Operating System

project 1A

Department: 土木5B
ID: 0511330
Name: 劉紘華
Vedio Link: https://youtu.be/YTXkAP0tN2c

Goal

Get familiar with tools which is used to setup an environment for Linux kernel including kernel debugging, compilation and profiling.

Basic concepts and Tools Introduction

1. **GDB:**
   GNU **D**ebugger, 支援C, C++, Fortran等高階語言的除錯，主要用在user program (application)的開發，特色是支援Remote debugging模式，可用在embedded system開發，能設定breakpoint方便特定的程式碼區段除錯。

2. **KGDB:**
   Kernel **G**NU **D**ebugger, 支援**Linux**, NetBSD和FreeBSD的kenrnel除錯，Serial Connection在透過UDP/IP protocol連接**Target** Machine & **Host** Machine，對Target Machine的kernel進行Remote Debug。

   註: Target Machine is the one **being debugged** and running the patched kernel.
   Host Machine is the one which runs gdb to debug the target.

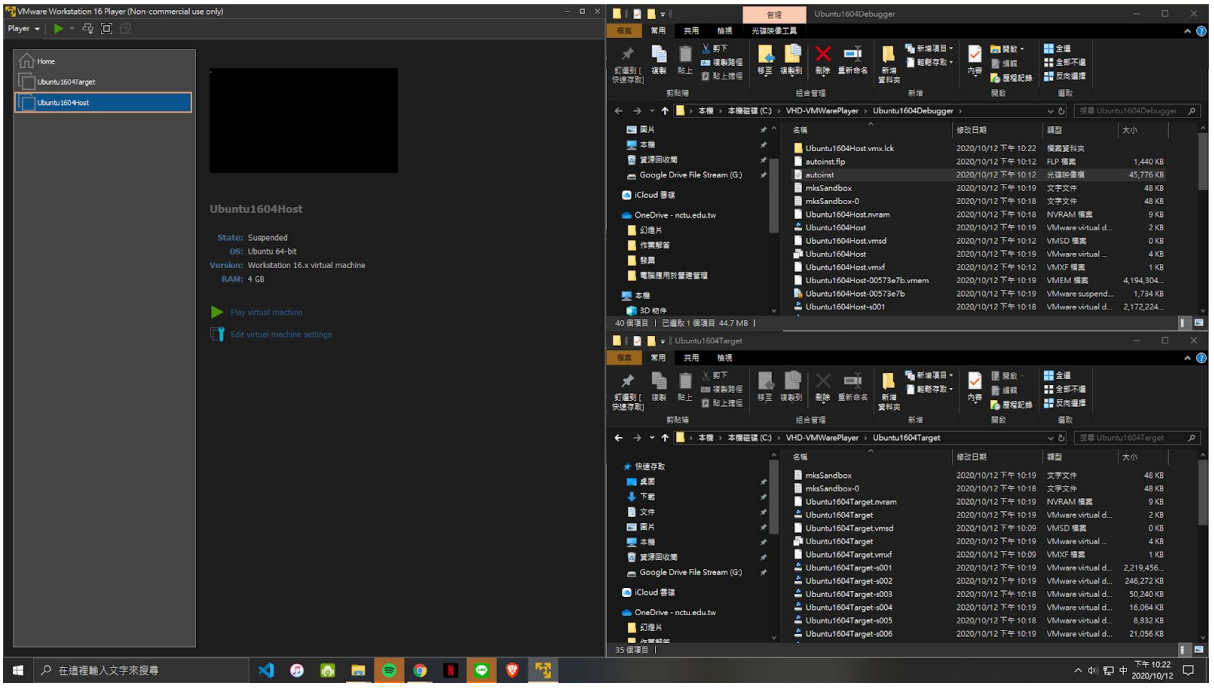
3. **Grub:**
   GNU Grub是一個Opensouce的專案，是一種bootloader，讓電腦可以安裝多種作業系統，幫助引導到使用者到欲使用的作業系統上。

4. **SSH and OpenSSH:**
   SSH是一種通訊協定(protocol)，在Client-Server model間傳遞的訊息，都被會經過加密，只有在認證的機器上才能被安全解密；OpenSSH是一個OpenSource的套件，能透過Command Line Interface在Linux上安裝並執行。

[Section 1-1]

Screenshot 1 - Virtual Machine setup

When doing kernel debugging, we need two machines - **Target** and **Host** mentioned above. Consequently, we use VMware to simulate the Target and Host, which provides us to run two machines independently just on a single PC.



[Section 1-2]

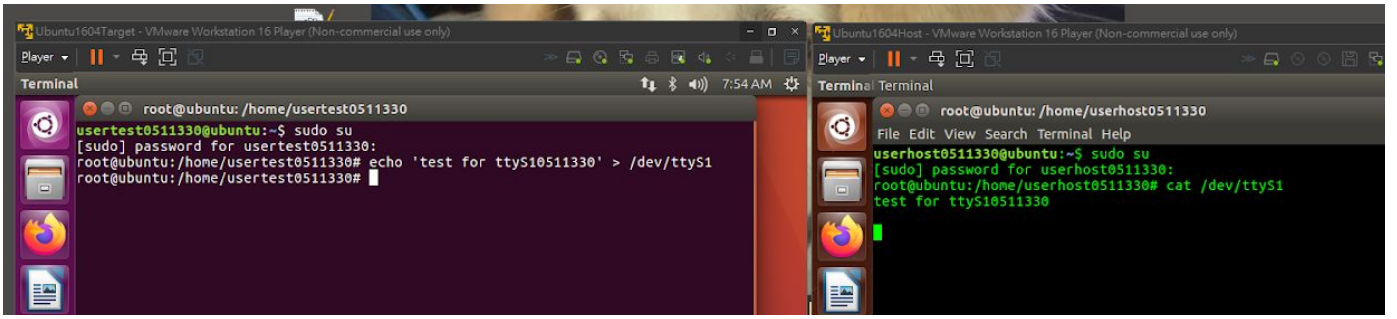
Screenshot 2 - Verifying the connection is built correctly.

In the real world, we need to connect two computers through wired (e.g. ethernet) or wireless (e.g.WiFi); But for our two virtual machines, we can simply build the physical connection through “Serial port”. To check the port was built correctly, using commands as follows:

Host: **# cat /dev/ttyS1**

Target: **#echo ‘test for ttyS1 0511330 > /dev/ttyS1**

The host will capture the message from /ttyS1 port, and the target is making an echo message in the port ttyS1 (like broadcasting).

