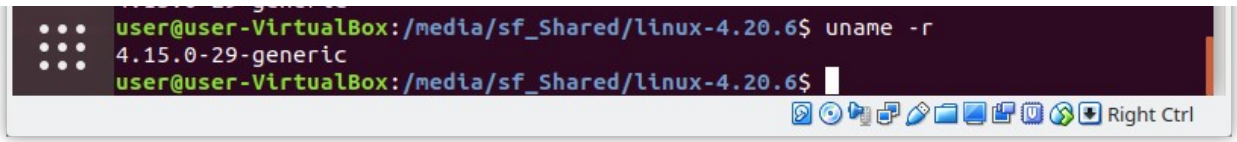
Assignment 1

OS : Kubuntu
Downloaded the linux kernal



Version 4.20.6 was the stable version was chosen.

Command
wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.20.6.tar.xz
unxz -v linux-4.20.6.tar.xz

To gernerate the config file.

```
make defconfig

make -j $(nproc)
```

Question 2
What is built-in?
When the kernal is booted up the kernal automatically inserts these drivers into the kernal. This makes the kernal size larger. The module will be loaded inside the memory whenever the kernal is loaded. And the beneifts are the all these module are not fragmented. We should only use built-in on the modules that are nessecary for the user.

What is excluded?
A exculded module is not added to the kernal at compile time.

What is module?
These modules are loaded at runtime. These are also known as Loadable Kernal Module.

In practice.
This config means.
Driver_A=y
The driver_A is a built-in module. Hence it will be loaded during the kernal bootup.

While, This config means.
Driver_B=m
The driver_b is a Loadable Kernal Module. A LKM is a module that gets loaded dynamically.

To view a list of the LKM Modules we can use the command.

lsmod

Module	Size	Used by
ccm	20480	6
rfcomm	77824	16
pci_stub	16384	1

.

.
.

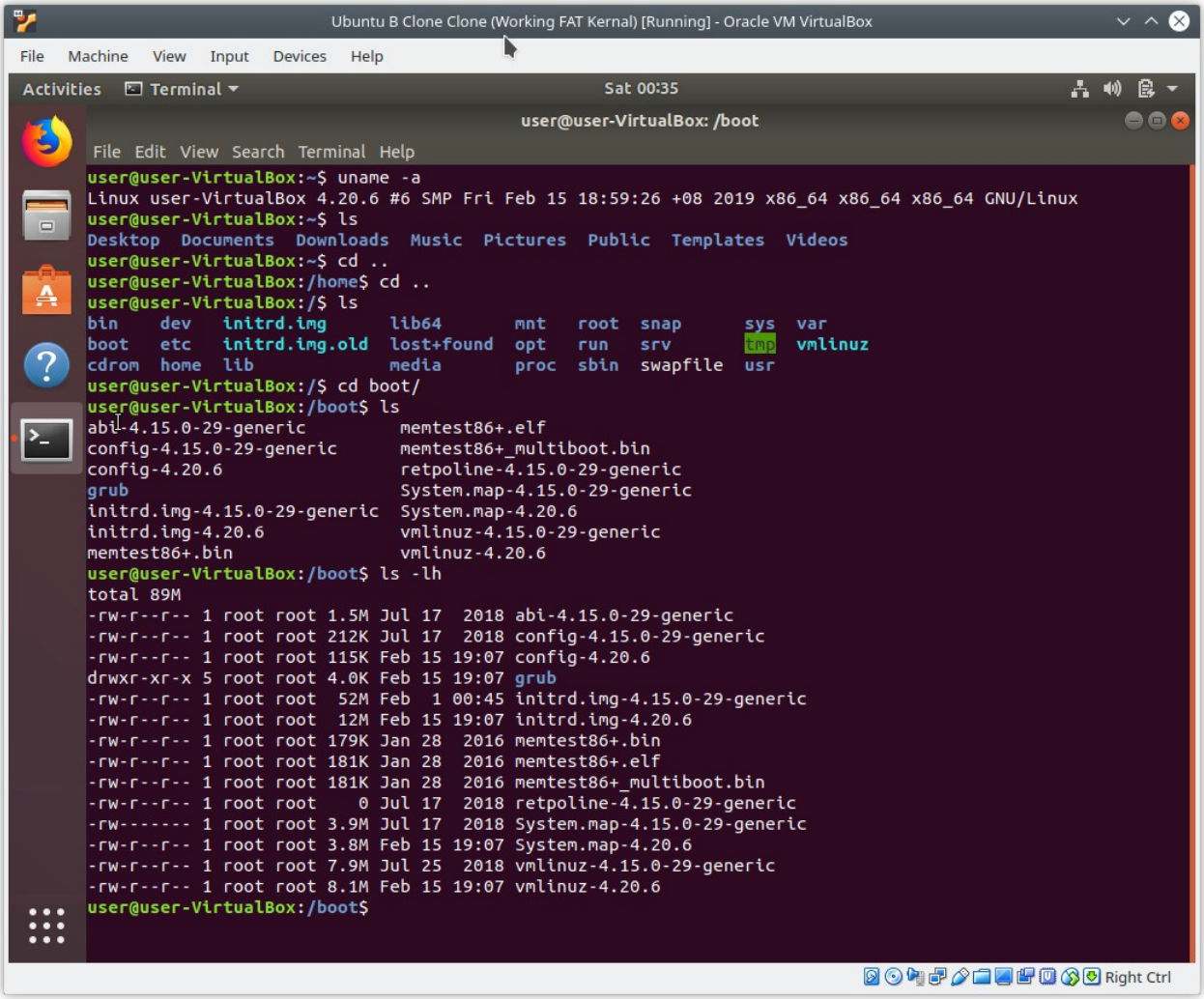
Since LKM are loaded inside the kernal process. This is a potential security issue. As if a user might load a malicaious LKM.

