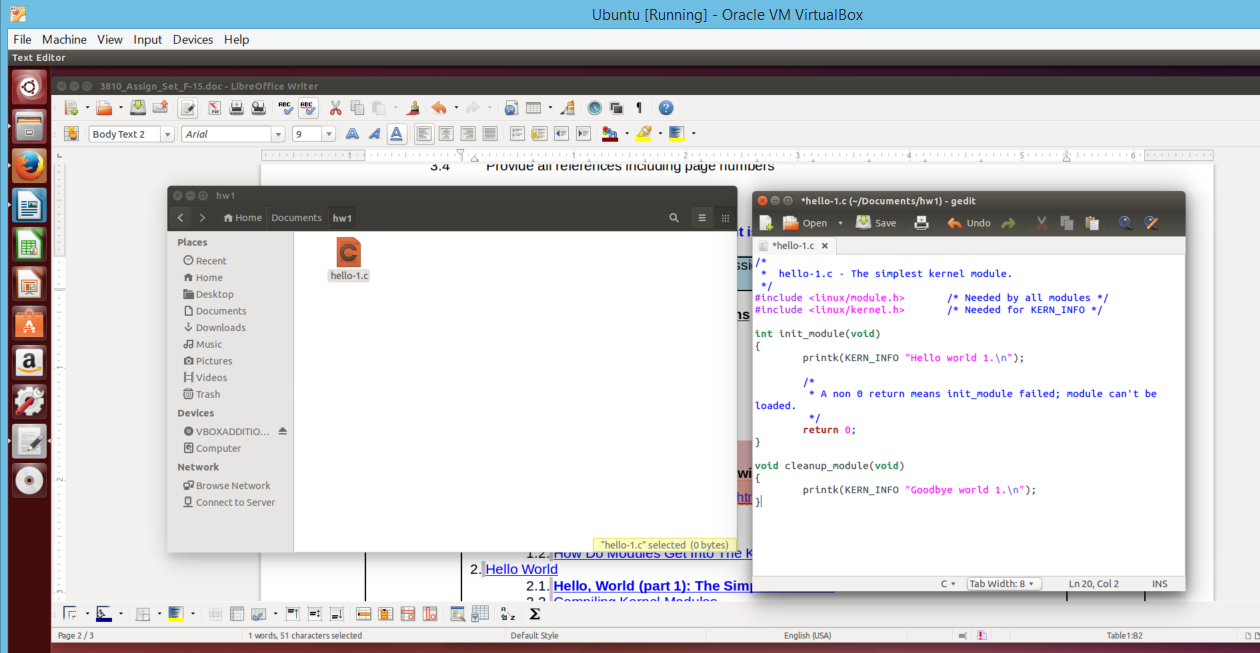
**Student name:** Federico Watkins

**CSIS 3810:** Operating Systems

**Chapter:** 1, 2

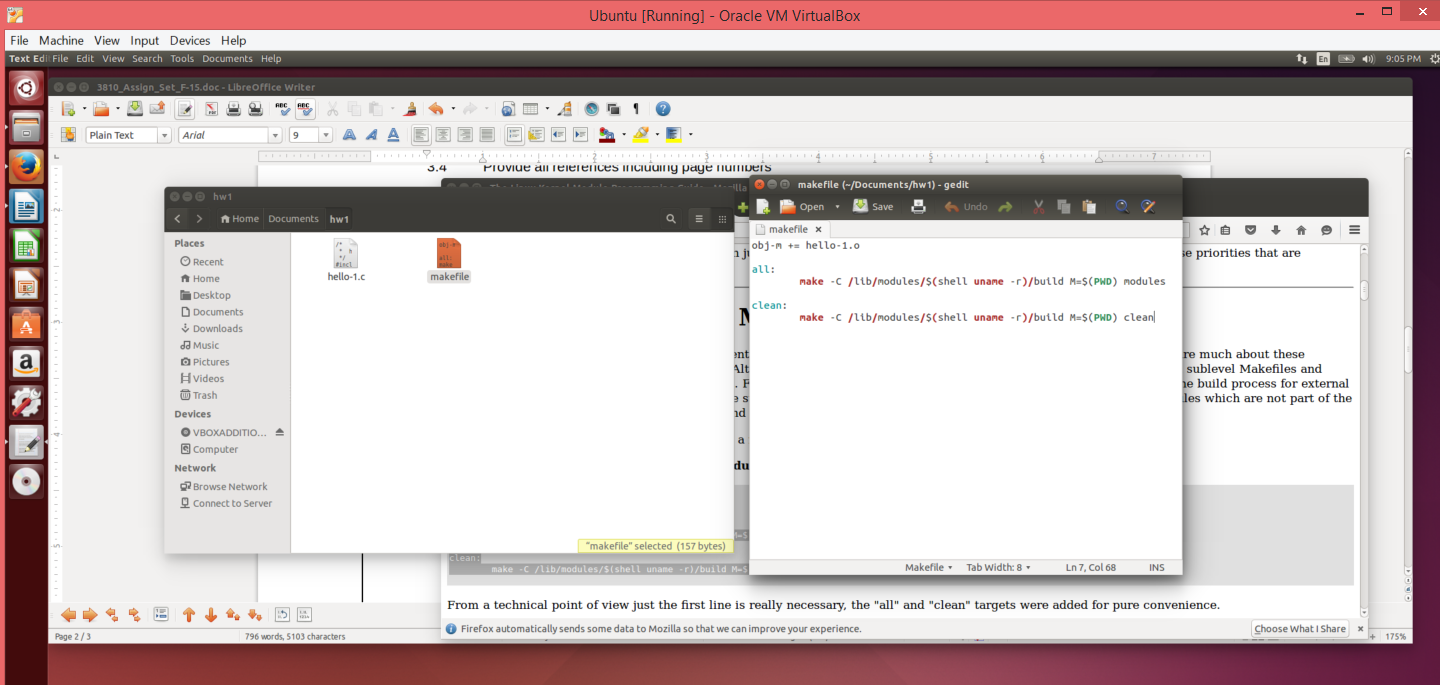
**Assignment:** 1

1. **Create a  C code into your directory  and show the screen having this file:**