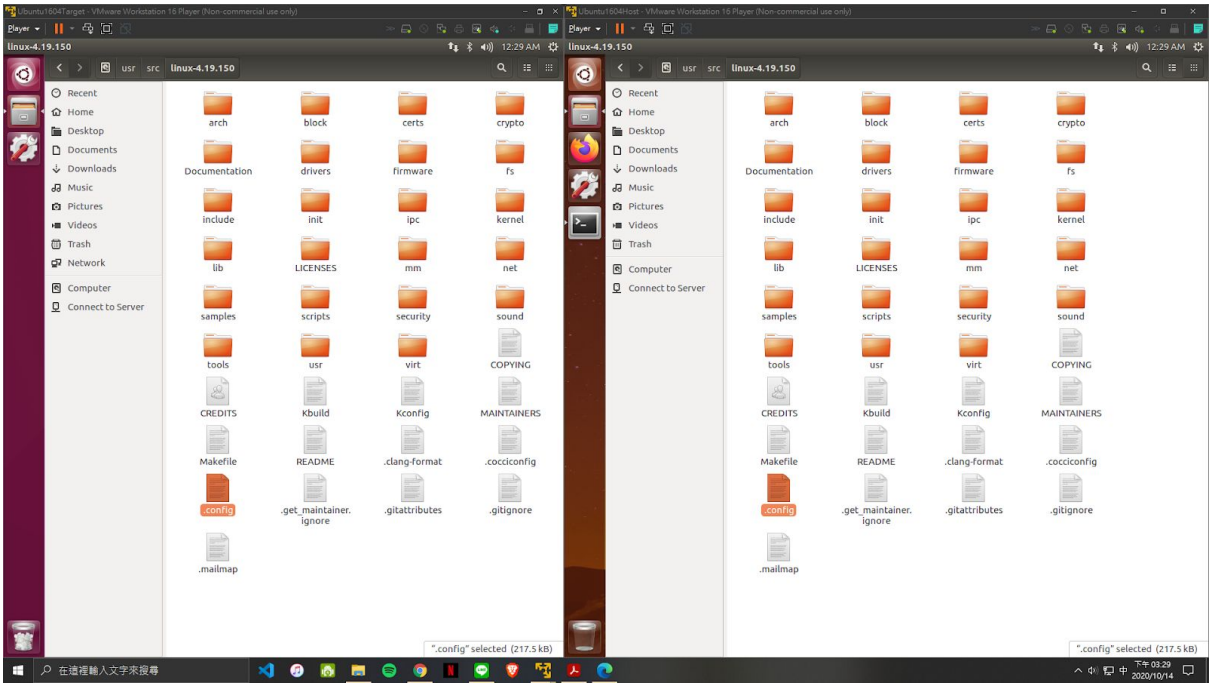


[Section 2-1~2-2]

Screenshot 3 - Download the source code for Linux kernel

We download the Linux source code, and use the following command to prebuild the .config file: **$sudo cp -v /boot/config-$(uname -r) .config**

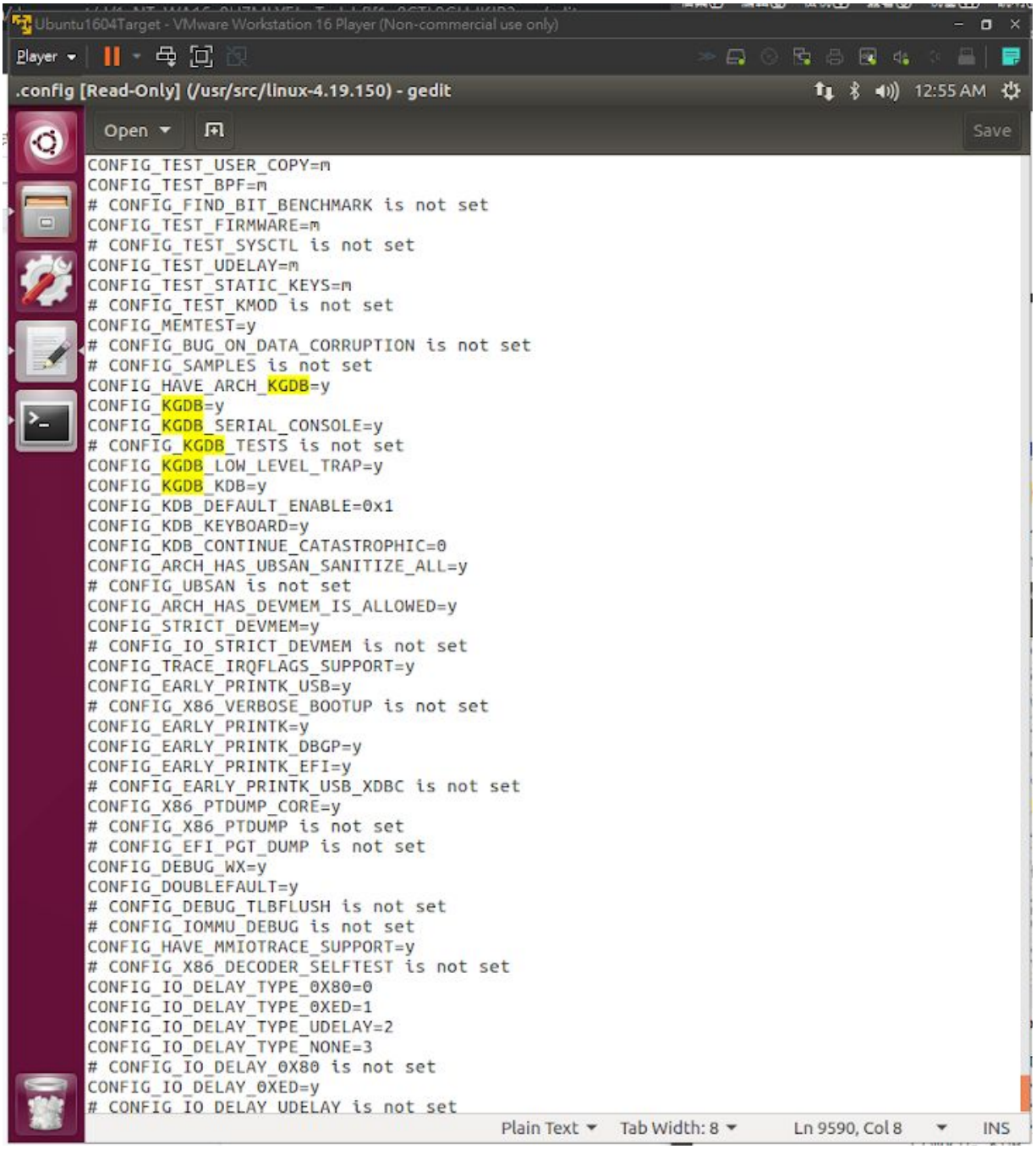
Config file like a function table of the kernel, allowing us to modify the functions which told the underlying hardware how to work. For example, we can set the functions like power management options, virtualization, I/O bus, etc.



[Section 3-1~3-2] Linux kernel patching

Screenshot 4 - check the KGDB setup

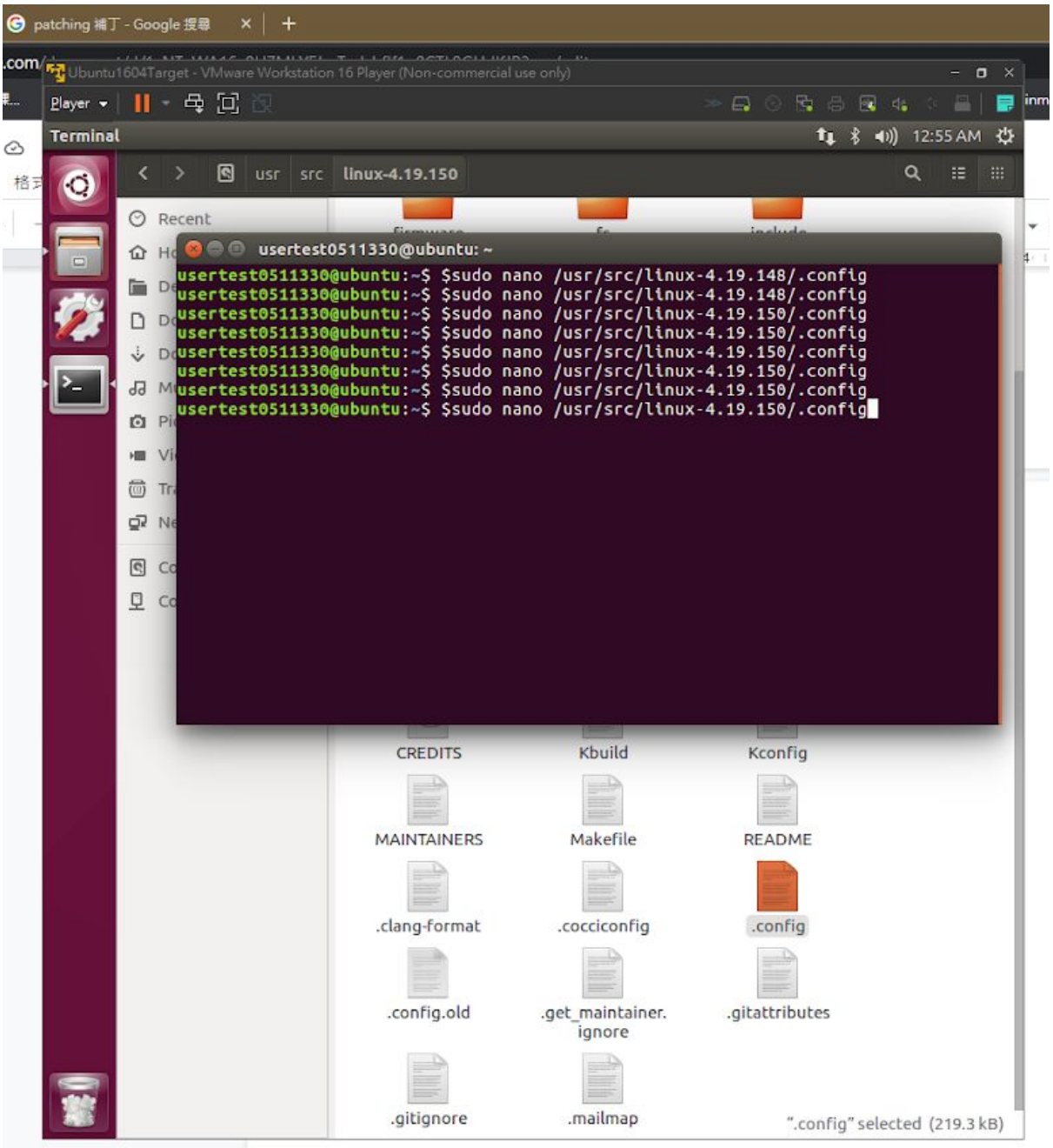
checking the KGDB settings in .config file is set to true. The thing we modify the settings in the.config file calls kernel patching. Use “Ctrl” + “F” to search the keyword KGDB, and remember the line shown on the right corner. (9590).