Which are the ones that will appear in the kernal image?
Only the built-in drivers appear in the kernal. And the module drivers will be under the folder */lib/modules/ or /lib/modules/[Kernal Version]*

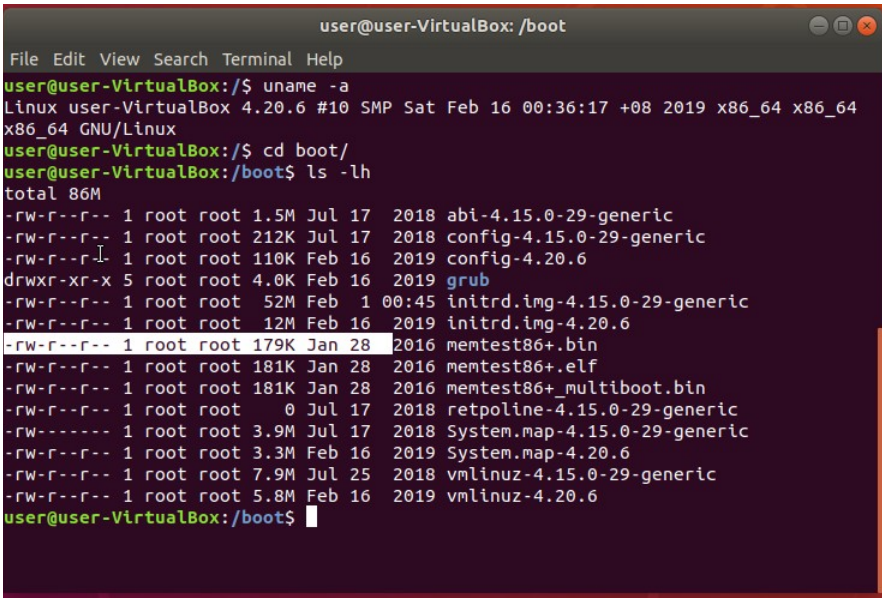
What is *make defconfig*?
Defconfig use the basic set of needed driver needed to run the server.It uses the x86_64_defconfig.

Problems encountered
The complie time is long, to speed it up. To make the complie multi-threaded.
We use the command *make -j 4* to speed up the complie time.

Size of the kernal when using defconfig.
System is 8261 kB



Size of the kernal after removing.
Change zip format.
Change zip to XZ
Optimize for size.
Remove Kernal Hacking mods
System is 5829 kB



Part B
81 46 f8 83 00 00 00

OpCode	MOD	REG	RM	Displacement	Constant
81	46			f8	83
1000 0001	01	000	110	1111 1000	1000 0011

If opcode high-order bit set to 1, then instruction has an immediate constant.
If opcode lowest-order bit set to 1, then 8-bit operands

REG is 000 is eax

MOD is 01 means One-byte signed displacement follows addressing mode byte(s).
MOD R/M
01 110 = [esi + disp8]

The code will looks like.
add eax, [esi + disp8]

Adding the displacement
add eax, [esi + f8]

Include the constant
add 83, [esi + f8]

What is the 32-bit instruction encoding for the following instruction:

addl $13, %edx

Size of the operands are 32 bits

OpCode	MOD	REG	R/M	Constant
0	8			13
0000 0000	00	010	000	0001 0011

What would be the 64-bit encoding for the same instruction above?

addl $13, %rdx

REX Prefix	OpCode	MOD	REG	R/M	Constant
	0	8			13
01001000	0000 0000	00	010	000	0001 0011

Refence websites
https://wiki.osdev.org/X86-64_Instruction_Encoding
https://www-user.tu-chemnitz.de/~heha/viewchm.php/hs/x86.chm/x86.htm