KGDB HOWTO

Introduction

After 2.6.26, kgdb is integrated into kernel, it may pay less effort if we use a newer kernel version. For following examples, we use a stable version 2.6.35.7 from http://www.kernel.org/ to demonstrate kgdb.

We highly recommend you rebuild your kernel without kgdb first. Following examples we assume you have already succeeded in building kernel.
(A simple method is using

- `make menuconfig', it will generate a default generic `.config'
- `make all', it will make linux images(compressed and uncompressed) and modules
- `make modules_install', it will install modules
- `make install', it will move linux images to /boot and edit boot manager configuration in some distributions

Next, you may need `mkinitrd' or `mkinitramfs'(in some distributions) to make cpio-related files. Finally, configure your boot manager if needed. However, rebuilding kernel is not our focus here, please try to debug with Internet if you met any trouble. Good luck! ☺)

It is nice to use virtual machine to learn operating system. For following examples, we use `VirtualBox', a kind of virtual machine, to demonstrate how to use kgdb.
We are so sorry if this software is not familiar with you.

Step1

First, we must know kgdb is very similar to gdb. When we use gdb, we put `-g' option to gcc for debug info. We also put debug info related options in kernel configuration when we use kgdb.

After using `make menuconfig', we will see a user-friendly interface like figure 1. We can use `/kgdb' to find kgdb-related options.

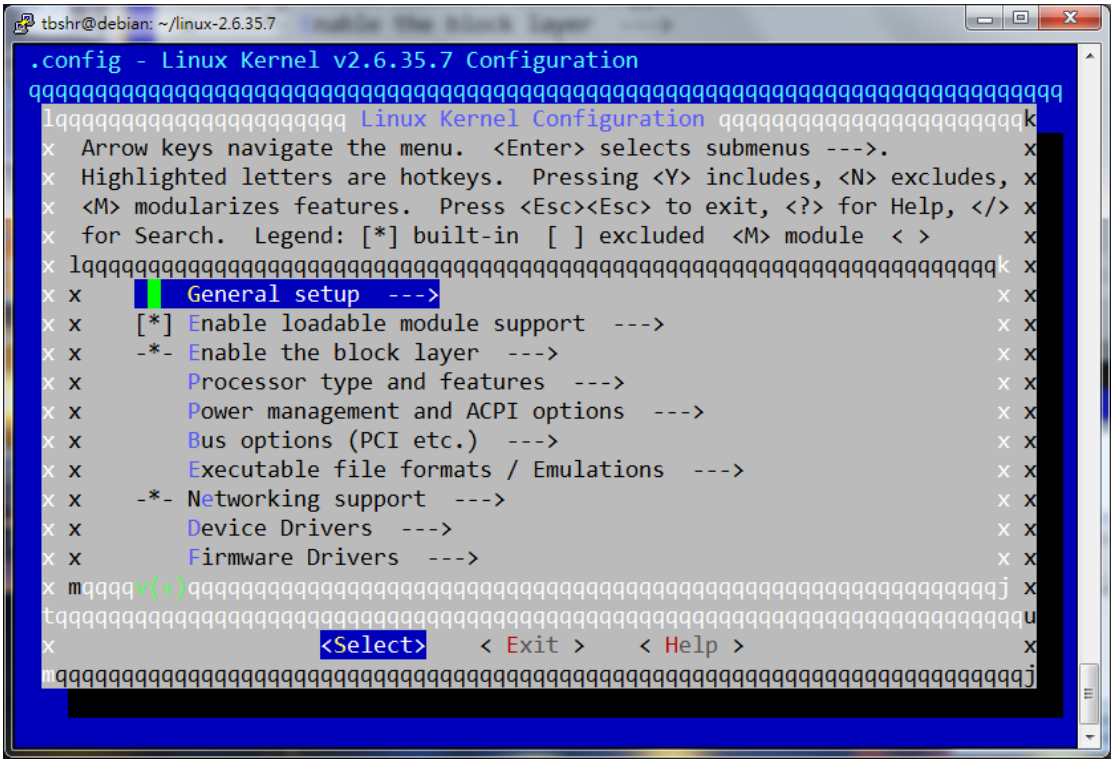

Figure 1

There are some essential options for kgdb. Please enter the `Kernel hacking' entry.

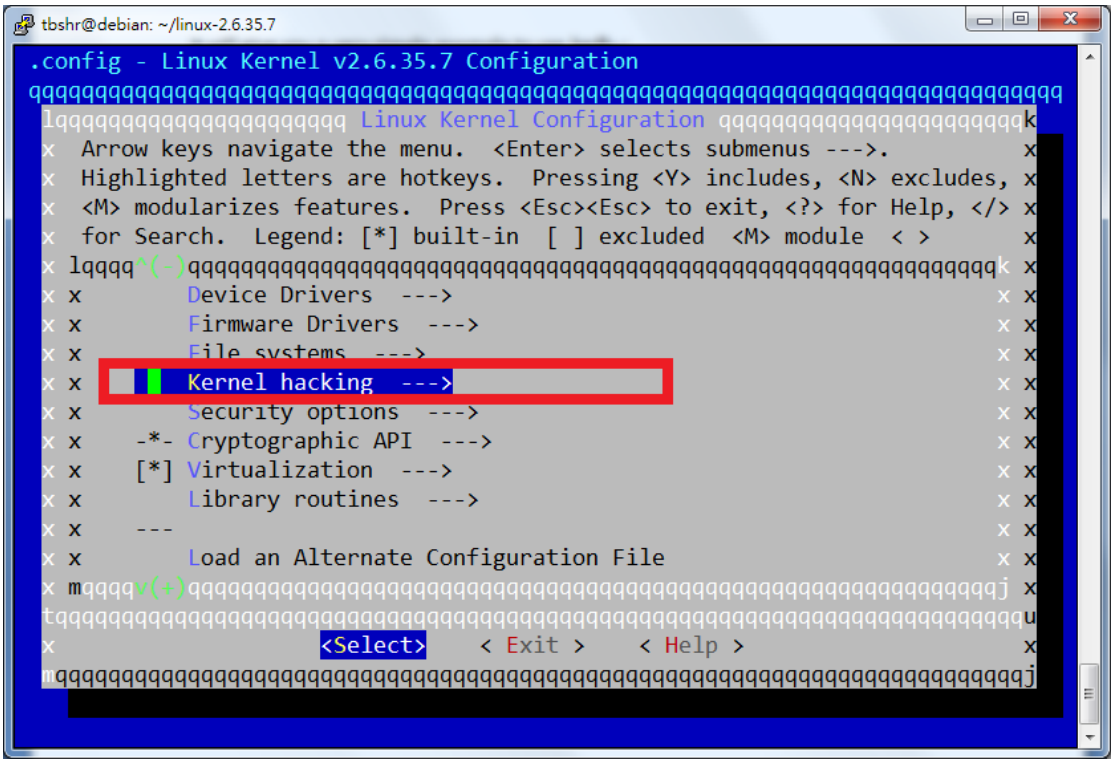

Figure 2

Please select option `Kernel debugging'.