Screenshot 5 - modify the KGDB settings through CLI

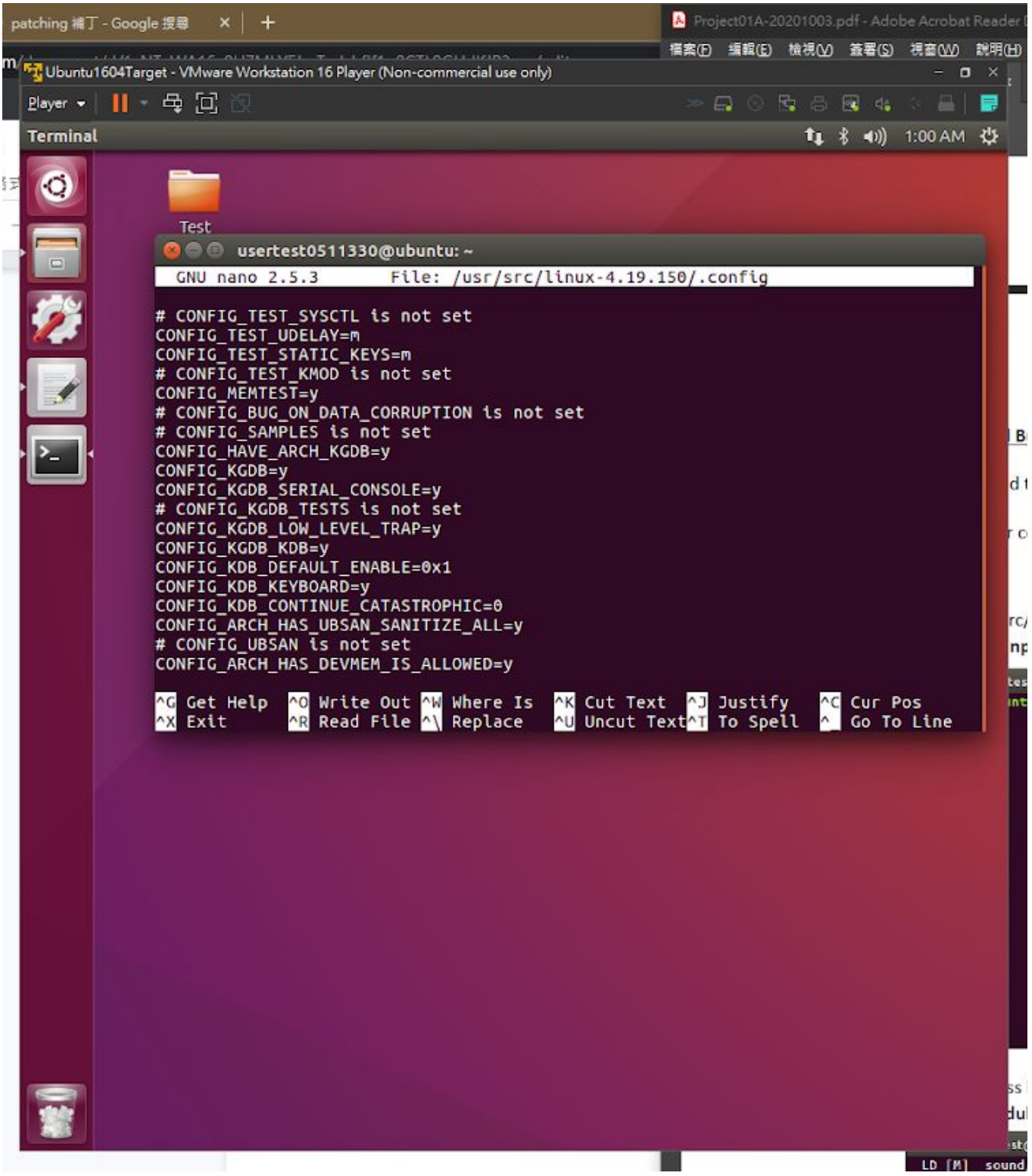
if you need to change some settings, use the command:

$sudo nano /usr/src/linux-4.19.150/.config



Screenshot 6 - check the KGDB setup

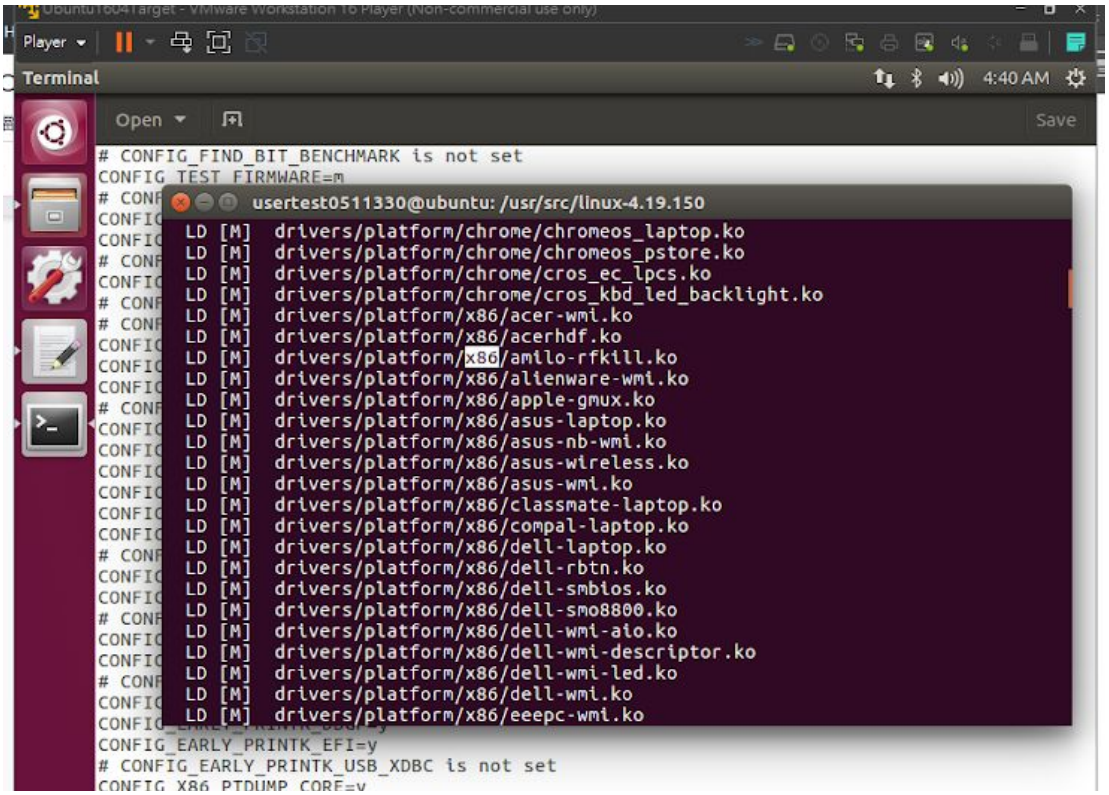
press the button of combination "Ctrl" + "Shift" + "-" to go to line:9590 ("-" is the button next to "=" on keyboard) and you can modify it.



