README

1. Make a working directory.
    1. "mkdir task" to make a new directory.
    2. "cd task" to start working in the directory.
2. Create the module.
    1. copy the source code for the module.
    2. "vim <filename>.c"
    3. click "i" to insert in the file, paste the code in there
    4. type ":wq" to write and quit the file
3. Create the Makefile.
    1. copy the Makefile provided.
    2. "vim Makefile"
    3. click "i" to insert in the file, paste the content in there
    4. type ":wq" to write and quit the file
4. Install required packages.
    1. "sudo apt install build-essential" will download all the essential packages such as gcc and make.
5. Test the module.
    1. run "make" command in the terminal. After that, you should see a file "<filename>.ko" in the current directory.
    2. "sudo insmod <filename>.ko" to insert the module into kernel.
    3. The message is printed to the kernel log, so you can run "sudo dmesg" to see the message. There should be a "Hello World" message with a timestamp in front right now.
    4. "sudo remmod <filename>.ko" can remove the module from kernel.
    5. Now, if you run "sudo dmesg", there should be a "Goodbye!" message that have a timestamp after the message "Hello World".

6. I have put the commands into the Makefile. To "automate" the process, type "make test" in the terminal. This will compile the module code and run the command under "test" section in the Makefile. The output should be something like

```
[  155.626681] Hello World
[  155.700225] Goodbye!
```

Things that I struggled with

- I tried to do it on attu, it did not have the qemu package installed and I do not seem to have access to download that package. I dug a bit further and found that it might be better for me to use VMware for a linux environment. However, I have mac with silicon chip and I do not think I can have VMWare set up yet. Eventually, I borrowed someone's linux computer to do the task.