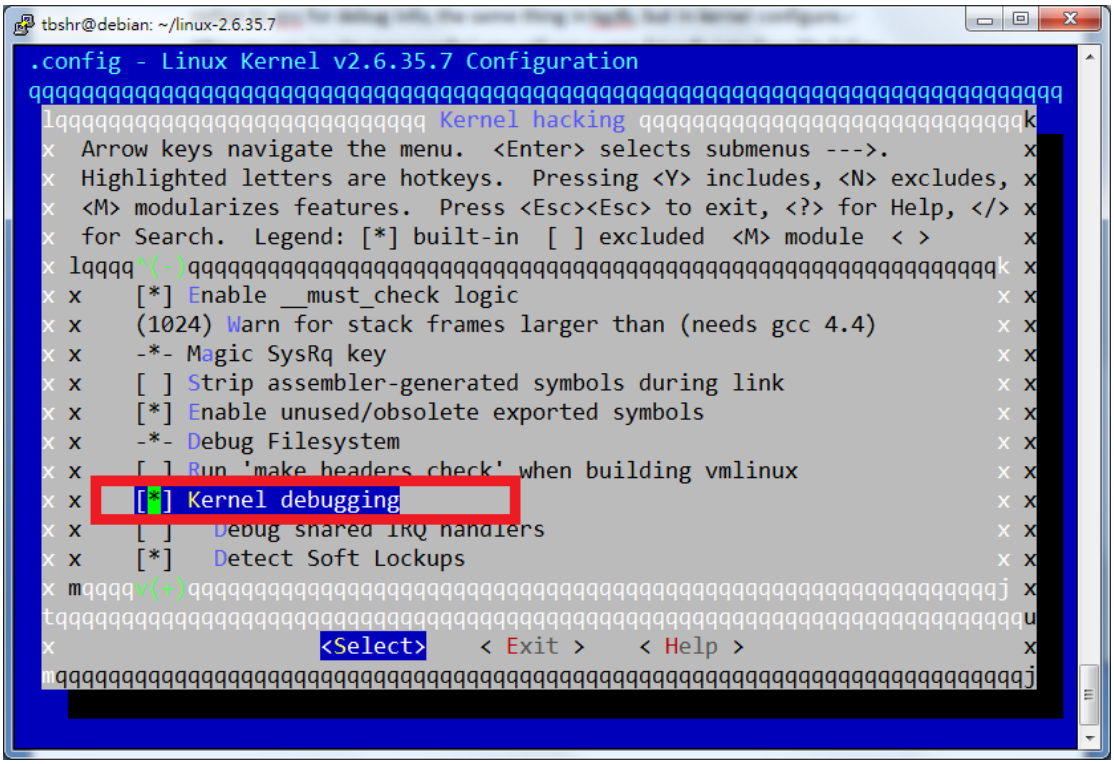

Figure 3

Please select option `Compile the kernel with debug info'.

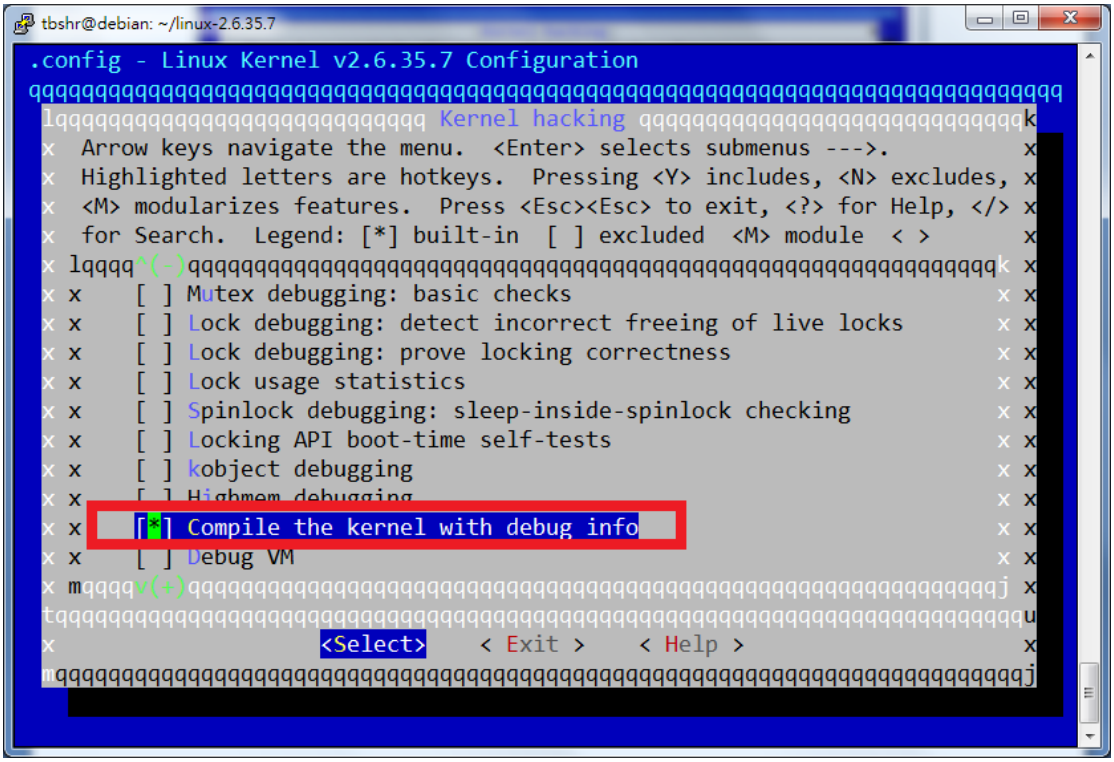

Figure 4

Please select option `Compile the kernel with frame pointers'.