[Section 3-1~3-2] proceed to build the Linux kernel

Screenshot 6 - Compilation to .ko
We compile it to .ko files (kernel object code).
Use the command: **$sudo make -j $(nproc)**

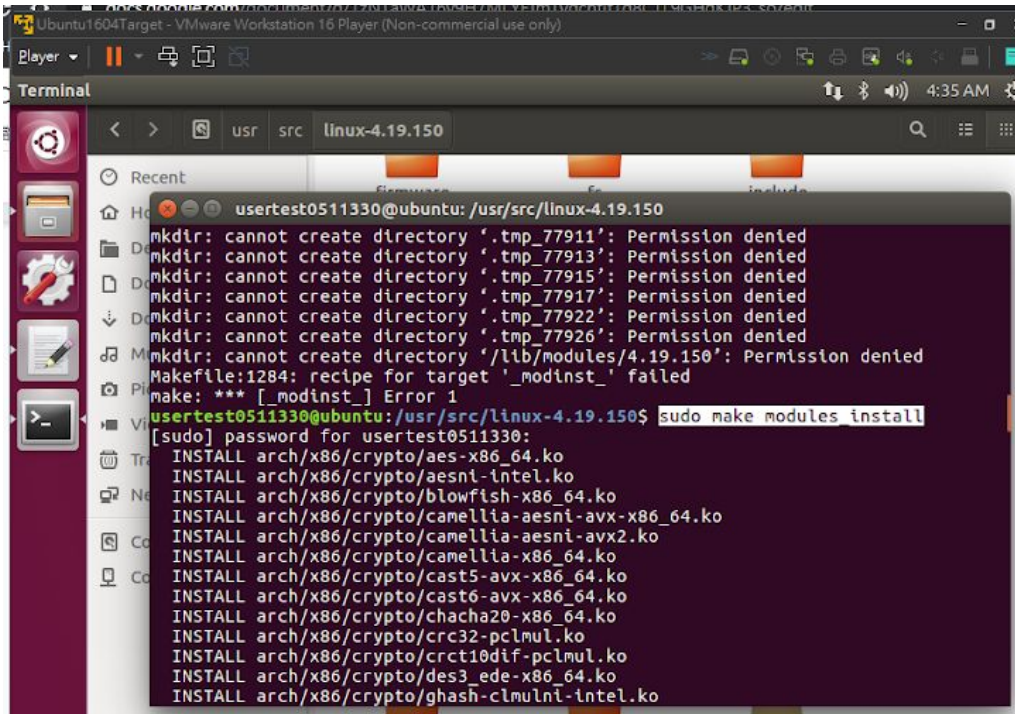
x86!!

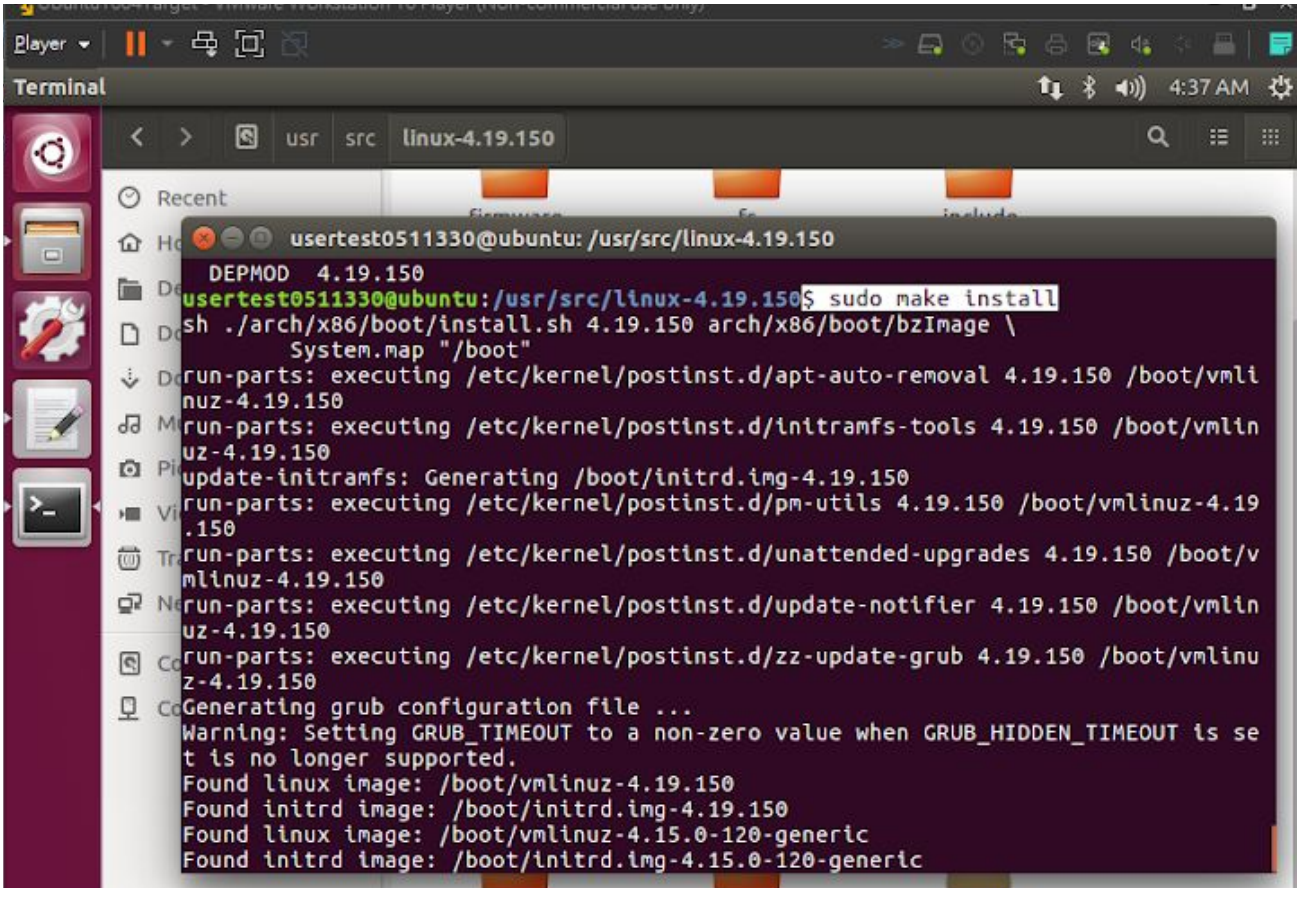


Screenshot 7, 8 -

use the command: **$sudo make modules_install**, **$sudo make install**

Install the modules which we've just compiled to the default path.