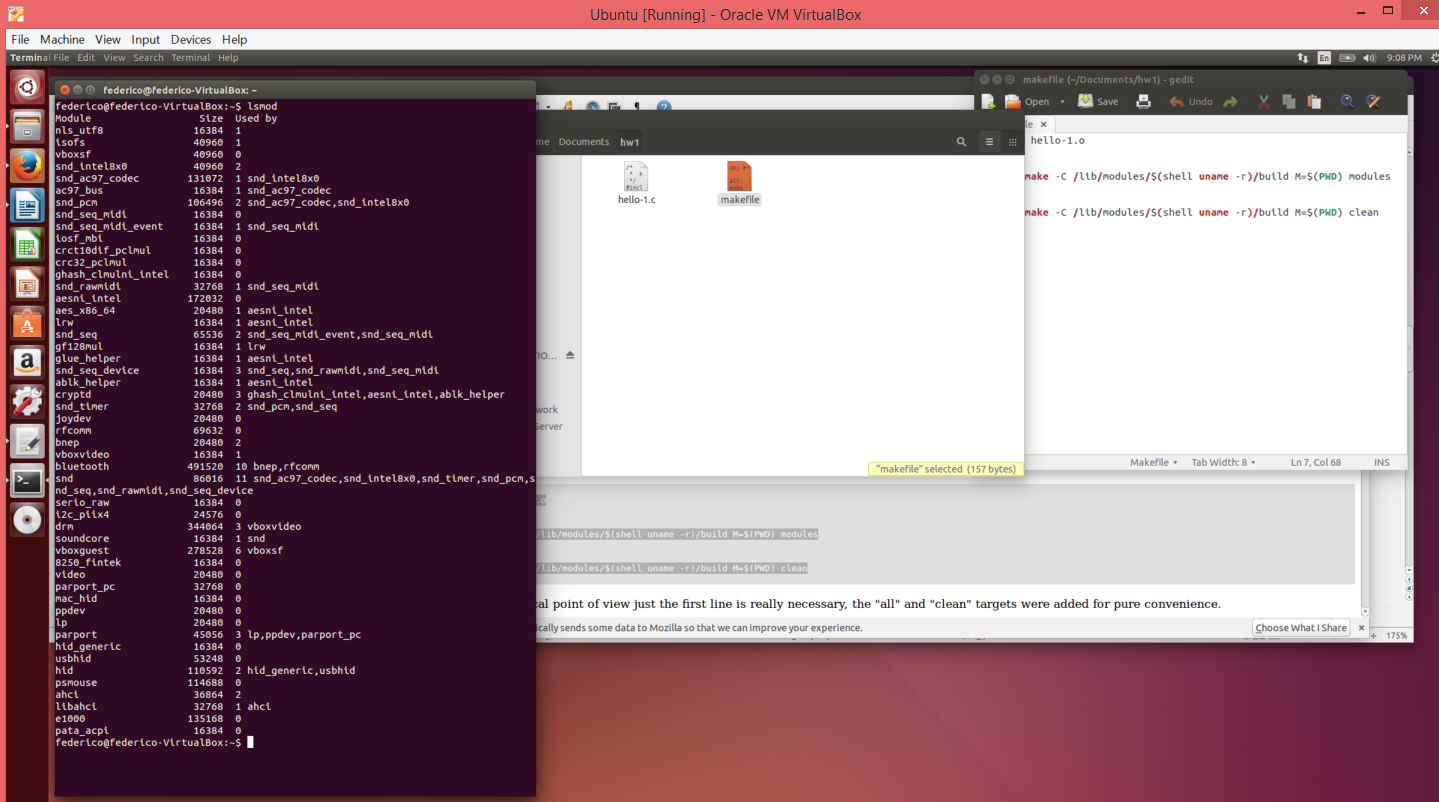
At this point I create a text file with C code and hence extention “.c”, this file holds my C program “hello-1.c” which is set to output “Hello world 1” once inserted and “Goodbye world 1” once removed.

