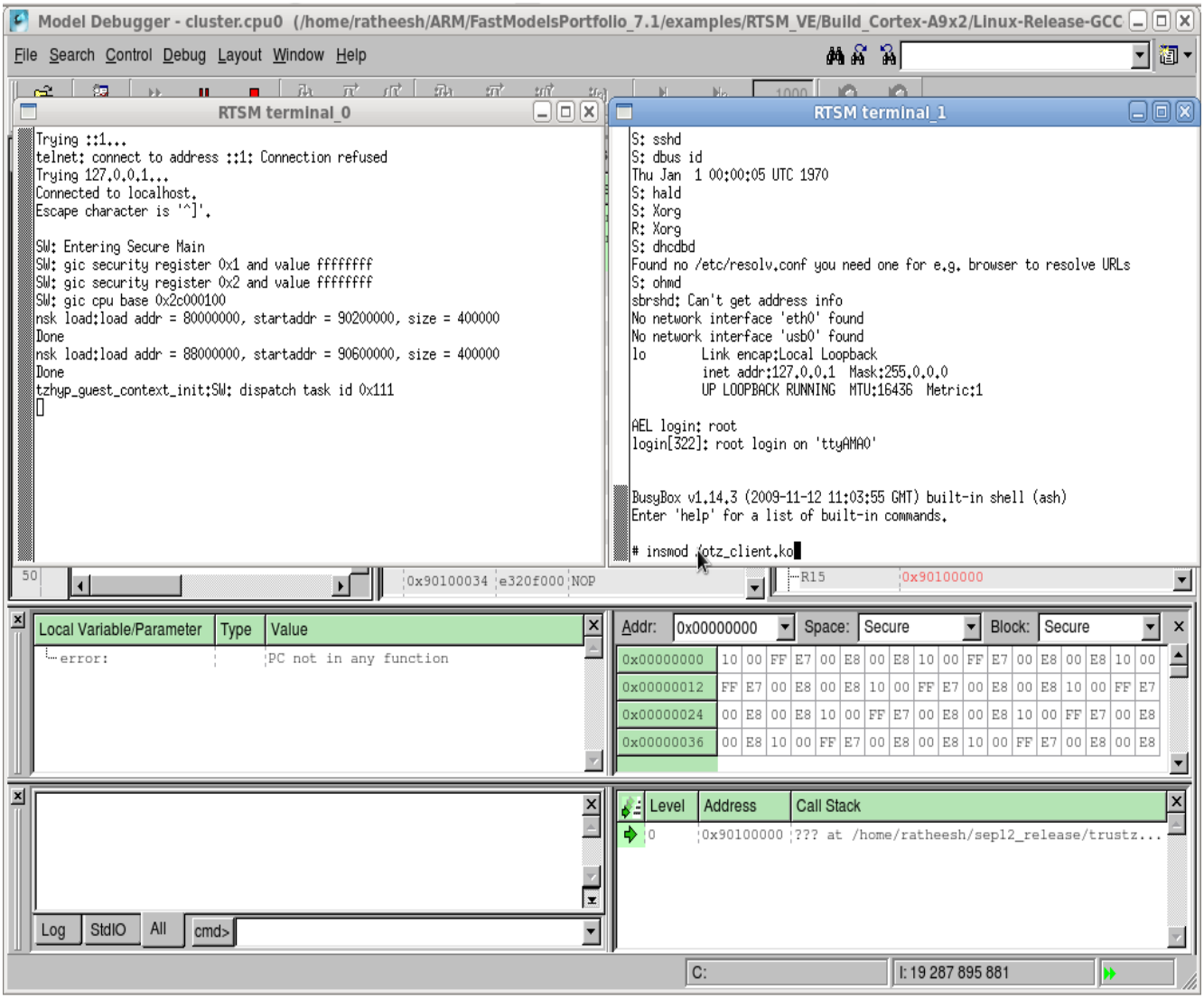
The first linux comes up on booting the system.



Login as root.

Load the Open-Trust-Zone kernel driver module for the first linux by executing the following instruction.