[Section 3-4(2)] Grub update(2)

Screenshot 9

We update the grub through the commands:

$sudo update-initramfs -c -k 4.19.150

$sudo update grub

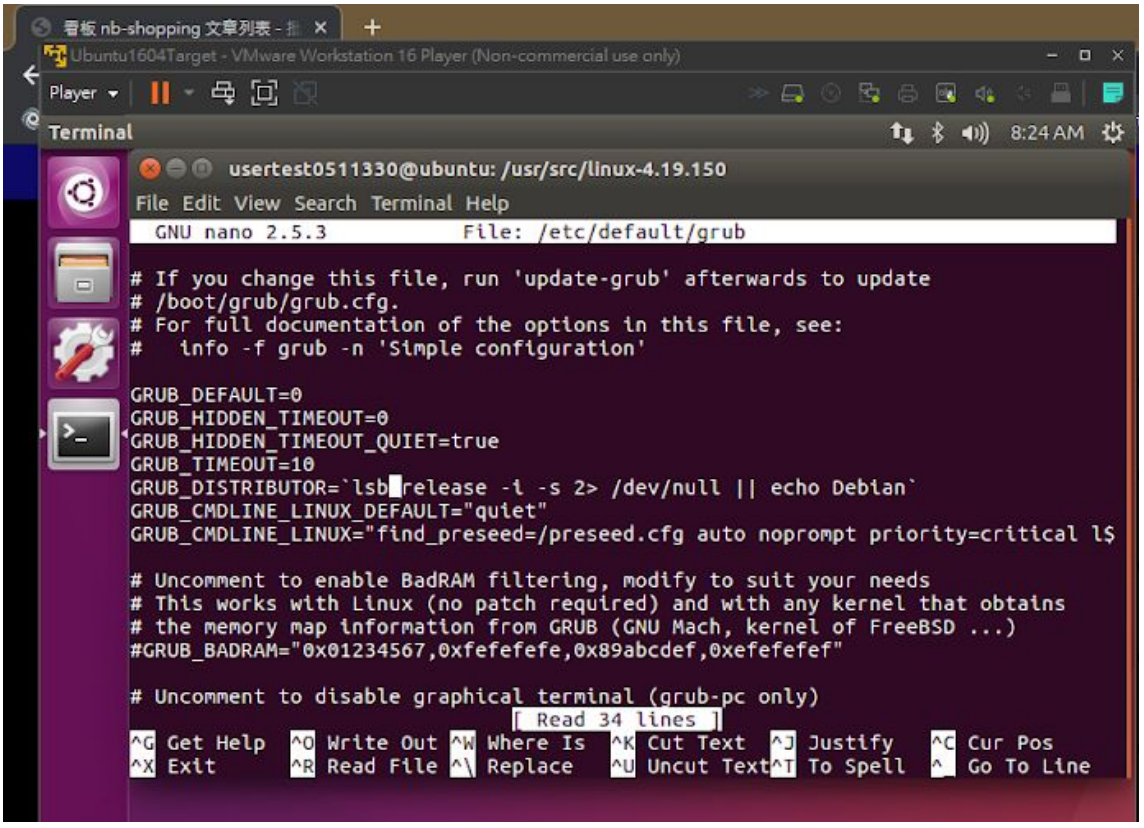
then we modify the settings in grub files through the command:

$sudo nano /etc/default/grub

disable the following 2 settings.

#GRUB_HIDDEN_TIMEOUT=0

#GRUB_HIDDEN_TIMEOUT_QUIET=true



Screenshot 10, 11 - **Assign the serial port for gdb connection.**

Open the grub.cfg under the path ***/boot/grub/grub.cfg***, and search for the keyword “Ubuntu”. In addition, Remember the line number.

Then we use nano to add two commands at the end of the line :

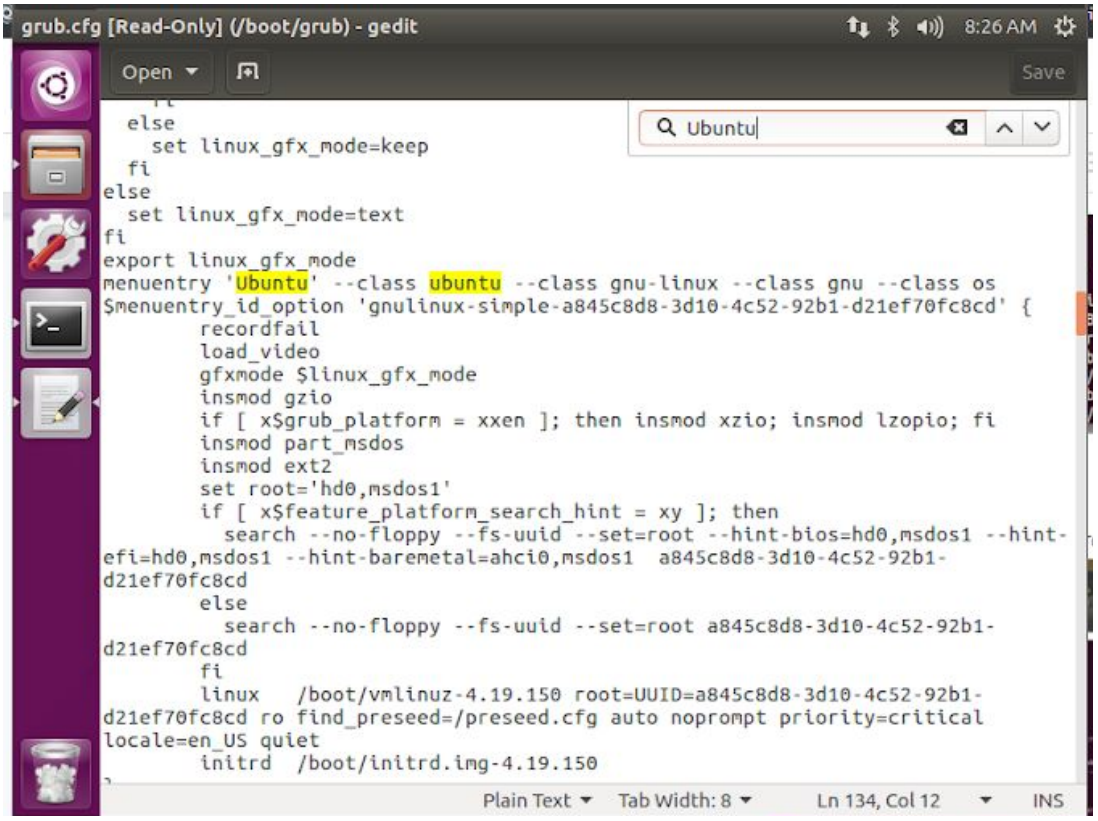
kgdbwait

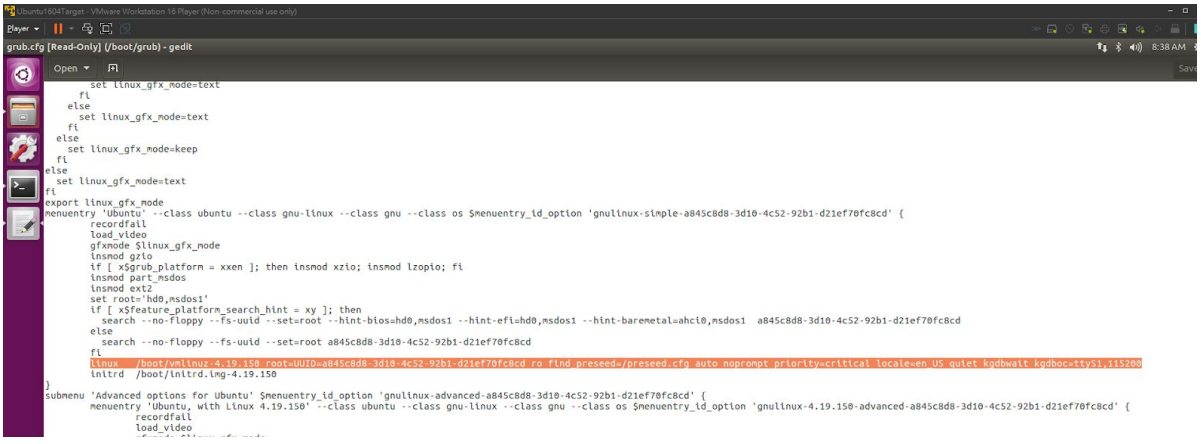
kgdboc=ttyS1,115200

kgdbwait: when we boot the machine, the kernel will wait for the gdb connection.

kgdboc: kgdb **o**ver **c**onsole, we choose the second serial port (**ttyS1**) as console, and the baud rate is set to **115200** (bits per second).