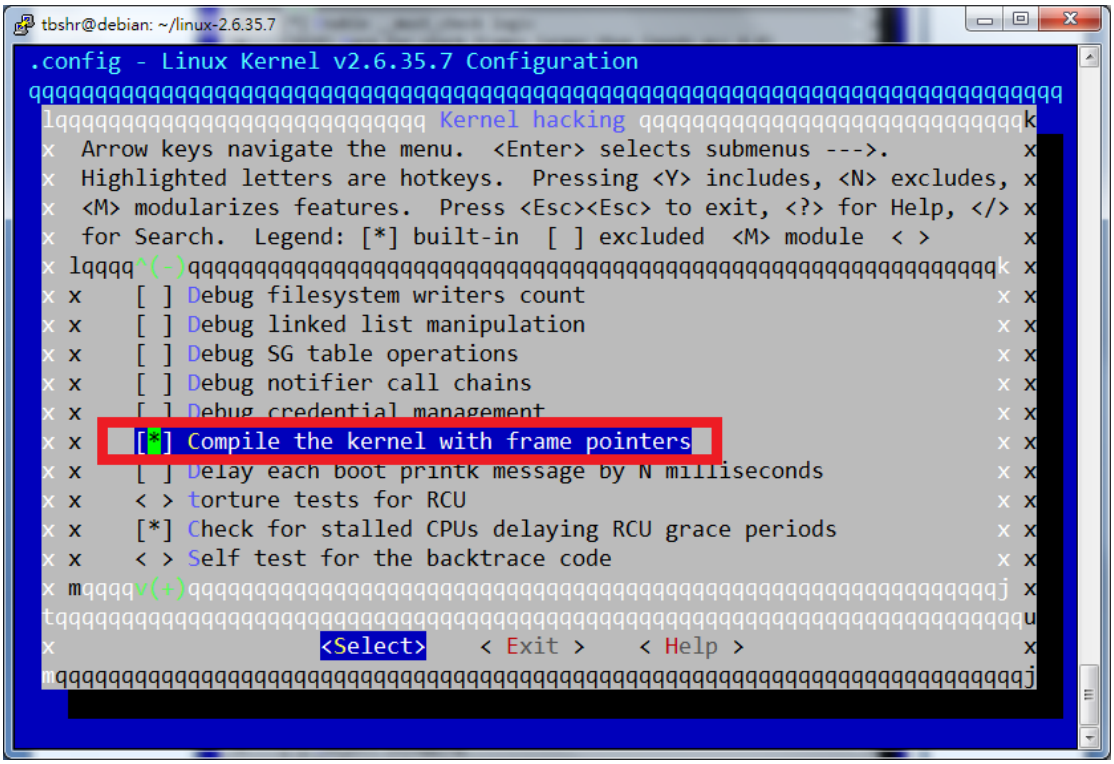

Figure 5

Please select option `KGDB: kernel debugger' like figure 6, and enter the `KGDB: kernel debugger' entry.

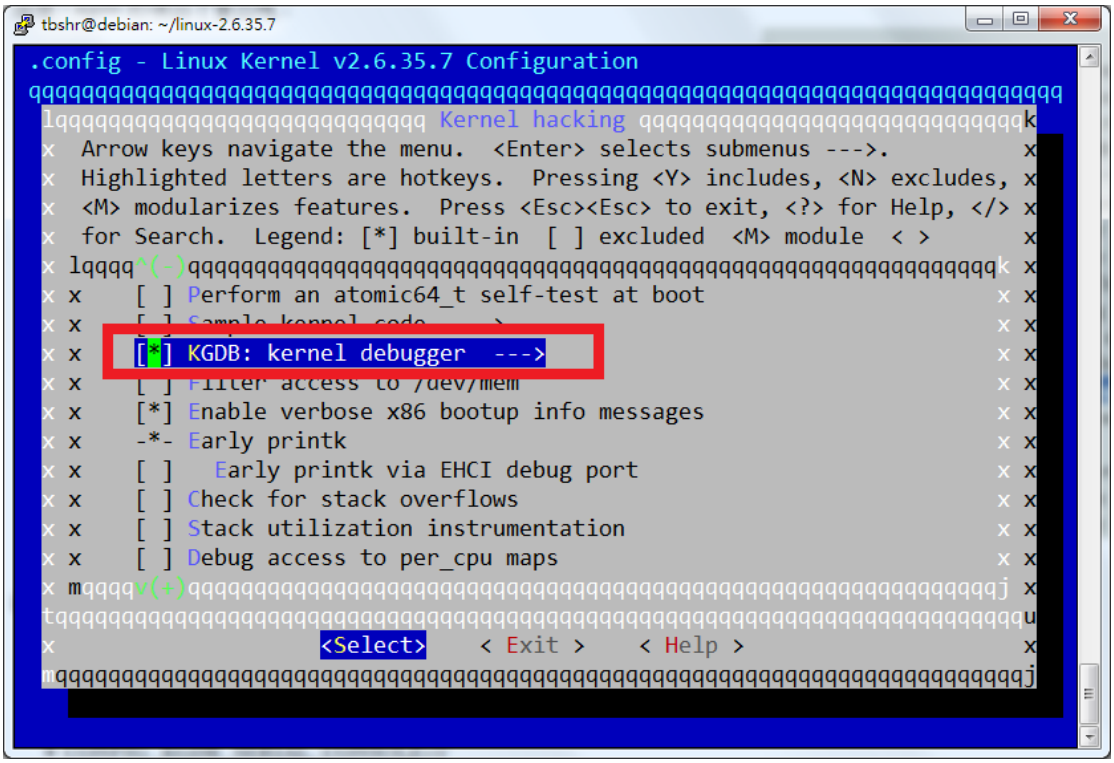

Figure 6

Because we want to use serial port rs232 in kgdb, please select option `KGDB: use kgdb over the serial console'.

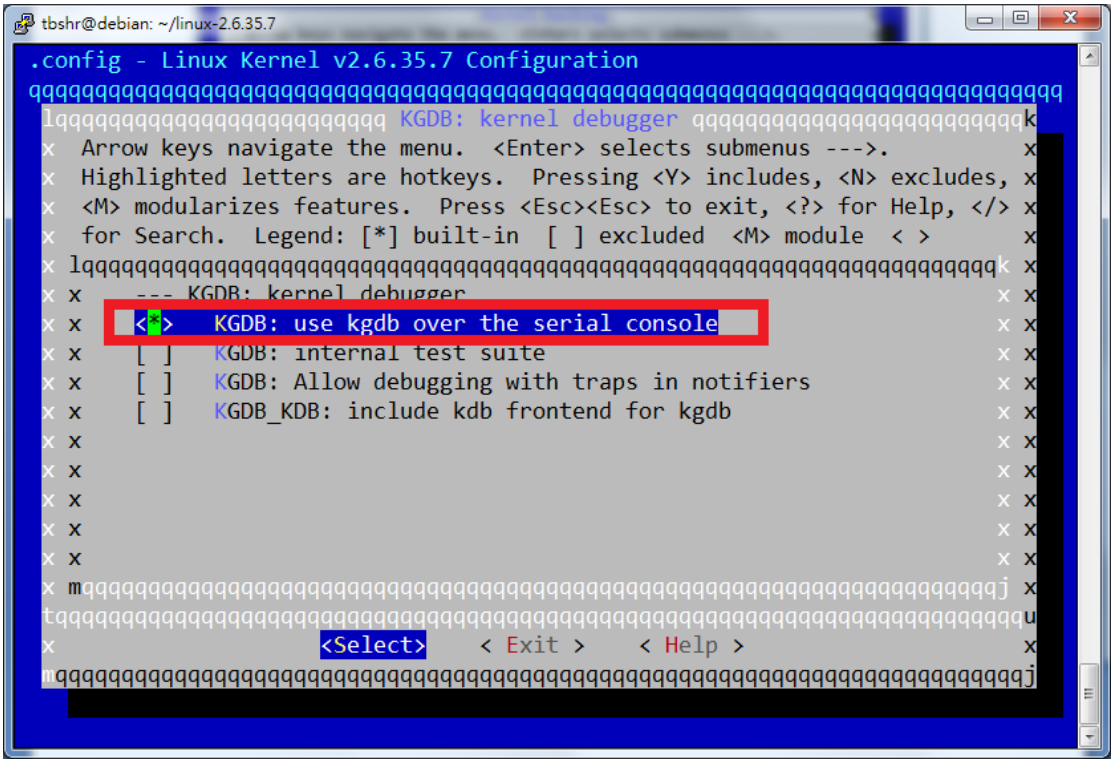

Figure 7

And then go through the same procedures to rebuild kernel. But in `make install' stage, we recommend you backup the old one.