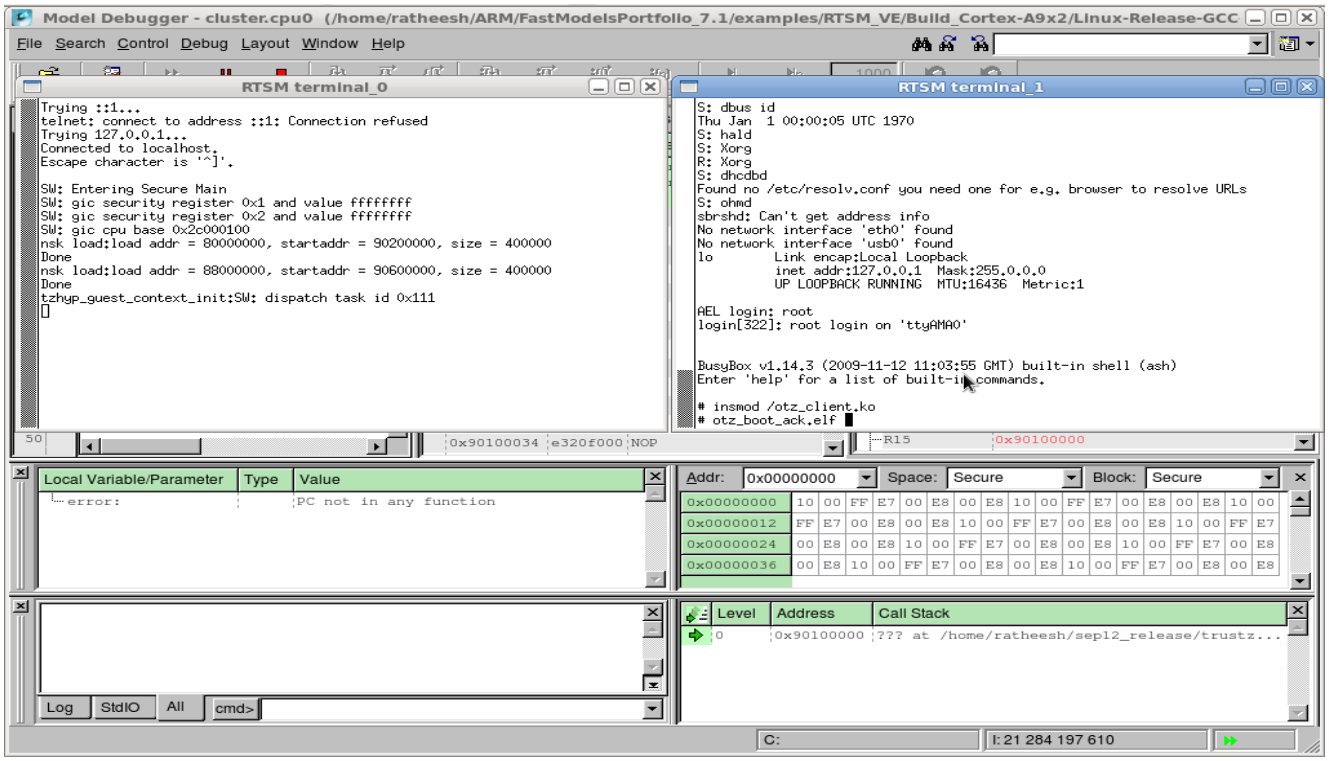
#insmod /otz_client.ko



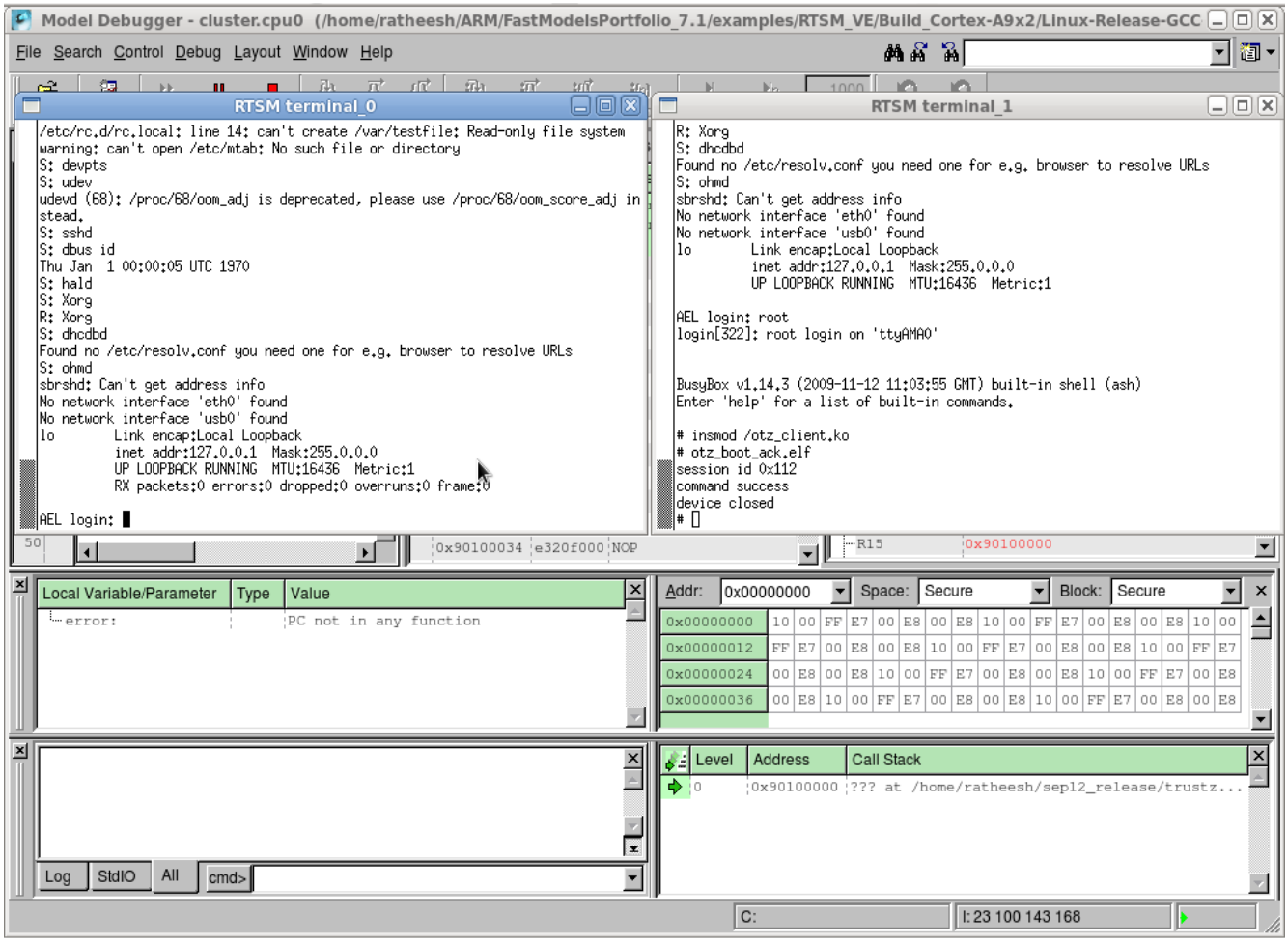
Only the first linux runs and the behaviour would be similar to that of a single guest system.

To start the second linux, run the following application.

#otz_boot_ack_elf


Model Debugger - cluster.cpu0 (/home/ratheesh/ARM/FastModelsPortfolio_7.1/examples/RTSM_VE/Build_Cortex-A9x2/Linux-Release-GCC
File Search Control Debug Layout Window Help
RTSM terminal_0
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
SW: Entering Secure Main
SW: gic security register 0x1 and value ffffffff
SW: gic security register 0x2 and value ffffffff
SW: gic cpu base 0x2c000100
nsk load:load addr = 80000000, startaddr = 90200000, size = 400000
Done
nsk load:load addr = 88000000, startaddr = 90600000, size = 400000
Done
tzhyp_guest_context_init:SW: dispatch task id 0x111
RTSM terminal_1
S: dbus id
Thu Jan 1 00:00:05 UTC 1970
S: hald
S: Xorg
R: Xorg
S: dhcdbd
Found no /etc/resolv.conf you need one for e.g. browser to resolve URLs
S: ohmd
sbrshd: Can't get address info
No network interface 'eth0' found
No network interface 'usb0' found
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
AEL login: root
login[322]: root login on 'ttyAMA0'
BusyBox v1.14.3 (2009-11-12 11:03:55 GMT) built-in shell (ash)
Enter 'help' for a list of built-in commands.
insmod /otz_client.ko
otz_boot_ack.elf
Local Variable/Parameter
Type
Value
error:
PC not in any function
Addr: 0x00000000
Space: Secure
Block: Secure
Log
StdIO
All
cmd>
Level
Address
Call Stack
0x90100000
??? at /home/ratheesh/sep12_release/trustz...
C:
I: 21 284 197 610