After things being settled down, when we boot the machine, the kernel will wait for the gdb connection and we can debug remotely.



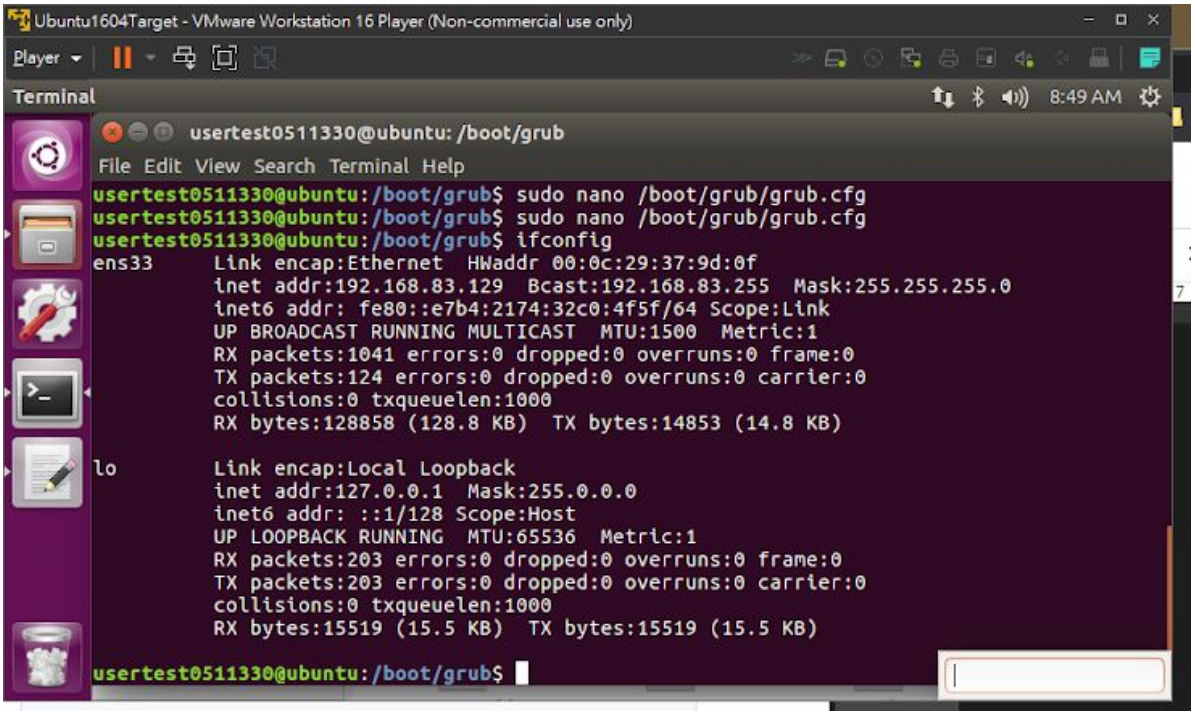


Screenshot 12 - **building ssh protocol**

The scp command means we can remotely command the target through the ssh protocol.

$sudo scp usertest0511330@192.168. 83.129:/usr/src/linux-4.19.150/vmlinux .

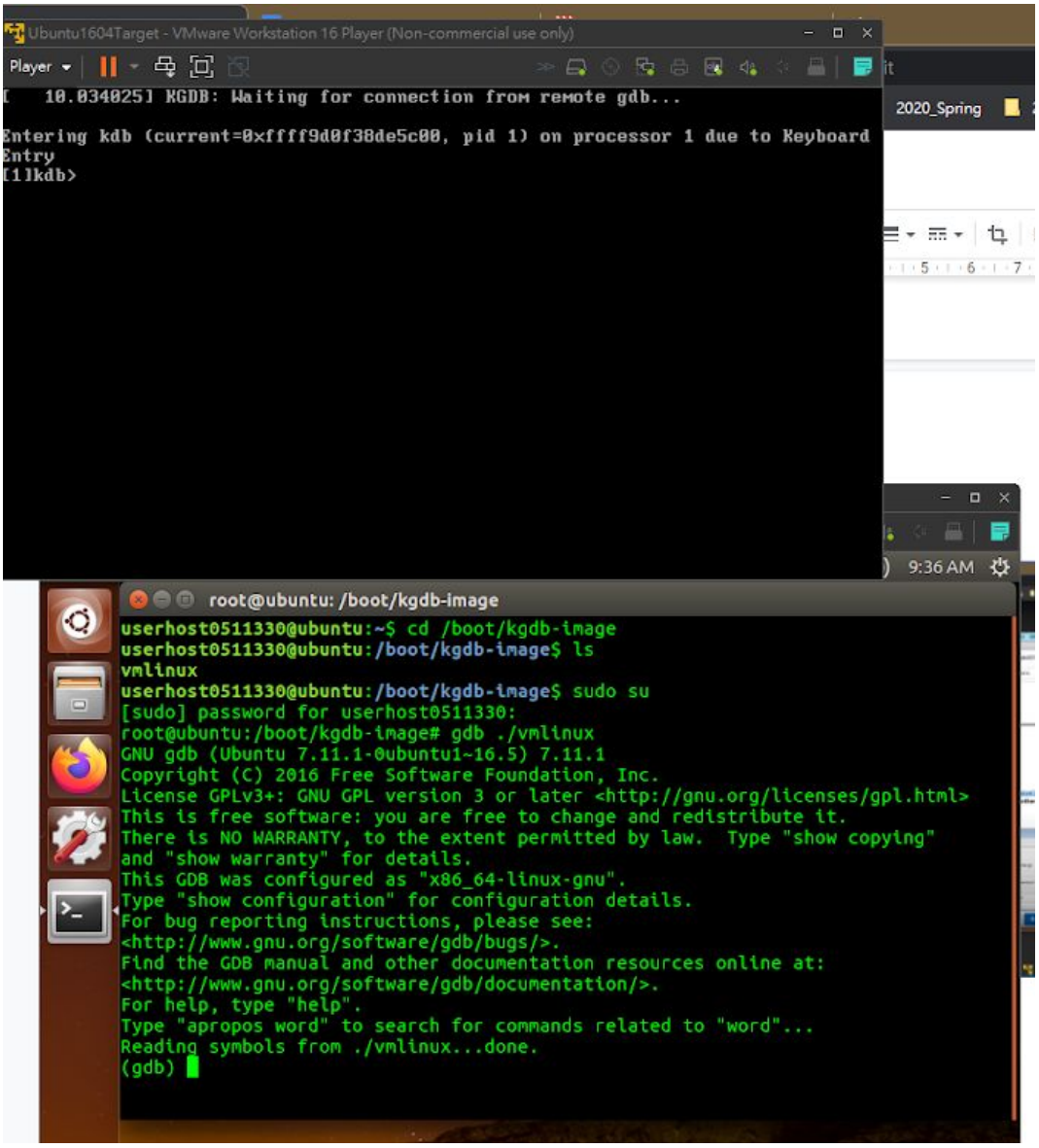
Note that we can use the **$ifconfig** to find the Target IP : 192.168.83.129