1. **Create Make file and show it:**



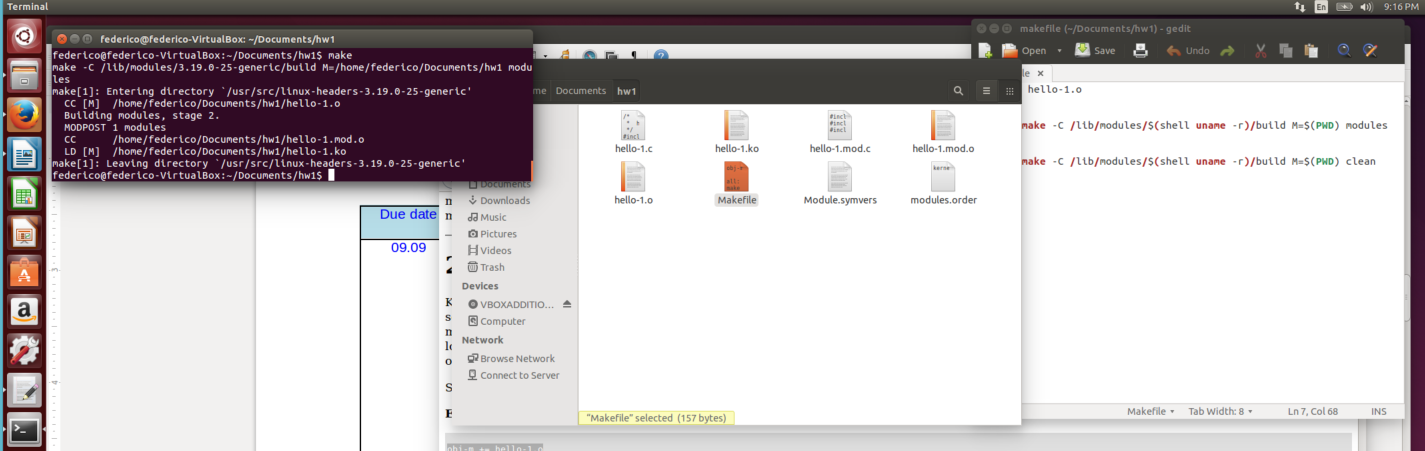
At this point I create a file called Makefile which will be used by the make command to compile my module.