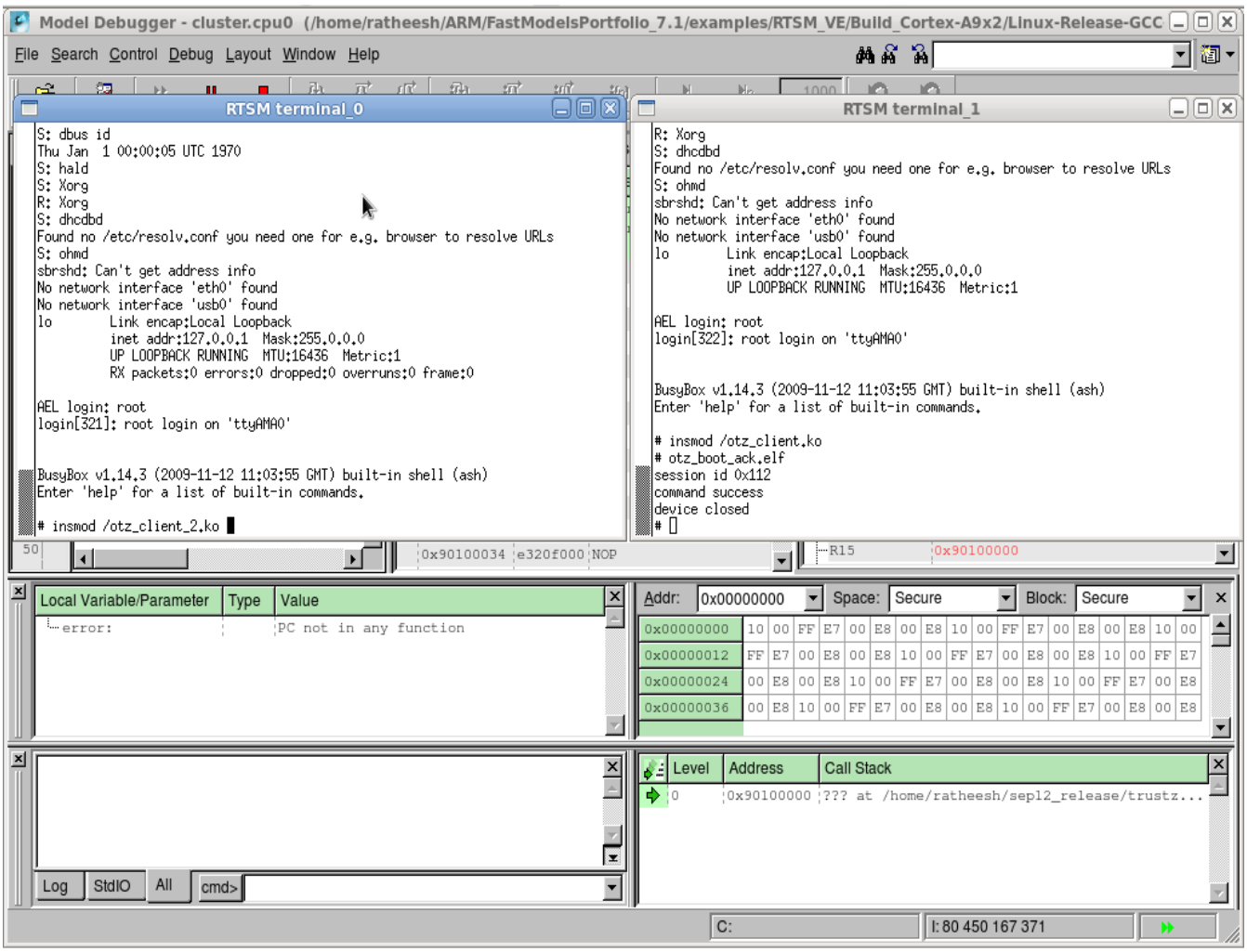
The second linux comes up as shown below.



Login as root.

Load the Open-Trust-Zone kernel driver for the second guest by executing the following instruction