[Section 4] Kernel Debugging

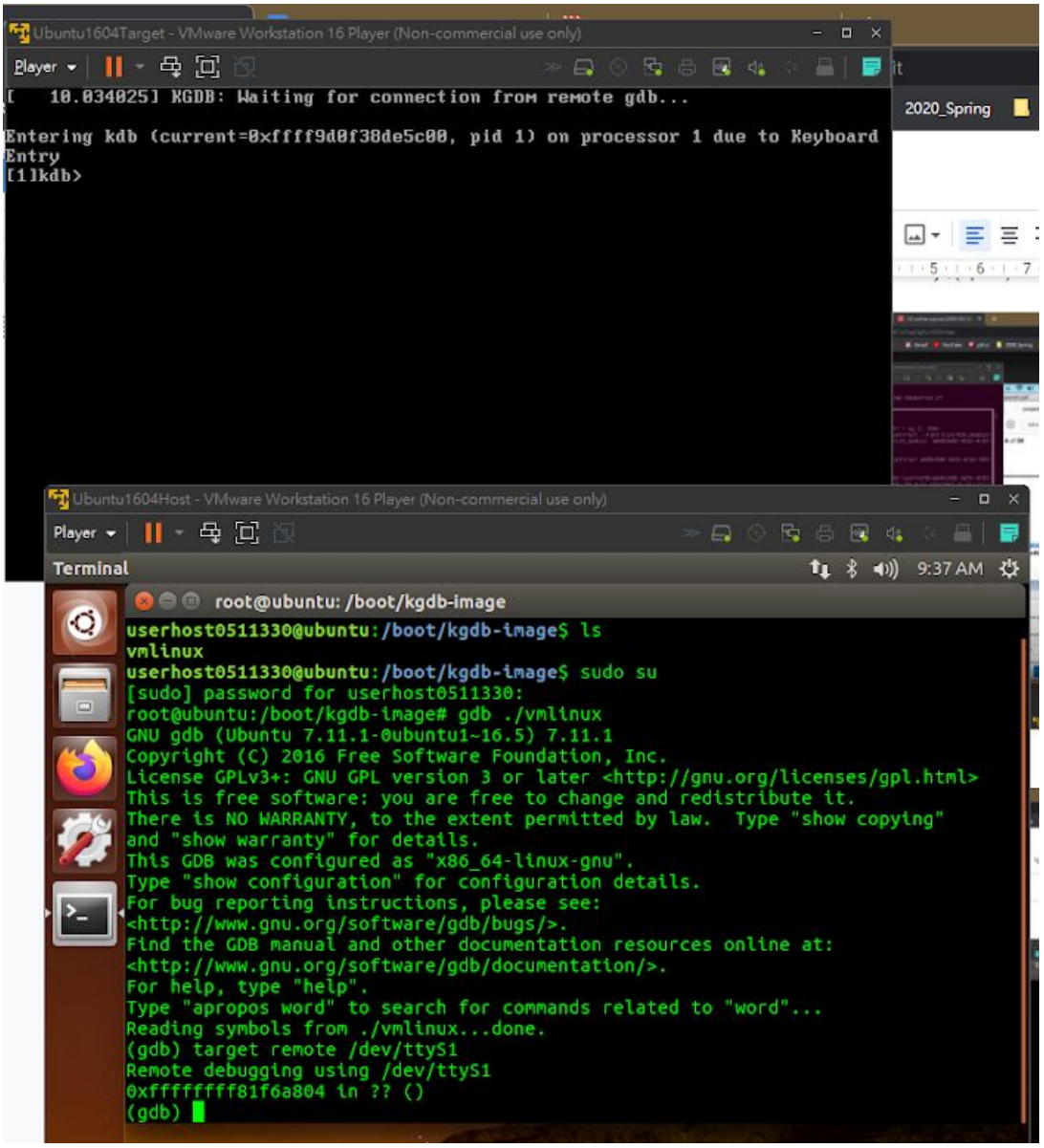
Screenshot 13

As shown in the figure, the target is waiting for connection from remote gdb.
It means that the commands **kgdbwait & kgdboc=ttyS1,115200** works.
Now we need the highest authority of the Host machine, use the command **$sudo su** to access root authority.



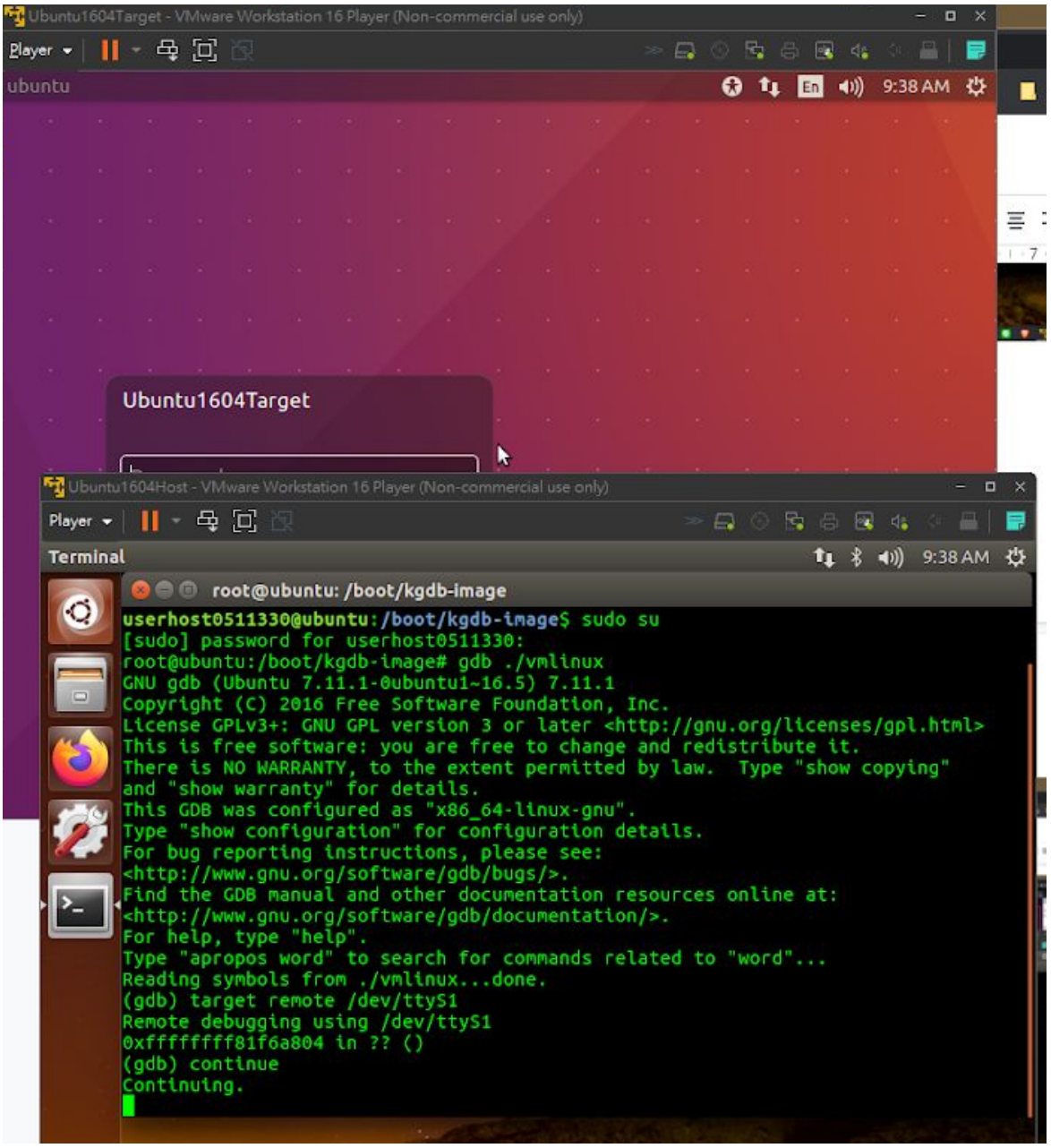
Screenshot 14

Then we runs the command **# gdb ./vmlinux** and **(gdb) target remote /dev/ttyS1.** Finally, when **(gdb)** shows on the terminal , it means that we will be able to remotely debug the patched kernel on the target machine.