(

- mv vmlinuz-2.6.35.7 vmlinuz-2.6.35.7.old;
- mv System.map-2.6.35.7 System.map-2.6.35.7.old;
- mv initrd.img-2.6.35.7 initrd.img-2.6.35.7.old;
- mv config-2.6.35.7 config-2.6.35.7.old

)

Edit boot manager configuration like figure 8. We use `grub' here since most distributions will install `grub' as default boot manager. If new kernel crash or panic, we can use old kernel to boot up.

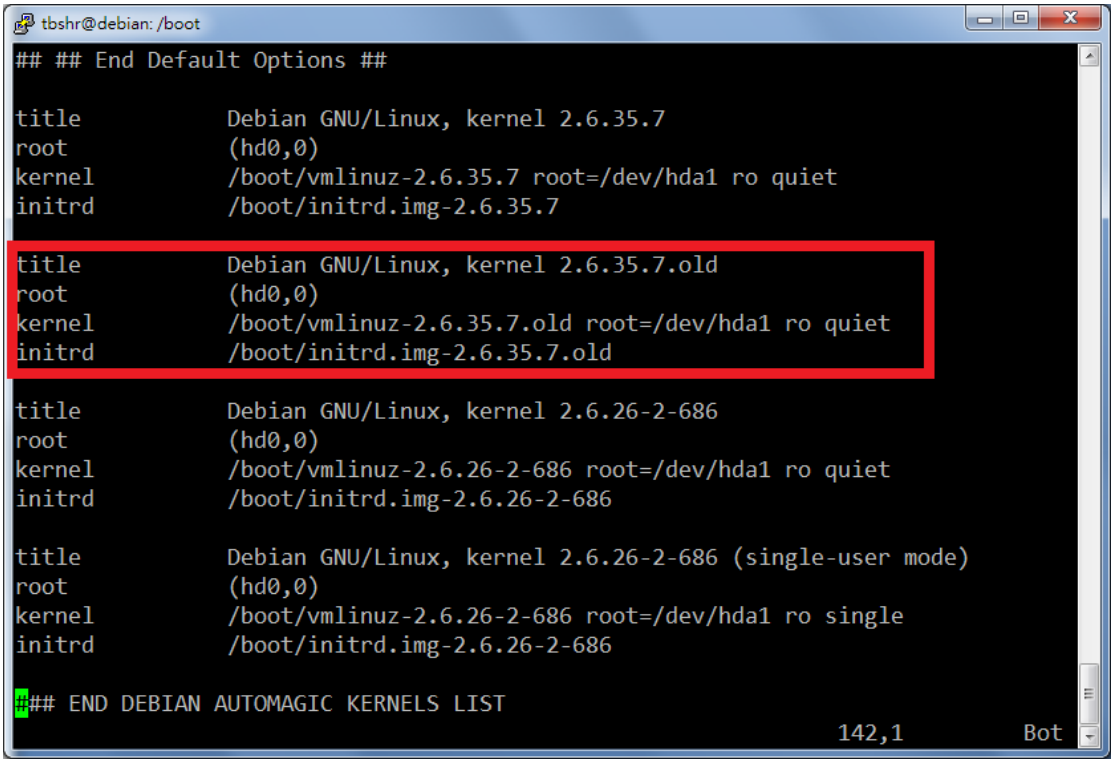

Figure 8

Then, please shut down your linux by `init 0'.

Step 2

In step 1, we use a machine named `LabDebian2', its virtual hard disk is at `${HOME}/.VirtualBox/HardDisks' by default.

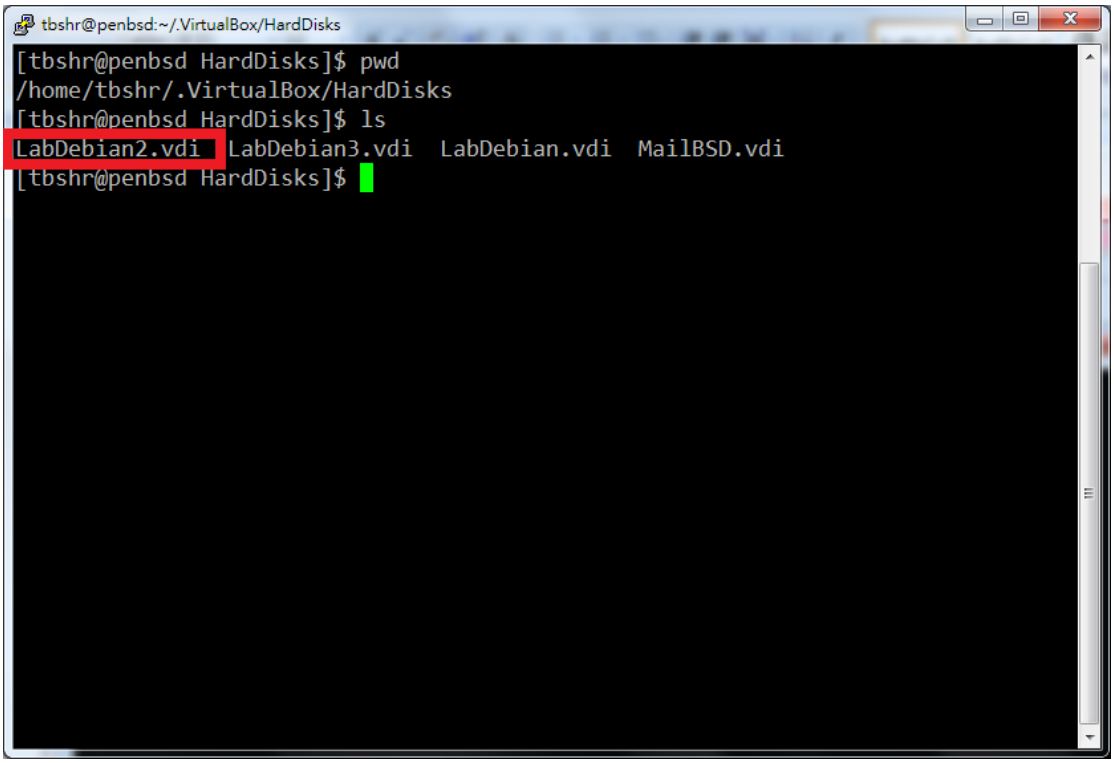

Figure 9

We need two linux virtual machines, one is observer and the other is observee. We can use `VBoxManage clonehd LabDebian2.vdi LabDebian3.vdi' to clone another one. Then, we configure these two virtual machines in logically serial connection.