#insmod /otz_client_2.ko

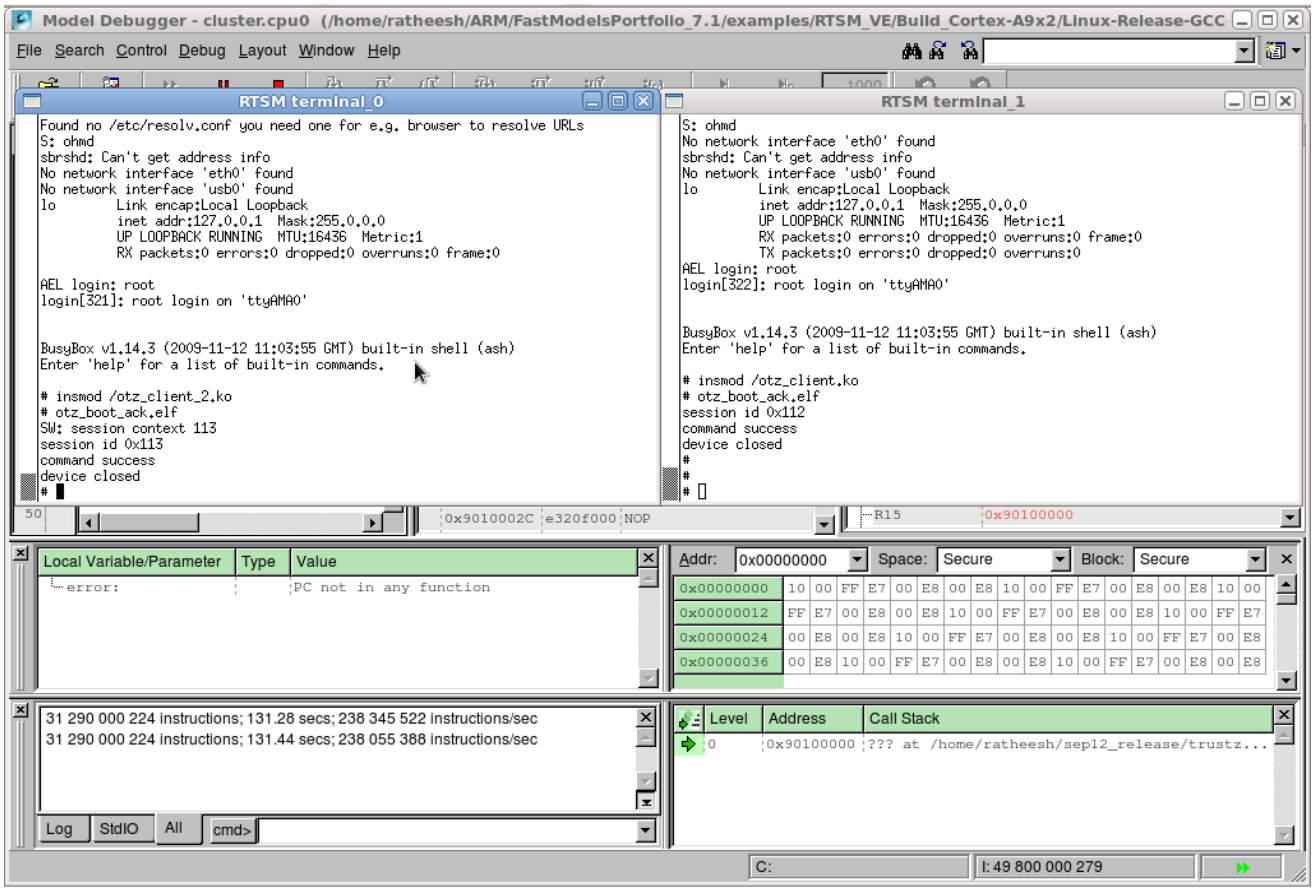


Only the second guest runs at this point and the behaviour would be similar to that of a single guest system.

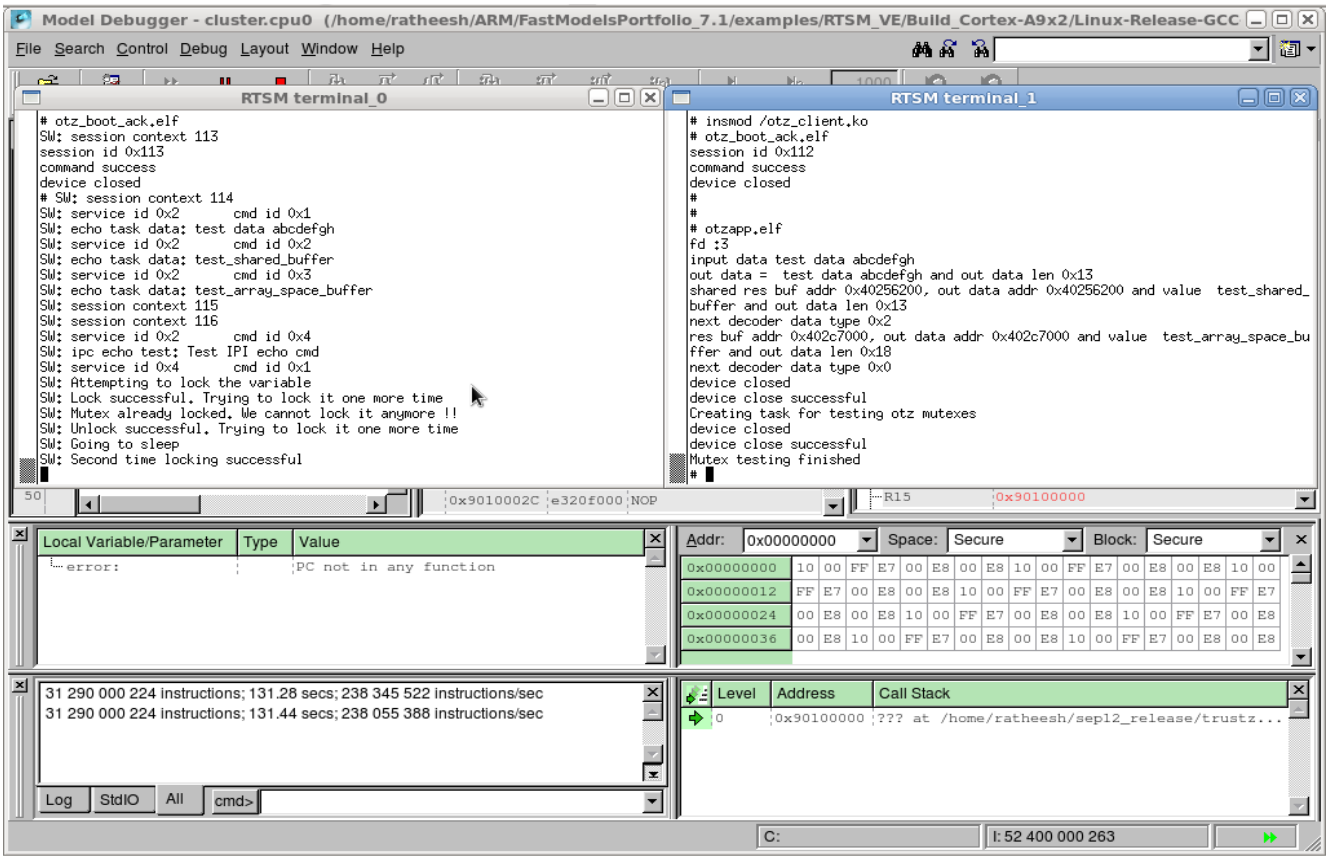
To make both guests run simultaneously, run the following application from second linux.