1. **Show  what modules you had in the kernel before the installation:**

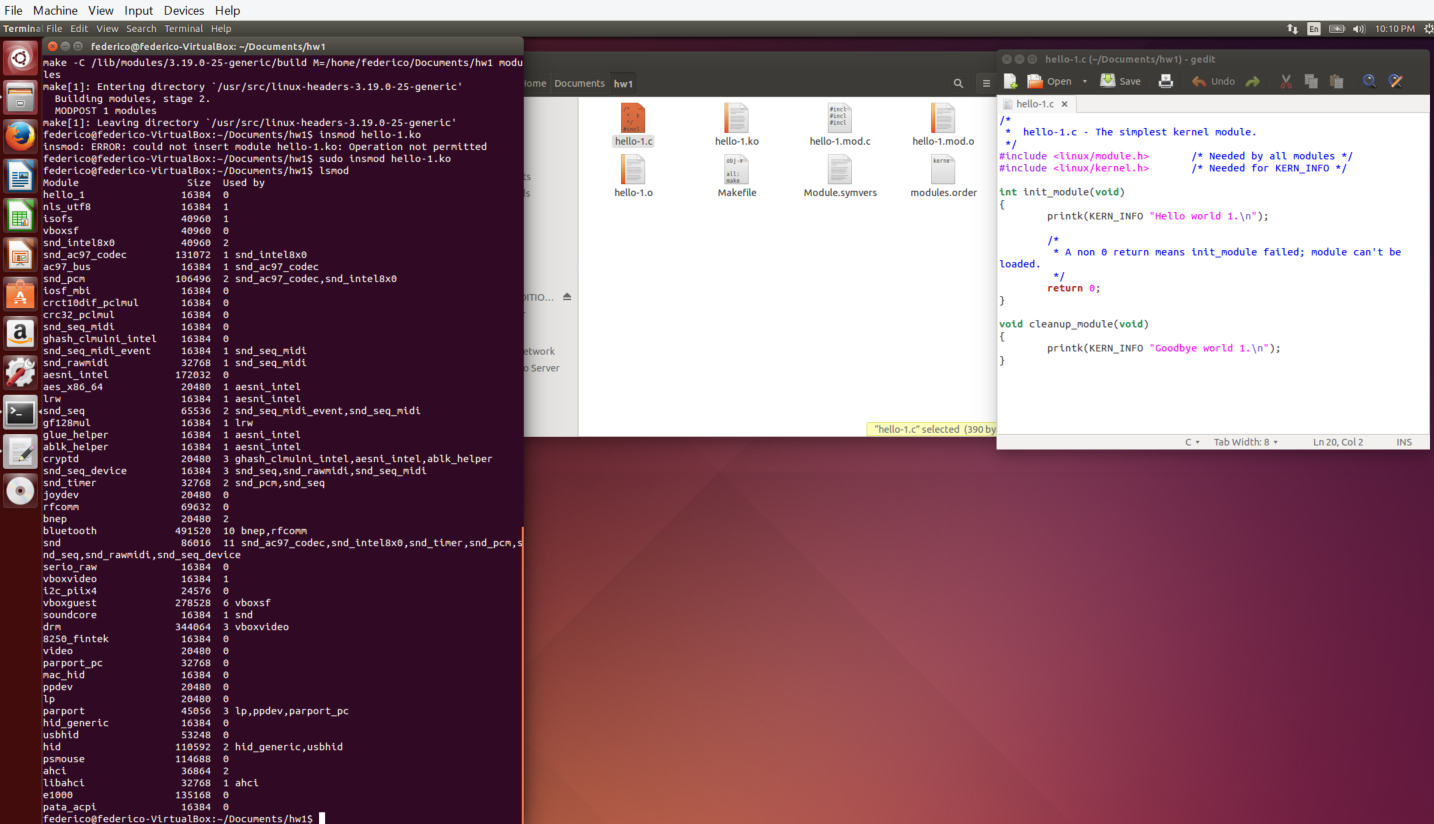


At this point I show that my kernel doesn’t have a module called “hello\_1” by using the “lsmod” command.

1. **Install the module into the kernel  and show that it is there:**

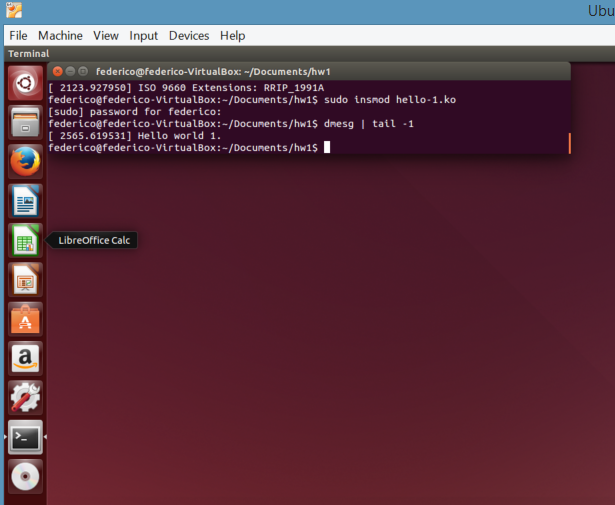


At this point I run the make command within my working folder to look for my “Makefile” file and compile my module.