In figure 10, we configure `LabDebian3'. Note that we

- select port mode to `Host Pipe'
- select `Create Pipe' option
- use path `/tmp/vbox1' (this path can be changed whatever you like)

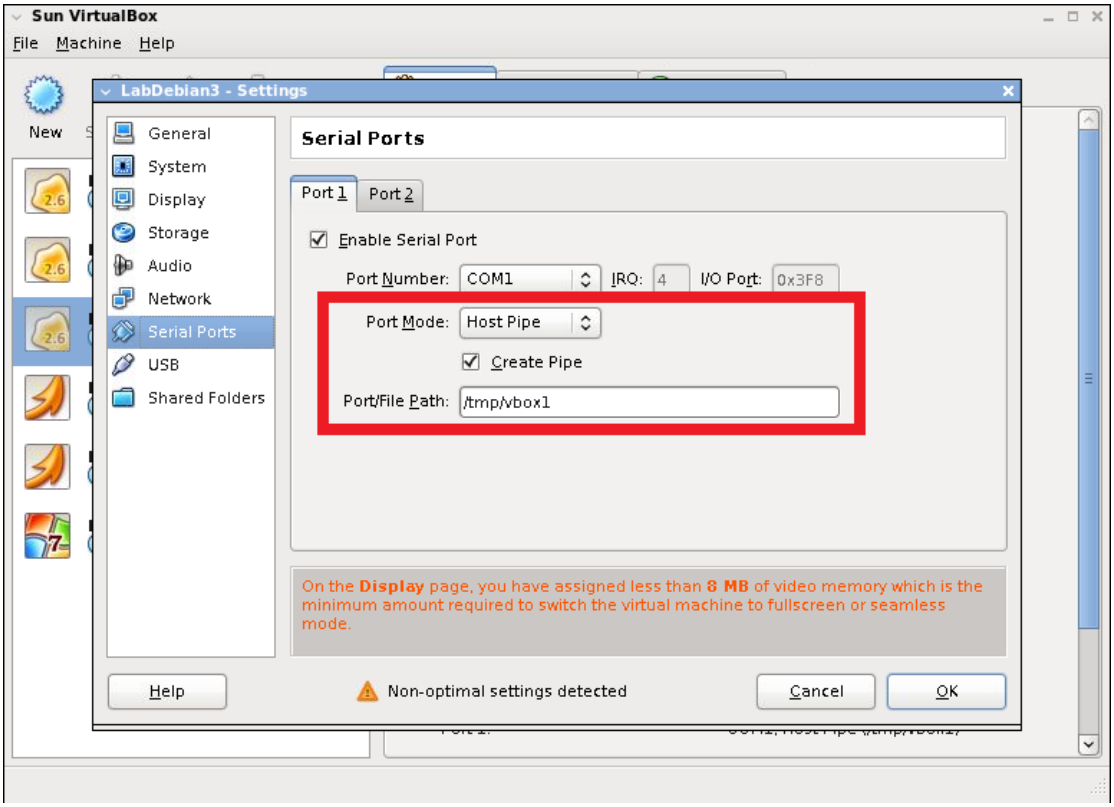

Figure 10

In `LabDebian2', we do the same thing, but do not select `Create Pipe' option.

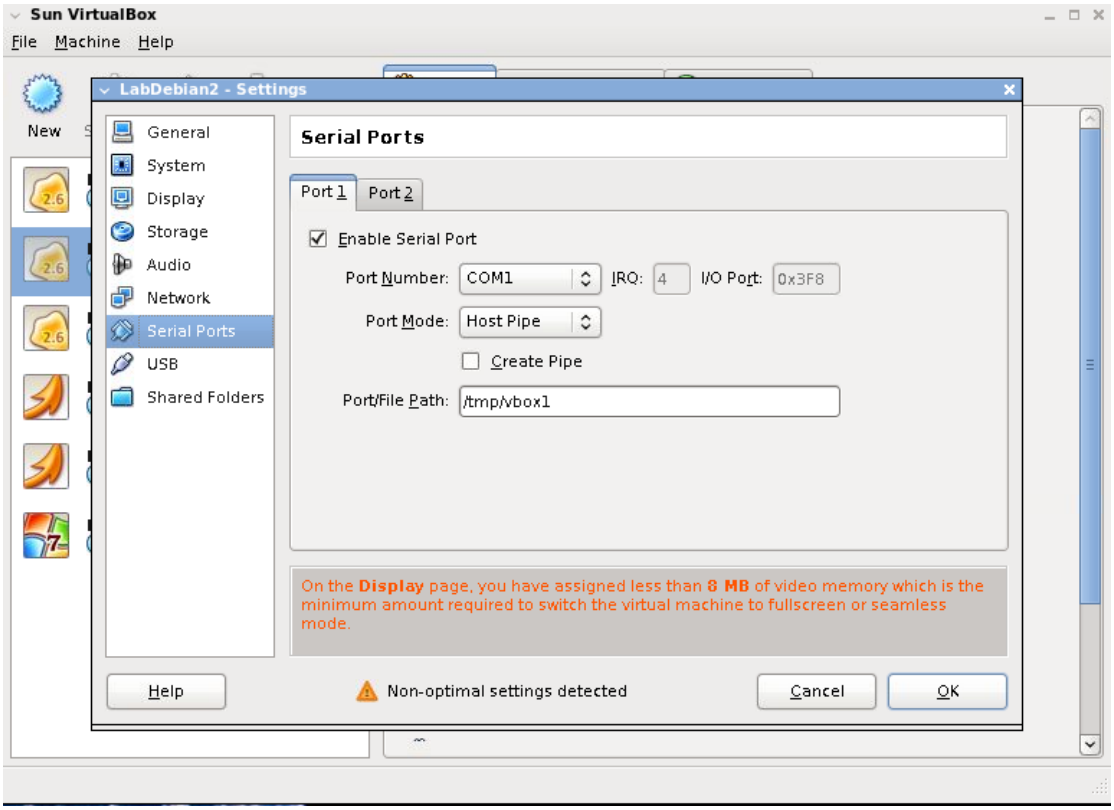

Figure 11

Step3

Please start the `LabDebian3'. When enter the grub menu press `e' on first entry. Note that we should start `LabDebian3' first because it will `create' the pipe.