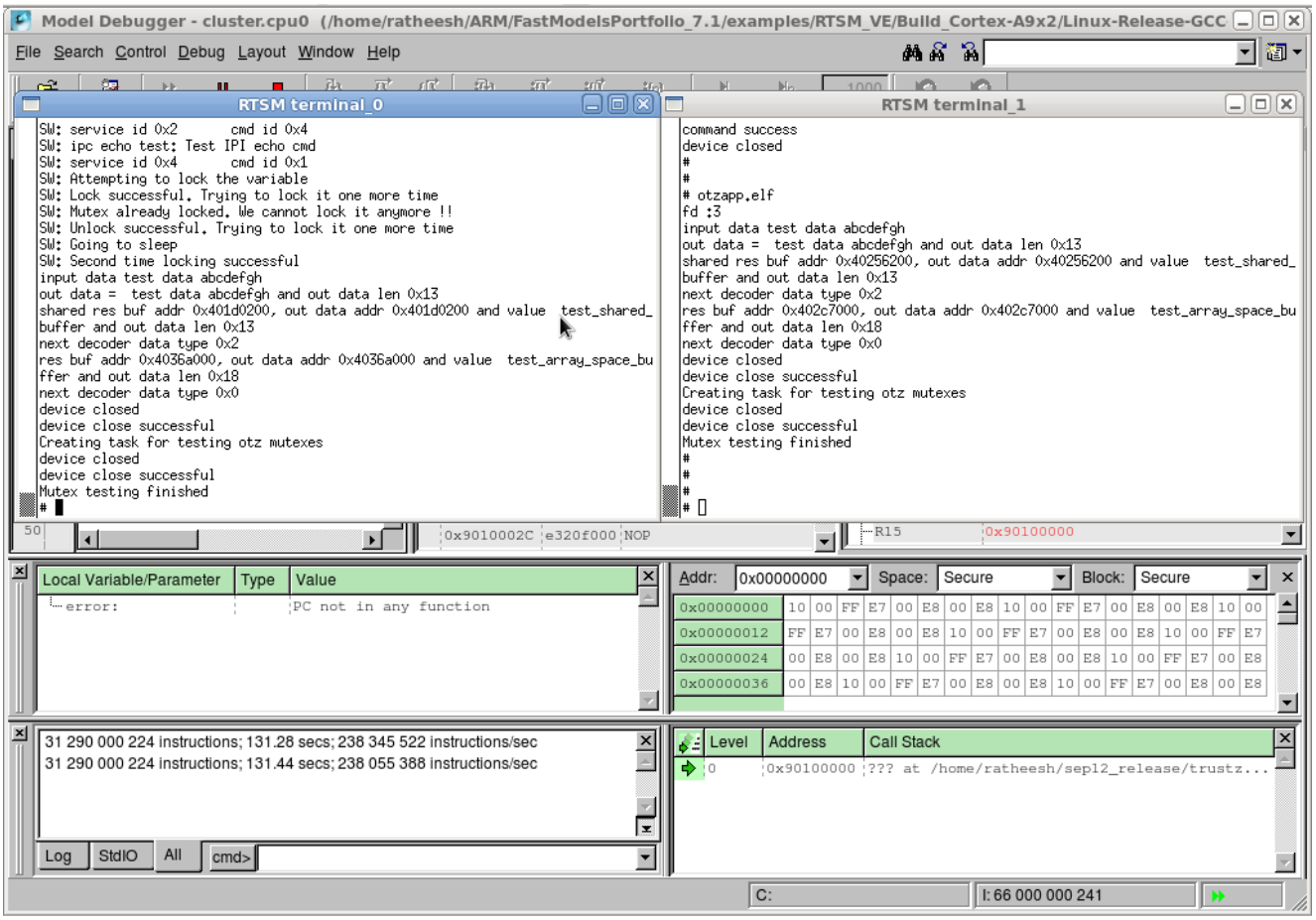
#otz_boot_ack.elf

Running both the guests simultaneously would result in a screen similar to the one below.



Applications can be run from both linux independently.




Model Debugger - cluster.cpu0 (/home/ratheesh/ARM/FastModelsPortfolio_7.1/examples/RTSM_VE/Build_Cortex-A9x2/Linux-Release-GCC
File Search Control Debug Layout Window Help
RTSM terminal_0
SW: service id 0x2 cmd id 0x4
SW: ipc echo test: Test IPI echo cmd
SW: service id 0x4 cmd id 0x1
SW: Attempting to lock the variable
SW: Lock successful. Trying to lock it one more time
SW: Mutex already locked. We cannot lock it anymore !!
SW: Unlock successful. Trying to lock it one more time
SW: Going to sleep
SW: Second time locking successful
input data test data abcdefgh
out data = test data abcdefgh and out data len 0x13
shared res buf addr 0x401d0200, out data addr 0x401d0200 and value test_shared_
buffer and out data len 0x13
next decoder data type 0x2
res buf addr 0x4036a000, out data addr 0x4036a000 and value test_array_space_bu
ffer and out data len 0x18
next decoder data type 0x0
device closed
device close successful
Creating task for testing otz mutexes
device closed
device close successful
Mutex testing finished
RTSM terminal_1
command success
device closed
otzapp.elf
fd :3
input data test data abcdefgh
out data = test data abcdefgh and out data len 0x13
shared res buf addr 0x40256200, out data addr 0x40256200 and value test_shared_
buffer and out data len 0x13
next decoder data type 0x2
res buf addr 0x402c7000, out data addr 0x402c7000 and value test_array_space_bu
ffer and out data len 0x18
next decoder data type 0x0
device closed
device close successful
Creating task for testing otz mutexes
device closed
device close successful
Mutex testing finished
0x9010002C e320f000 NOP
R15 0x90100000
Local Variable/Parameter Type Value
error: PC not in any function
Addr: 0x00000000 Space: Secure Block: Secure
31 290 000 224 instructions; 131.28 secs; 238 345 522 instructions/sec
31 290 000 224 instructions; 131.44 secs; 238 055 388 instructions/sec
Level Address Call Stack
0 0x90100000 ??? at /home/ratheesh/sepl2_release/trustz...
Log StdIO All cmd>
C:
I: 66 000 000 241

3.2 Steps to Build Open Virtualization SDK(Multiple guests):

3.2.1 Steps to build the secure kernel :

Follow the same steps as in Section 1.2.2 and in addition perform the following two steps before running make from the SDK path.