Screenshot 15 -

When kernel debugging is done, runs the command **(gdb) continue**. The target will be able to boot into the OS.



Q&A

1. What is a Kernel? What are the differences between mainline, stable and longterm? What is a Kernel panic?

mainline: 由Linus Torvalds親自制作的kernel版本 (Mainline tree is maintained by Linus Torvalds.)

stable: Stable version. After each mainline kernel is released, it is considered "stable."

longterm: Long term support version. There are usually several "longterm maintenance" kernel releases provided for the purposes of backporting bugfixes for older kernel trees. Only important bugfixes are applied to such kernels and they don't usually see very frequent releases, especially for older trees.

kernel panic: Kernel panic occurs when there's some fatal error founded by Operating System. The machine will freeze and we need to reboot it.

2. What are the differences between building, debugging and profiling?

debugging: debugging is the process of running a debug session in a local development environment attached to a remotely deployed application.

profiling: profiling is the process of measuring an application or system by running an analysis tool called a profiler.

building: compile the source code.

3. What are GCC, GDB, and KGDB, and what they are used for?

(As mentioned in the previous page.)

GCC: **G**NU **C**ompiler **C**ollection, a compiler used for compiling C Language.

GDB: **G**NU **D**e**b**ugger, a debugger used for software development by high-level Language.

KGDB: **K**ernel **G**NU **D**ebugger, a debugger used when developing the kernel.

4. What are the /usr/, /boot/, /home/, /boot/grub folders for?

/usr：存放系統資訊和目錄，目錄用來存放程式與指令。

/boot/：你的開機檔所在位置，這裡就是放置你 Linux 核心與開機相關檔案的地方。

/boot/grub : grub bootloader 的程式碼也放在開機檔所在位置，當你開機時grub會引導你做開機後選擇。

/home：系統預設的home directory，新版本的Linux中，http和ftp等程式的home directory也會存放在此。

5. What are the general steps to debug a Linux Kernel?

1. download the source code
2. prebuilt the config file
3. patch the kernel and build it.
4. Setup grub and kgdb settings. (vmlinux files)
5. used kgdb to remotely debug the kernel.

6.For this project, why do we need two virtual machines?

Since each time we recompile the kernel take us a long time. Besides, after the compilation, the system may freeze due to others bugs.
So we use the kgdb to remotely debug the specific partition of the kernel code, then it will be more efficient when developing the kernel.

7. In Section 3.3, what are the differences between make, make modules_install and make install?

make: the command is used to compile the kernel.

make_modules_install: the command is used to download the kernel's modules.

make install: install the kernel which has been built int /.vmlinux file.

8. In Section 3.4, what are the commands kgdbwait and kgdboc=ttyS1,115200 for?

kgdbwait: grub will told the target machine wait for gdb connection instead of booting into the system

kgdboc=ttyS1,115200: kgdb **o**ver **c**onsole, we choose the second serial port (**ttyS1**) as console, and the baud rate is set to **115200** (bits per second).

9. What is grub? What is grub.cfg?

(As mentioned in the first page)

GNU Grub是一個Opensouce的專案，是一種bootloader，讓電腦可以安裝多種作業系統，幫助引導到使用者到欲使用的作業系統上。
Grub.cfg是Grub config files，能設定開機後Grub需要的參數，例如kgdbwait就是一種參數，能讓電腦等待kgdb連線。

10. List at least 10 commands you can use with GDB

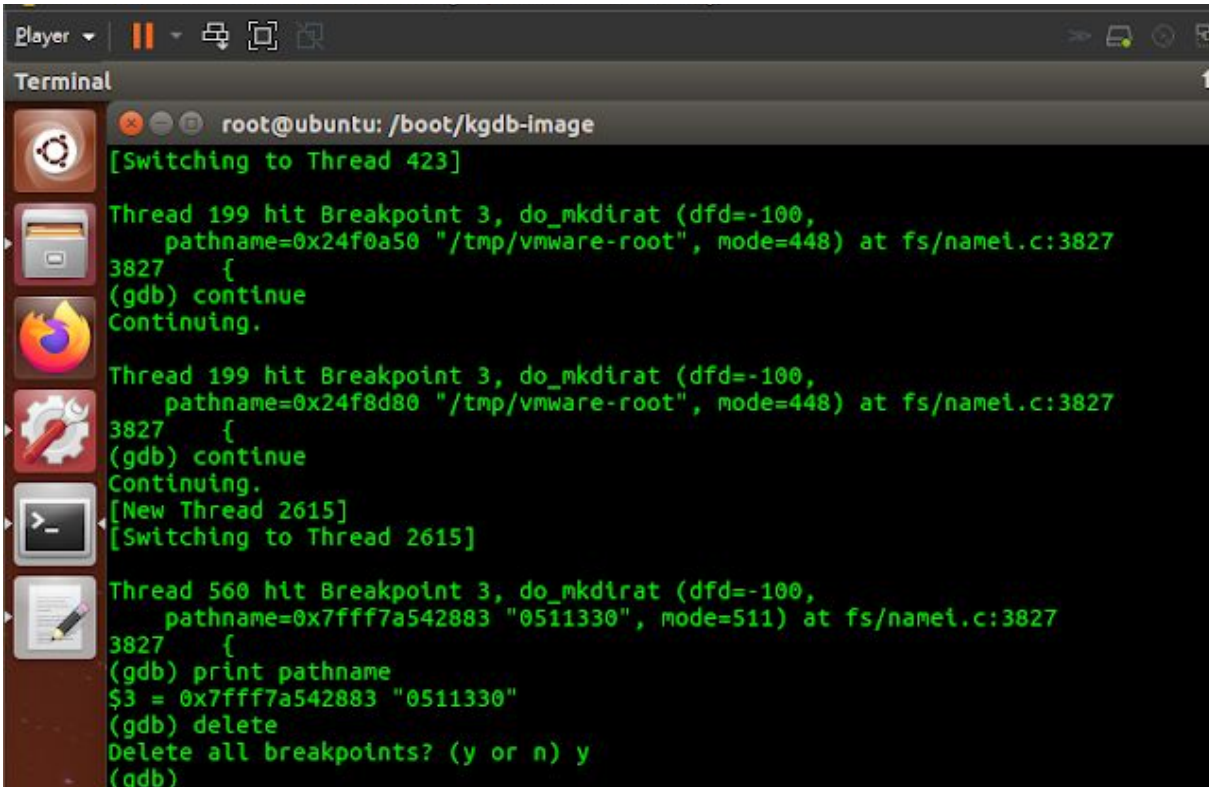
print:	印出變數內容。
whatis:	印出變數的型態。
continue:	繼續執行。和 breakpoint 搭配使用。
breakpoint:	設定中斷點。
clear/delete:	刪除某個/全部的 breakpoint。
next:	單步執行。
kill:	中止程式的執行。
list:	印出程式碼。
quit:	離開 gdb。
backtrace:	堆疊追蹤。
run:	執行程式，或是從頭再執行程式。

gdb commads sreenshot

1.Delete breakpoint

(gdb) delete

delete all breakpoints



2. Set breakpoint

(gdb) print [variable name]

e.g. Print out the function's (syscall: mkdir) parameter "pathname", we got the name of directory "0511330"