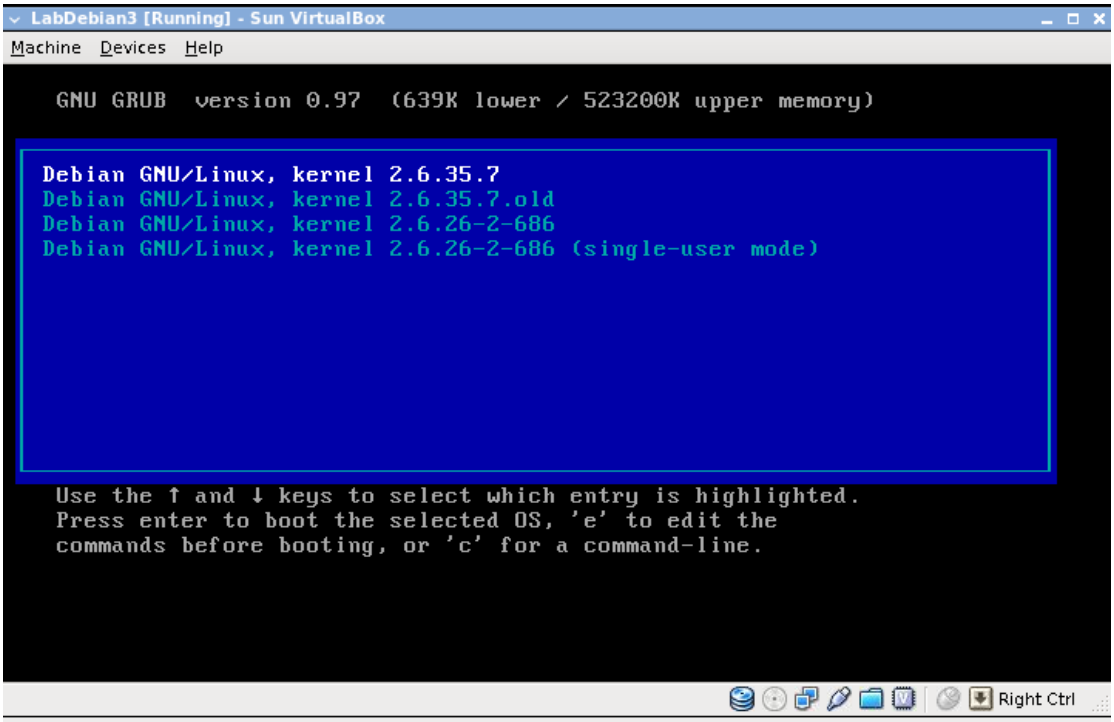

Figure 12

Please move highlight bar to second entry then press `e'.

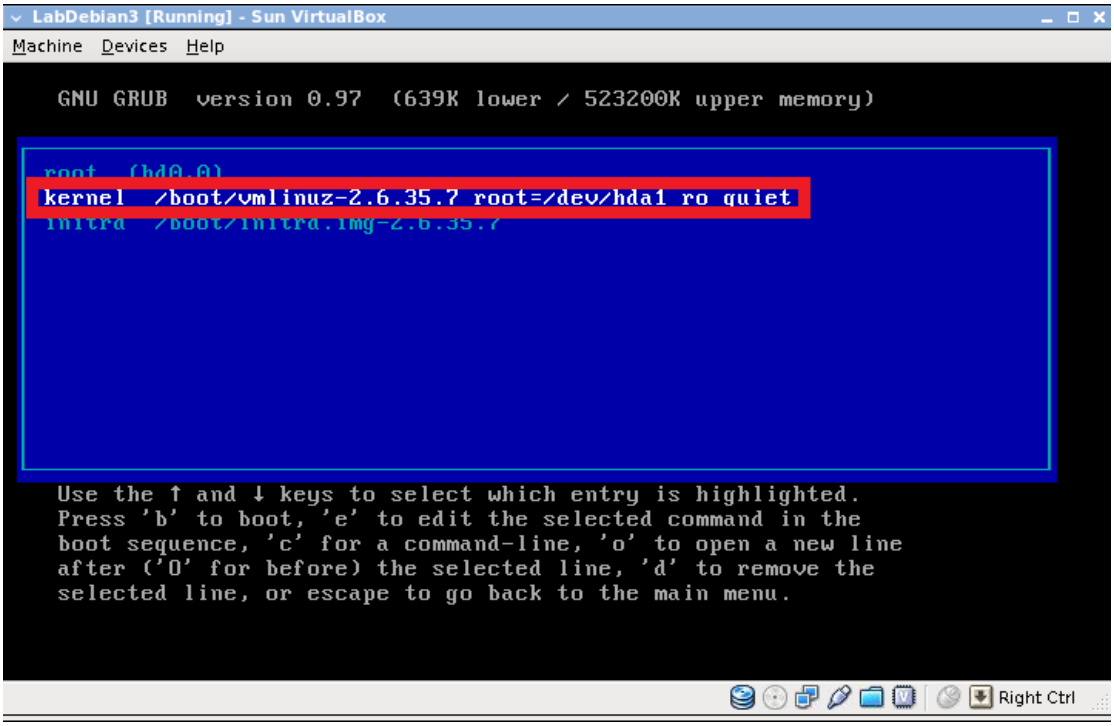

Figure 13

Append `kgdboc=ttyS0, 115200 kgdbwait' (You may need replace ttyS0 if you do not use serial port 1) then press enter.

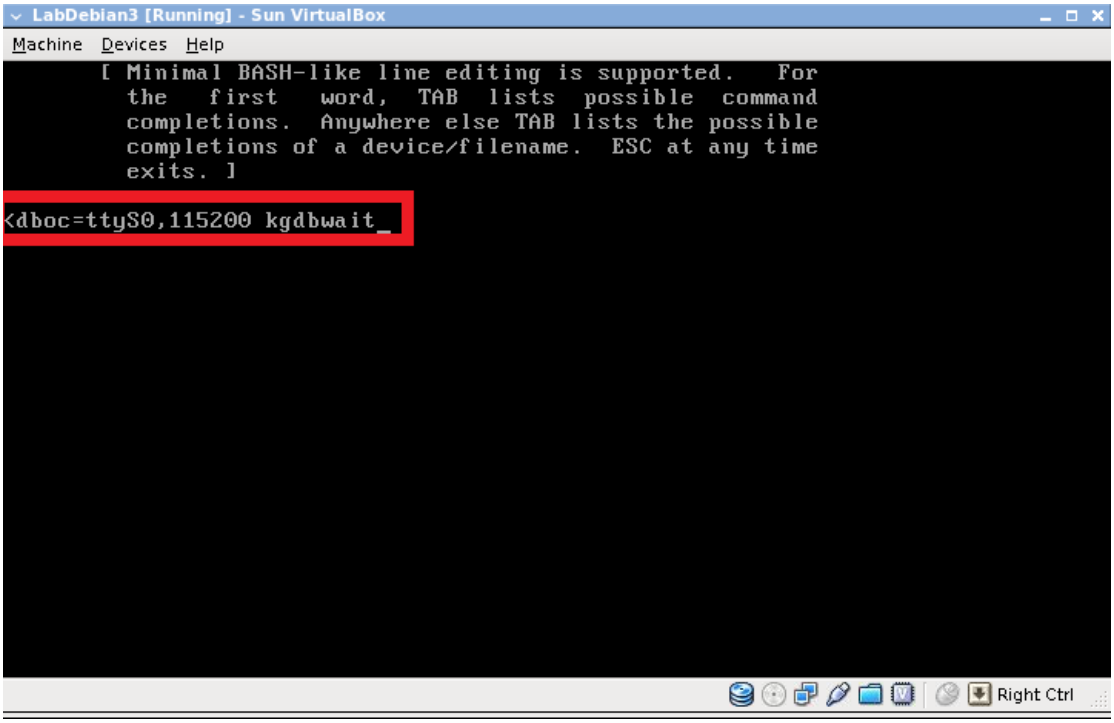

Figure 14

Back to this menu and press `b' to boot.