1. Enable the CONFIG_MULTI_GUESTS_SUPPORT variable in ***tzone_sdk/Makefile.***

2. Enable CONFIG_MULTI_GUESTS_SUPPORT in the ve configuration file which is present in the following path by uncommenting the corresponding lines.

   tzone_sdk/otzone/config/config.ve

   -DCONFIG_MULTI_GUESTS_SUPPORT=1

3. Run make from ***tzone_sdk*** .This would produce the required binaries and library files in ***tzone_sdk/bin*** and ***tzone_sdk/lib*** respectively.

4. Linux trustzone api client application (**otzapp.elf**), Linux trustzone client driver (**otz_client.ko**) and trustzone api shared library (**libtzapi.so**) are copied to the root file system.

5. Driver modules for linux guest 1 is otz_client.ko and linux guest2 is otz_client_2.ko .

3.2.2 Steps to build multiple linux guests:

1. Enable MULTIPLE_GUESTS_SUPPORT variable in ***otz-linux/Makefile***.

2. Running make will build two linux guests.
   #make

4 Build and booting Android based Open Virtualization in Fastmodel

4.1 Steps to build Android based Open Virtualization

4.1.1 Steps to build Android SDK:

(i) Download the script to build android from this link
http://snapshots.linaro.org/android/~linaro-android/vexpress-rtsm-ics-gcc46-armlt-stable-open-12.04-release/5/linaro_android_build_cmds.sh

(ii) Comment out the last two lines in the script corresponding to the building of android
#. build/envsetup.sh

#make -j${CPUS} boottarball systemtarball userdatatarball

(iii) Run the following script
#android_install.sh

(iv) Uncomment the last two lines in the android build script.
. build/envsetup.sh
make -j${CPUS} boottarball systemtarball userdatatarball

(v) Run the script
#linaro_android_build_cmds.sh

4.1.2 Steps to Build Open Virtualization SDK (Android supported kernel):

4.1.3 Steps to build the secure kernel :

1. Export the CROSS_COMPILE variable to point to the codesourcery toolchain that would be used.

   #export CROSS_COMPILE=<toolchain installed directory>/bin/arm-none-linux-gnueabi-

2. Export the CROSS_COMPILE_NEWLIB variable to point to the supplied toolchain that would be used. This toolchain is used while building crypto application.

   #export CROSS_COMPILE_NEWLIB=<toolchain installed directory>/bin/arm-none-eabi-

3. Enable the CONFIG_ANDROID_GUEST variable in ***tzone_sdk/Makefile***

4. Modify the ANDROID_PATH variable to the corresponding Android install directory.

5. Run make command from SDK path.

   #make

6. Binary files are available in ***tzone_sdk/bin***.

7. Linux trustzone client driver (**otz_client.ko**) is copied to the android root file system.

4.2 Steps to boot Android based kernel:

4.2.1 Create an SD card image:

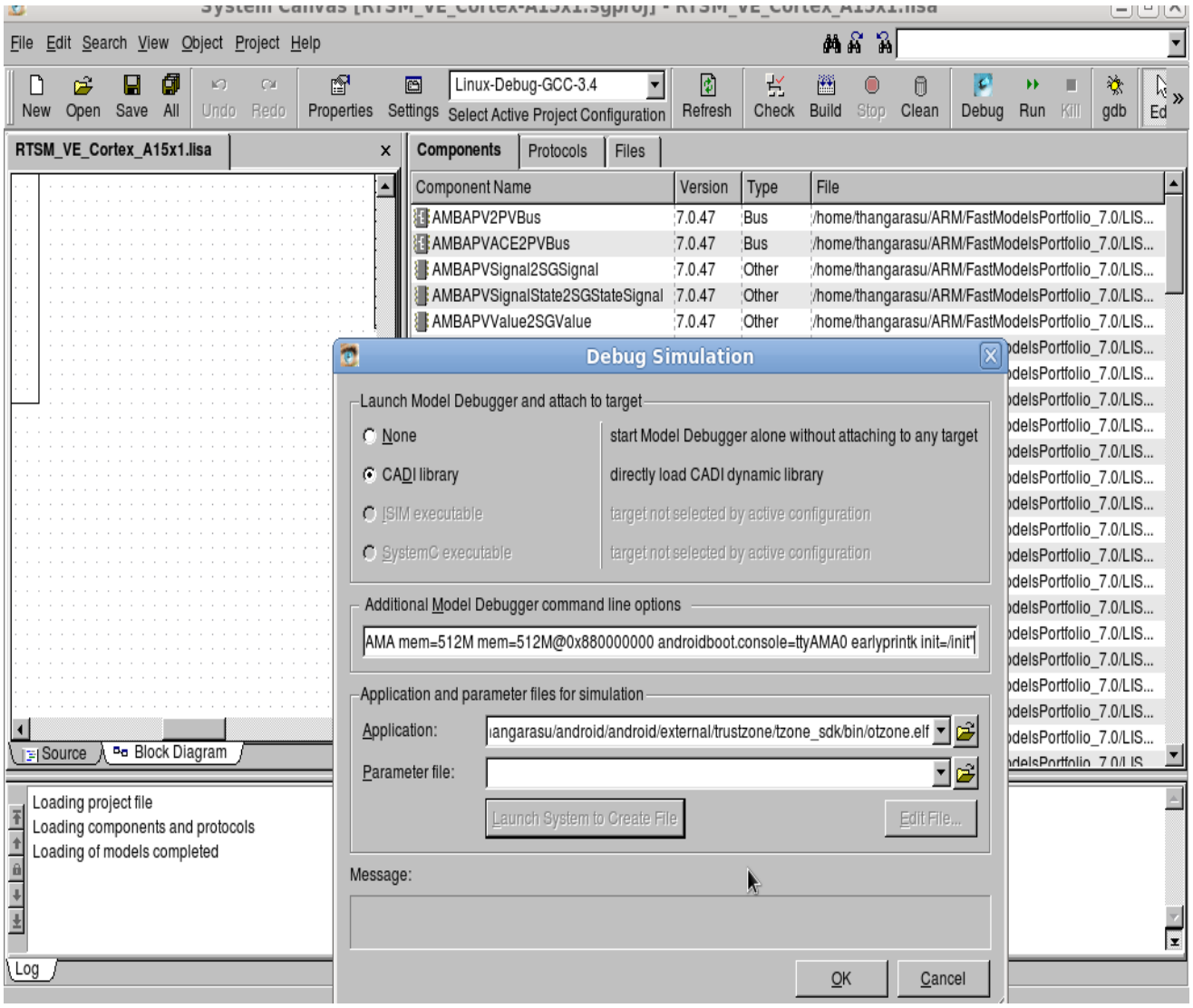
1. Untar the boot.tar.bz2 file in a directory.
   #cd <android_temp_extracted_directory>
   #tar -xvf boot.tar.bz2