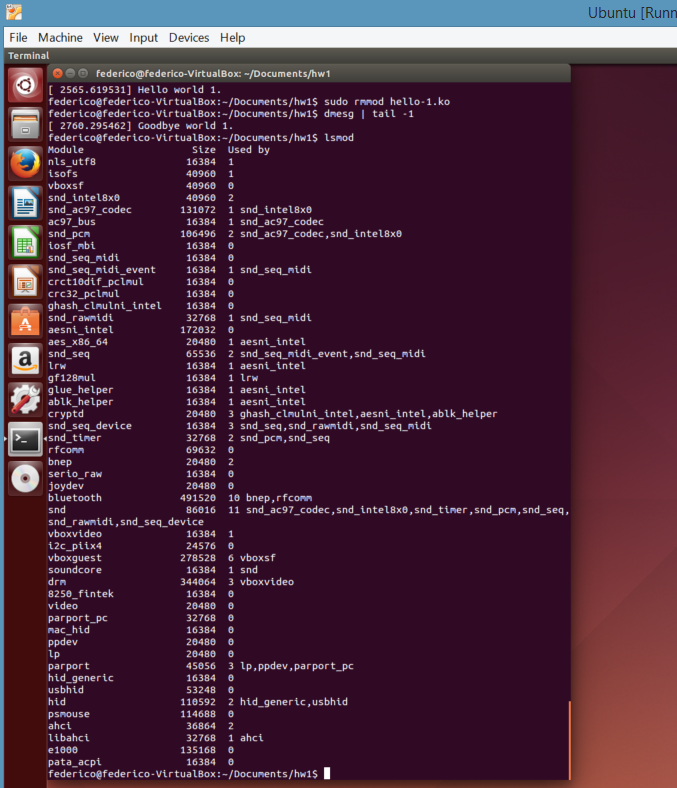
At this point I use “sudo” to exercise my administrative right and “insmod” to insert my module, then “lsmod” again to show the my module is now loaded into the kernel.

1. **Display the text ( directly or find it in sys\log) :**



At this point I use “dmesg | tail -1” to display the message in the system log showing the output of my module.

1. **Uninstall/clean the kernel and show that your module is not there anymore; support it by showing the corresponding text:**



At this point I use “rmmod” to remove my module from the kernel, then I use “dmesg | tail -1” again to display the message in the system log indicating that my module has been removed. Lastly, I use “lsmod” again to show that my module is no longer in the kernel

7.  **At the end (or in the beginning), tell me how this program relates to the material of your chapter:**

This program exposes many of the material in our Chapter 1 and 2. It shows the importance of understanding Computer-System Organization, Architecture, Structure, and Operations amongst other things. From Kernel Data Structures to Computing Environments, to User and Operating System and Operating-System Debugging as we encountered errors. In one way or another this program related directly to the chapters material. I was forced to learn a great deal of information outside the scope of the program because I made many mistakes that required deeper research to resolve.

**---Things learned in Chapter 1:**

In chapter 1 I learned about the basic organization of computer systems. We looked at storage-devise hierarchy of magnetic tapes, optical disk, magnetic disk, solid-state disk, memory, cache and register. Overview of how the CPU, disk controller, USB controller, graphics adapters access a shared memory. I also learned about the major components of operating systems. The relation between computer hardware, operating systems, system and application programs like compilers assemblers, text editors, database systems, etc. Another subject covered was the many types of computing environments. Interesting discussions about kernel data structures, protection and security, storage management, process management, memory management. I was also surprised to learn about several open-source operating systems, computer system architecture and its structure.

**---Things learned in Chapter 2:**

In chapter 2 I learned to describe the services an operating system provides to users, processes and other systems. We went in detail through error detection, protection and security accounting resource allocation, communication, file systems, I/O operations, program execution and how they relate to system calls and user interfaces such as the GUI, batch and command line. I also enjoyed the discussion on the various ways of structuring an operating system. We covered MS-DOS layer structure, traditional UNIX system structure, Mac OS X structure, Apple iOS, Google’s Android and the layered operating system approach. After reading this chapter I am better prepared to explain how operating systems are installed and customized and how they boot.