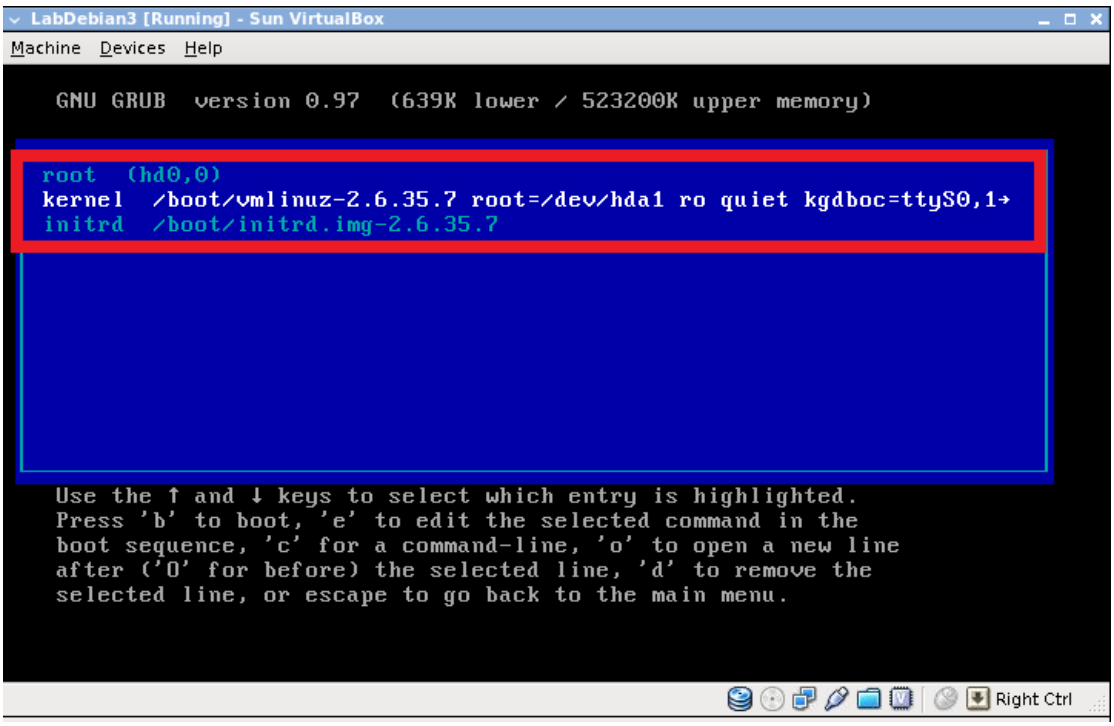

Figure 15

You may see messages like figure 16.

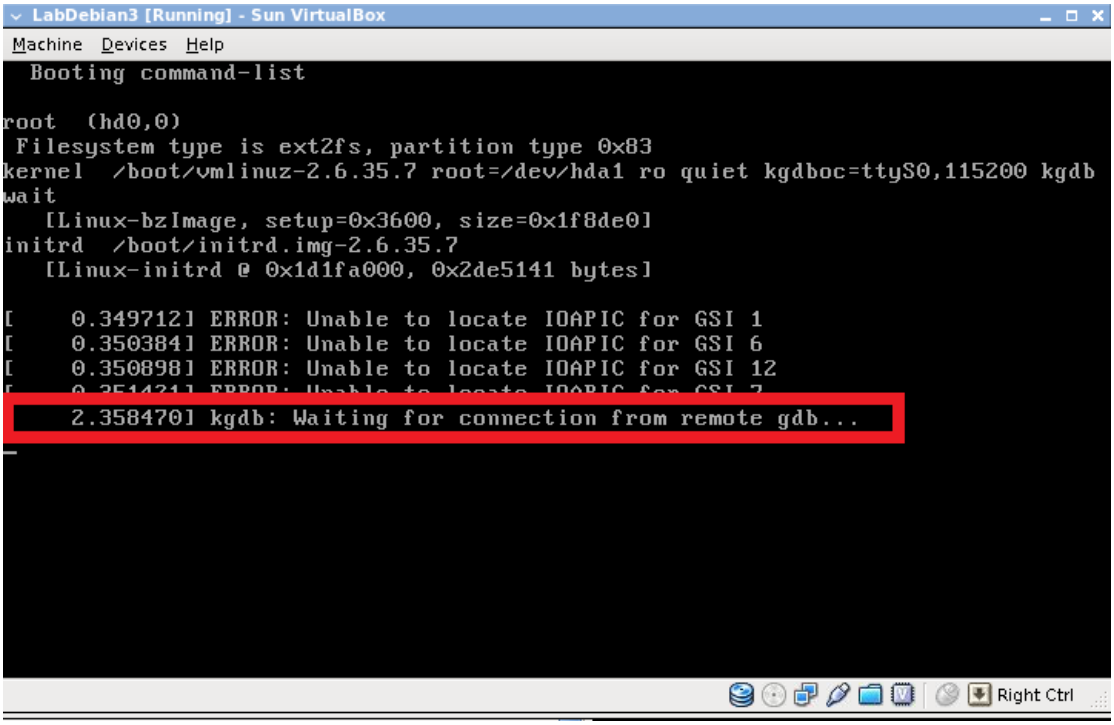

Figure 16

Please start `LabDebian2'. Here is a little tricky that we need `make vmlinux', so that we will have an uncompressed linux image the same to `LabDebian3'.