2. *Create mmc.bin by running this command.*
   #bzr branch lp:linaro-image-tools
   #./linaro-image-tools/linaro-android-media-create --image_file *mc.bin --image_size 2000M --dev vexpress-a9 --system system.tar.bz2 --userdata userdata.tar.bz2 --boot boot.tar.bz2*

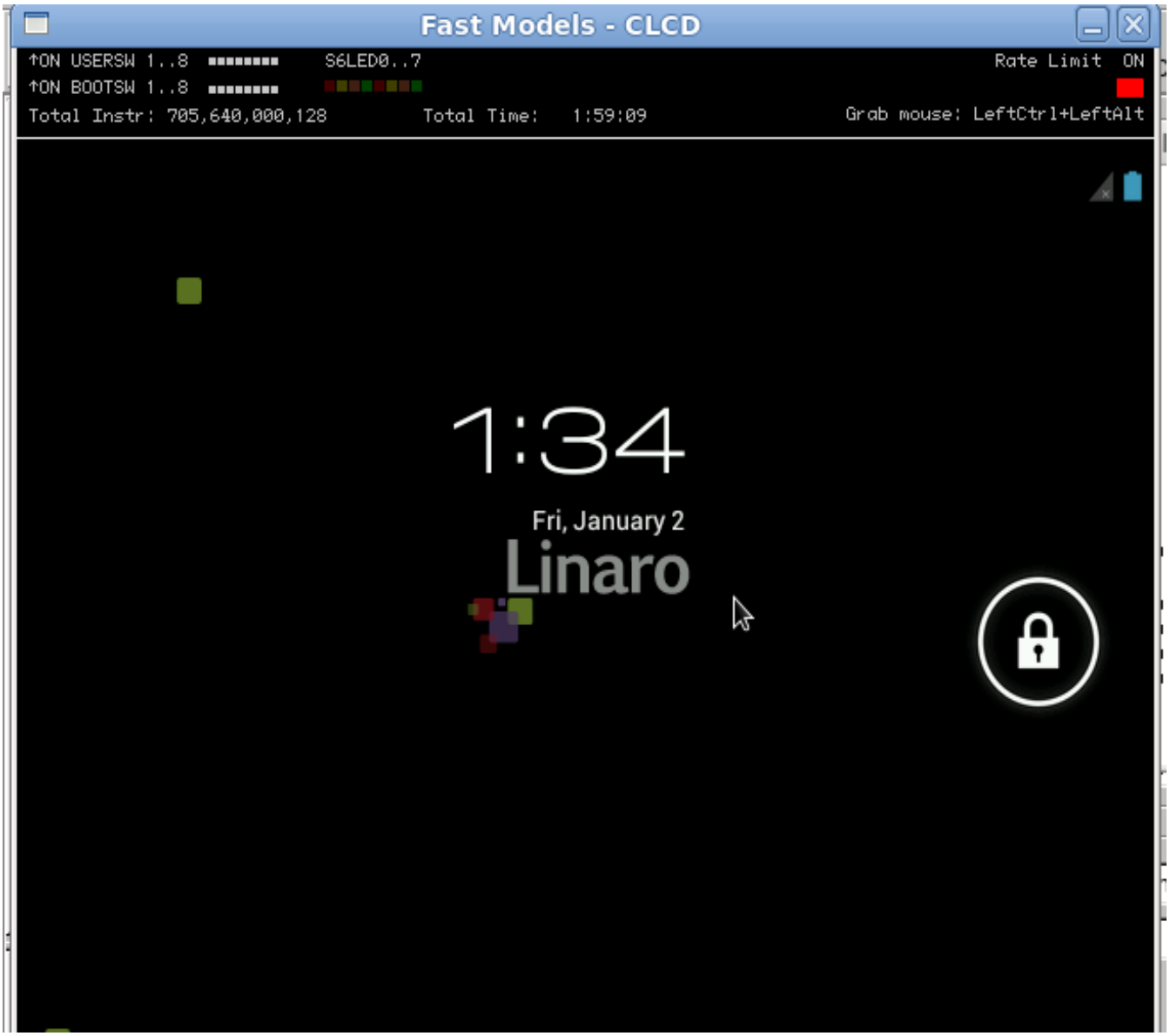
4.2.2 Steps to boot Open Virtualization in Fastmodel (Single Android guest)

1. Refer section 1 for building the Fastmodel.

2. Run on the A15 Simulator with semi-hosting.Give Additional Model Command Line Options at runtime as
   "-C motherboard.mmc.p_mmc_file =*<android_temp_extracted_directory>*/mmc.bin -C cluster.cpu0.semihosting- cmd_line="--kernel *<android_temp_extracted_directory>*/boot/uImagewithDT --initrd *<android_temp_extracted_directory>*/boot/Initrd -- console=ttyAMA mem=512M mem=512M@0x880000000 androidboot.console=ttyAMA0 earlyprintk init=/init"



Click OK and then run the android supported otzone.elf image


Fast Models - CLCD
↑ON USERSW 1..8
S6LED0..7
Rate Limit ON
↑ON BOOTSW 1..8
Total Instr: 705,640,000,128
Total Time: 1:59:09
Grab mouse: LeftCtrl+LeftAlt
1:34
Fri, January 2
Linaro

4.2.3 Steps to boot Open Virtualization in Fastmodel (Android guest and Vanilla kernel)

1. Please Refer section 3.1 to boot the multiple guests except that we need to pass the command line options in the fastmodel debug launch dialog.