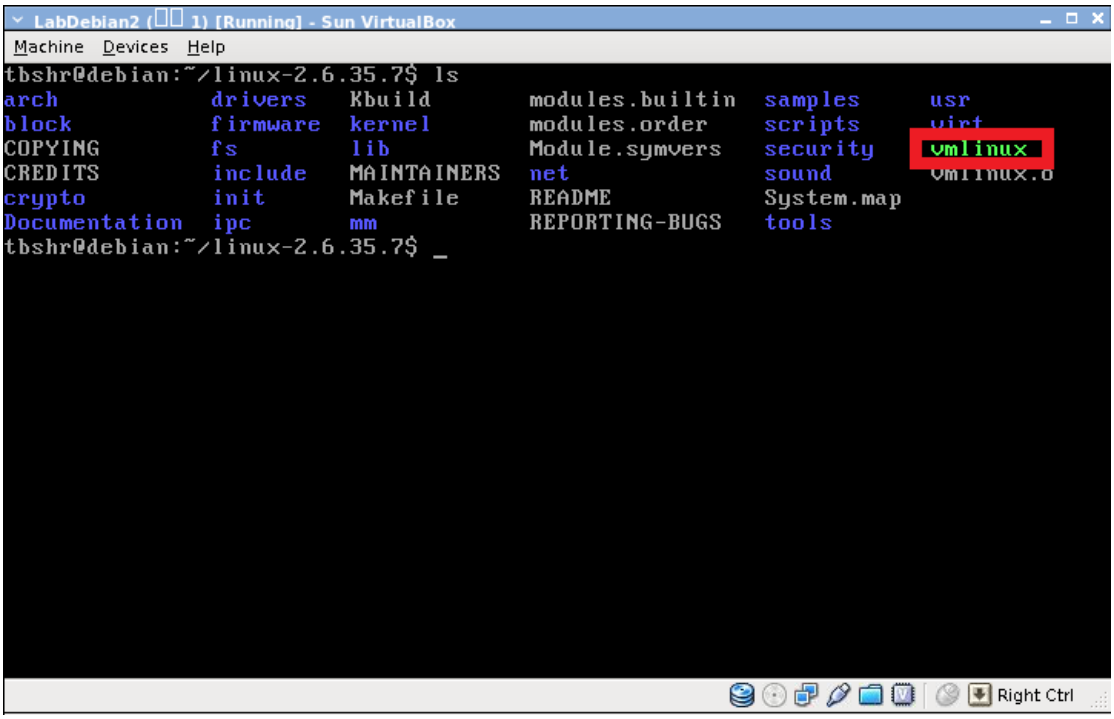

Figure 17

Please use `gdb vmlinux' to launch gdb. After enter gdb, please use

- `set remotebaud 115200'
- `target remote /dev/ttyS0'

to setup kgdb.

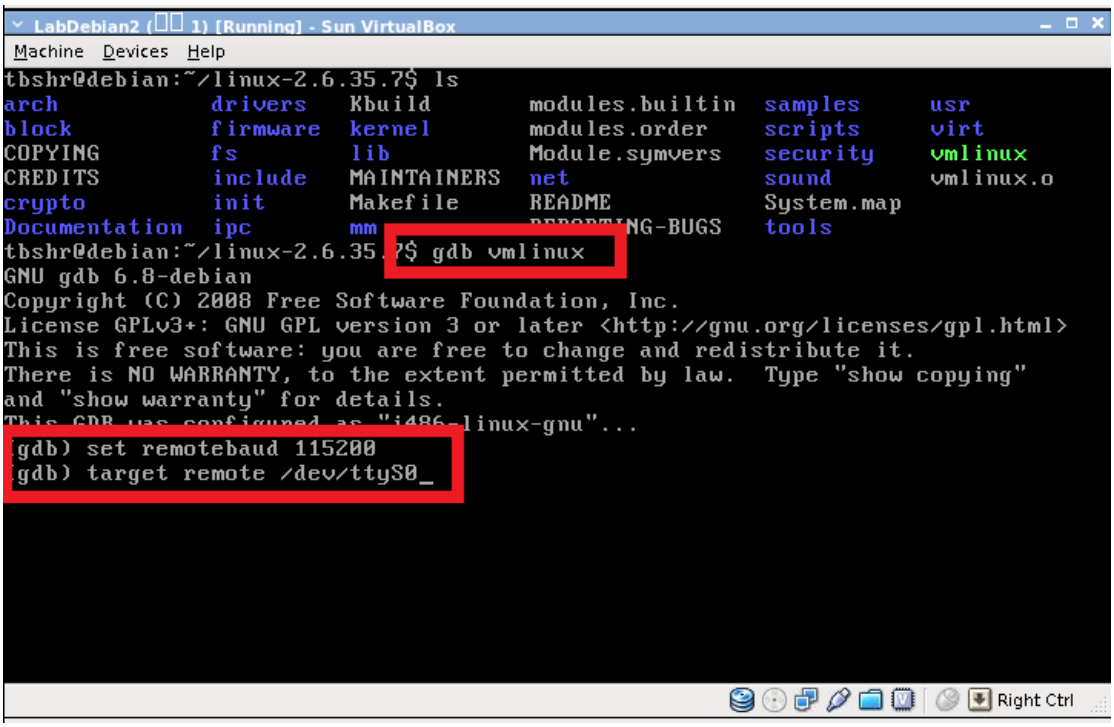

Figure 18

After press enter, enjoy your kgdb.

```
LabDebian3 [Running] - Sun VirtualBox
Machine  Devices  Help
  Booting command-list

root  (hd0,0)
 Filesystem type is ext2fs, partition type 0x83
kernel  /boot/vmlinuz-2.6.35.7 root=/dev/hda1 ro quiet kgdboc=ttyS0,115200 kgdb
wait
   [Linux-bzImage, setup=0x3600, size=0x1f8de0]
initrd  /boot/initrd.img-2.6.35.7
   [Linux-initrd @ 0x1d1fa000, 0x2de5141 bytes]

[    0.349712] ERROR: Unable to locate IOAPIC for GSI 1
[    0.350384] ERROR: Unable to locate IOAPIC for GSI 6
[    0.350898] ERROR: Unable to locate IOAPIC for GSI 12
[    0.351421] ERROR: Unable to locate IOAPIC for GSI 7
[    2.358470] kgdb: Waiting for connection from remote gdb...
[ 1052.317373] INFO: rcu_sched_state detected stall on CPU 0 (t=262581 jiffies)
[ 1052.318076] Process swapper (pid: 1, ti=dcc2a000 task=dcc298a0 task.ti=dcc2a0
00)
[ 1052.318695] Stack:
[ 1052.318964] Call Trace:
[ 1052.319404] Code: ee a1 94 22 35 c1 c1 e6 18 89 f2 2d f0 3c 00 00 89 10 a1 94
 22 35 c1 89 fa 09 da 80 cf 04 83 ff 02 0f 44 d3 2d 00 3d 00 00 89 10 <8b> 45 f0
 50 9d 8d 74 26 00 8d 65 f4 5b 5e 5f 5d c3 90 90 90 8b
_
                                                          Right Ctrl

LabDebian2 (□□ 1) [Running] - Sun VirtualBox
Machine  Devices  Help
tbshr@debian:~/linux-2.6.35.7$ ls
arch            drivers   Kbuild        modules.builtin  samples     usr
block           firmware  kernel        modules.order    scripts     virt
COPYING         fs        lib           Module.symvers   security    vmlinux
CREDITS         include   MAINTAINERS   net              sound       vmlinux.o
crypto          init      Makefile      README           System.map
Documentation   ipc       mm            REPORTING-BUGS   tools
tbshr@debian:~/linux-2.6.35.7$ gdb vmlinux
GNU gdb 6.8-debian
Copyright (C) 2008 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i486-linux-gnu"...
(gdb) set remotebaud 115200
(gdb) target remote /dev/ttyS0
Remote debugging using /dev/ttyS0
kgdb_register_io_module (new_dbg_io_ops=<value optimized out>)
    at kernel/debug/debug_core.c:967
967             wmb(); /* Sync point after breakpoint */
(gdb) s
[New Thread 1]
107             asm volatile(LOCK_PREFIX "decl %0"
(gdb) _
                                                          Right Ctrl
```

Figure 19

We just introduce basic usage about kgdb here.

References

[1] http://fotis.loukos.me/blog/?p=25
[2] http://blog.linux.org.tw/~jserv/archives/002045.html (in Chinese)

Keywords you may use: `kgdb' `kgdb virtualbox' `kgdb vmware' …