2. The Command Line Options are
   "-C motherboard.mmc.p_mmc_file =<*android_temp_extracted_directory*>/mmc.bin -C cluster.cpu0.semihosting- cmd_line="--kernel <*android_temp_extracted_directory*>/boot/uImagewithDT --initrd <*android_temp_extracted_directory*>/boot/Initrd -- console=ttyAMA mem=512M mem=512M@0x880000000 androidboot.console=ttyAMA0 earlyprintk init=/init"

3. Follow the other steps as in the section 3.1

4. Before running otz_boot_ack.elf from android console, wait for the GUI to appear.

5 Steps for installing other packages:

1. Experimental support for openssl has been added. Since this is experimental, a separate executable file **otzone-crypto.elf** would be created.

   To enable openssl support , the following changes need to be made.

   (i) Download openssl-1.0.1c from http://www.openssl.org/source/openssl-1.0.1c.tar.gz and place it in /*trustzone/package/storage folder.*

   (ii) Set ENABLE_LIBCRYPT to "y" in ***tzone_sdk/Makefile.***

   (iii) Set the value of the variable CONFIG_CRYPTO to "y" in the file config.package, found under the directory package.

   (iv) The variable CROSS_COMPILE_NEWLIB should be made to point to the appropriate path.

   #export CROSS_COMPILE_NEWLIB=<supplied toolchain installed directory>/bin/arm-none-eabi-

2. Load the image **otzone-crypto.elf ,** which would be available in the bin directory.

3. The app can be tested by launching otzapp.elf as discussed previously.

#otzapp.elf

6 Components description

Path	Component	Description
/trustzone/otz_linux	ov_android.patch	Patch to support the android with open virtualization
/trustzone/otz_linux/patches	otz_hyp_linux1.patch	Patch for linux kernel guest 1
/trustzone/otz_linux/patches	otz_hyp_linux2.patch	Patch for linux kernel guest 2
/trustzone/otz_linux/patches	otz_linux_async.patch	Patch to support the asynchronous
/trustzone/package/patches	openssl.patch	Patch to support the openssl
/trustzone/tzone_sdk/otzone /src	apps	Sample secure OS applications
/trustzone/tzone_sdk/otzone /src	arch	Architecture dependent codes
/trustzone/tzone_sdk/otzone /src	core	Core modules of the secure OS like memory management, task handler
/trustzone/tzone_sdk/otzone /src	drivers	Supported driver code
/trustzone/tzone_sdk/otzone /src	fs	Supported file system code
/trustzone/tzone_sdk/otzone /src	gui	Graphical User Interface code
/trustzone/tzone_sdk/otzone /src	lib	Library routines for memory, string, etc..

7 Configuration Flags

Flag	Description
CONFIG_FILESYSTEM_SUPPORT	Enables the support for file system
CONFIG_GUI_SUPPORT	Enables GUI support
TOUCH_SUPPORT	Enables the touch screen support in TEE
TIMER_NOT_DEFINED	Disables the secure timer support
CONFIG_MULTI_GUESTS_SUPPORT	Enable trustzone based hypervisor or running multiple guests, number of guests is slected by GUESTS_NO macro
CONFIG_SW_DEDICATED_TEE	Runs secure OS in a dedicated core
CONFIG_SW_BL	Enables the support for secure bootloader
CONFIG_SW_MULTICORE	Enables support for booting SMP kernels in non secure world, Number of cores to be enabled is selected by MAX_CORES macro
CONFIG_SW_NOBOOTLOADER	Enables support for directly running the secure world OS on target without running a bootloader
CONFIG_KSPACE_PROTECTION	Run tasks in a lower privilege mode (user mode) and not in the kernel(system mode)
CONFIG_CACHE_L2x0	Enables the L2 cache controller support in Secure OS.
CONFIG_CACHE_PL310	Enables the PL310 L2 Cache controller support
CONFIG_MMC_SUPPORT	Enables the eMMC support
CONFIG_GP_TEST	Enables to test Global platform internal API specification
CONFIG_SW_ELF_LOADER_SUPPORT	Enables the dynamic loading of modules
CONFIG_NEON_SUPPORT	Enables the Neon co-processor support
OTZONE_ASYNC_NOTIFY_SUPPORT	Enables the asynchronous support in secure world
CONFIG_ANDROID_GUEST	Enables to boot Android guest
NON_SECURE_BOOTWRAPPER_SUPPORT	Enables the boot wrapper support for non-secure kernel (e.g ARM Fastmodel)
CONFIG_CRYPTO, CRYPTO_BUILD	Enables the crypto to build

8 Additional Features:

8.1 Neon Support:

- Enable the CONFIG_NEON_SUPPORT in tzone_sdk/Makefile
  CONFIG_NEON_SUPPORT :=y

- Enable the Target flags in tzone_sdk/otzone/config/config.<board>

Uncomment the line -DCONFIG_NEON_SUPPORT=1 in both ASMFLAGS and CCFLAGS.

8.2 Dynamic Modules Support:

8.2.1 Steps to enable the Dynamic modules support

- Enable the CONFIG_SW_ELF_LOADER_SUPPORT in tzone_sdk/otzone/config/config.<board>
  CONFIG_SW_ELF_LOADER_SUPPORT=y
- Enable the Target flags in tzone_sdk/otzone/config/config.<board>
  Uncomment the line -D CONFIG_SW_ELF_LOADER_SUPPORT=1 in both ASMFLAGS and CCFLAGS.

8.2.2 Steps to add dynamic modules

- Application configuration
  <sa_config instance>.entry_func – Name of the entry function
  <sa_config instance>.elf_flag – Set it to ELF_FLAG
  <sa_config instance>.entry_point – Set it to '0' and ELF loader will handle the relocations automatically and set the entry point address to the address of the function specified in the "entry_func" field.
  <sa_config instance>.process_name – Processing function name of the module.
  <sa_config instance>.process– Set it to '0' and ELF loader will set its value to the appropriate address of the function specified in the "process_name" field.
- If the application invokes the kernel function, the appropriate symbol must be exported by "EXPORT_SYMBOL" macro.
- Dynamic modules support requires the file system support.

8.2.3 Testing dynamic modules

- Dynamic modules support added to mutex test and virtual keyboard application.

8.3 DRM Plugin:

DRM plugin supported for Android ICS based systems. This feature is not currently supported in Fastmodel VE platform.

8.4 MMC:

eMMC driver support is added. This feature is not currently supported in Fastmodel VE platform.