



Part 1

Implementation of Simple “hello world” Module and Run

Every module structure have two major functions:

1. init
 - executed when the module is loaded into kernel
2. cleanup
 - executed when the module is removed from kernel

the simple module implementation is here:

A screenshot of a code editor window showing the implementation of a simple kernel module. The code is as follows:

```
#include <linux/module.h>
#include <linux/kernel.h>

int init_module(void){
    printk(KERN_ALERT "hello world!\n");
    return 0;
}

void cleanup_module(void){
    printk(KERN_ALERT "goodbye world!\n");
}
```

The editor interface includes a menu bar (File, Edit, View, Search, Tools, Documents, Help), a toolbar with icons for Open, Save, and other functions, and a status bar at the bottom showing 'C', 'Tab Width: 8', 'Ln 12, Col 20', and 'INS'. The system tray in the top right corner shows the time as 9:23 PM.

notes:

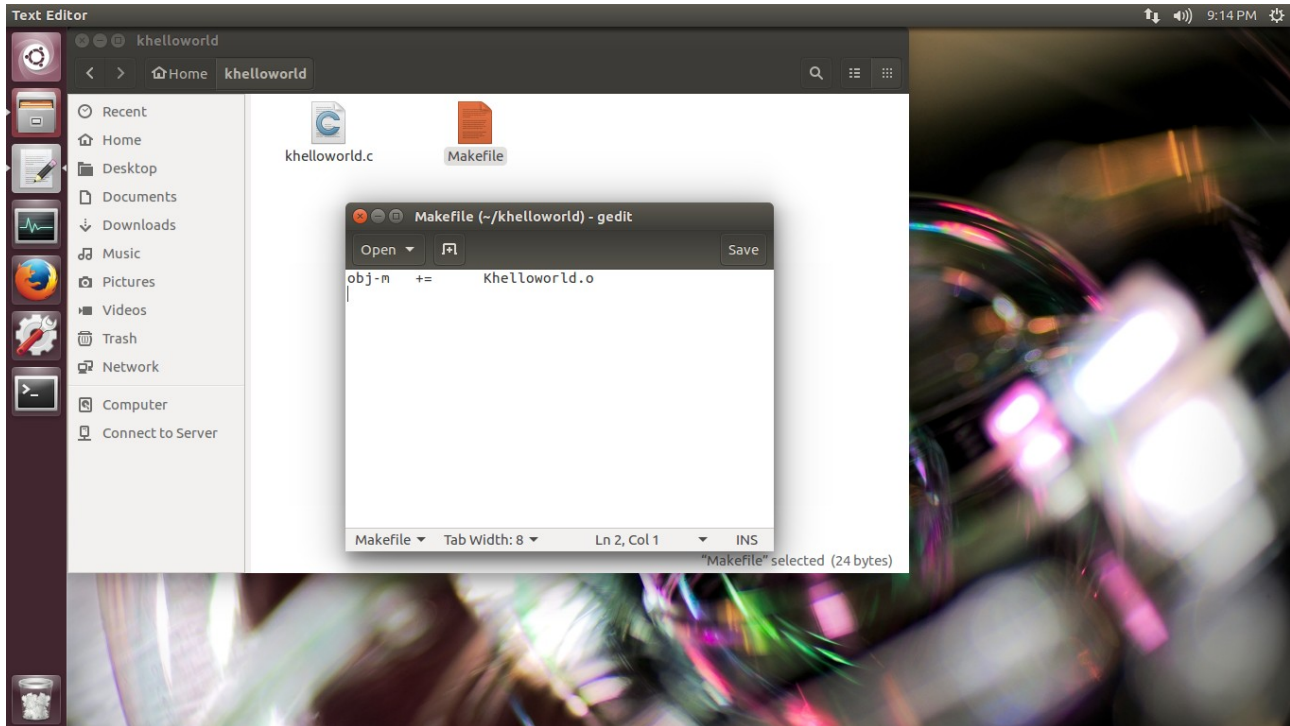
module.h is for the module itself



Operating Systems
Spring 2017

kernel.h is for the KERN_ALERT

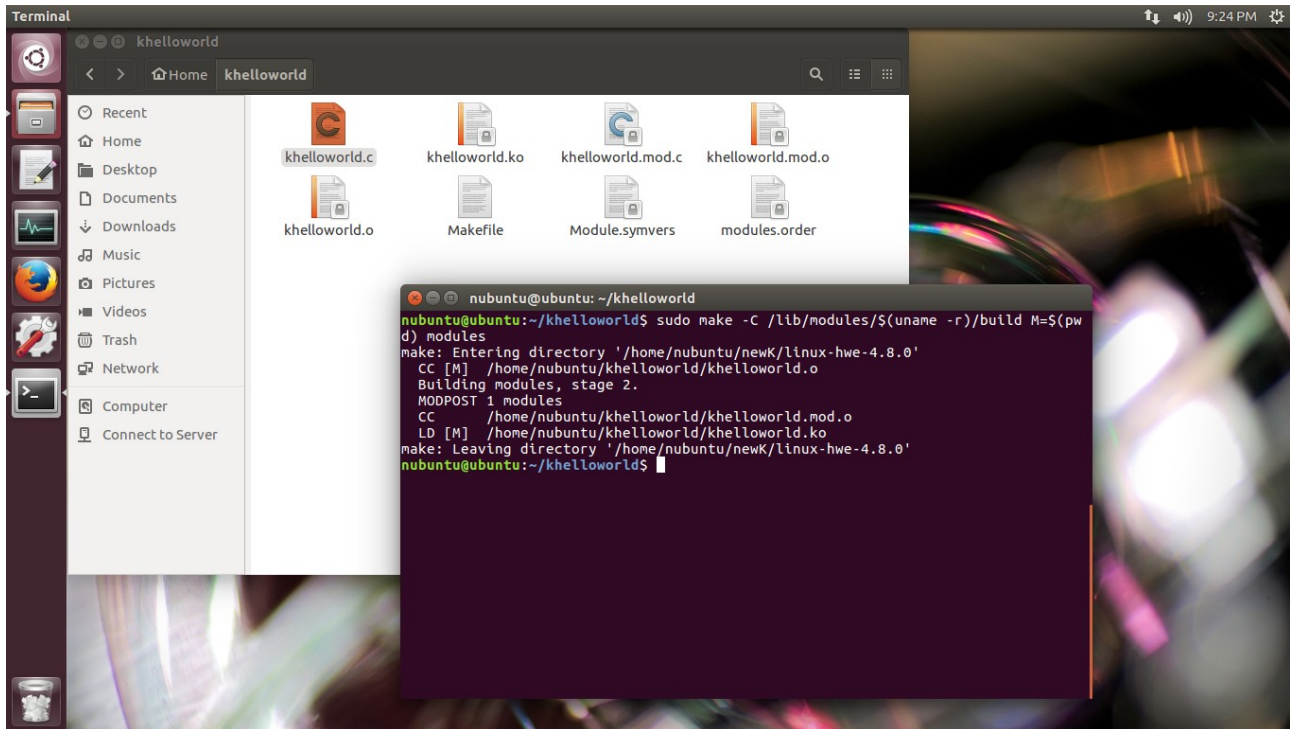
the make file for this module:



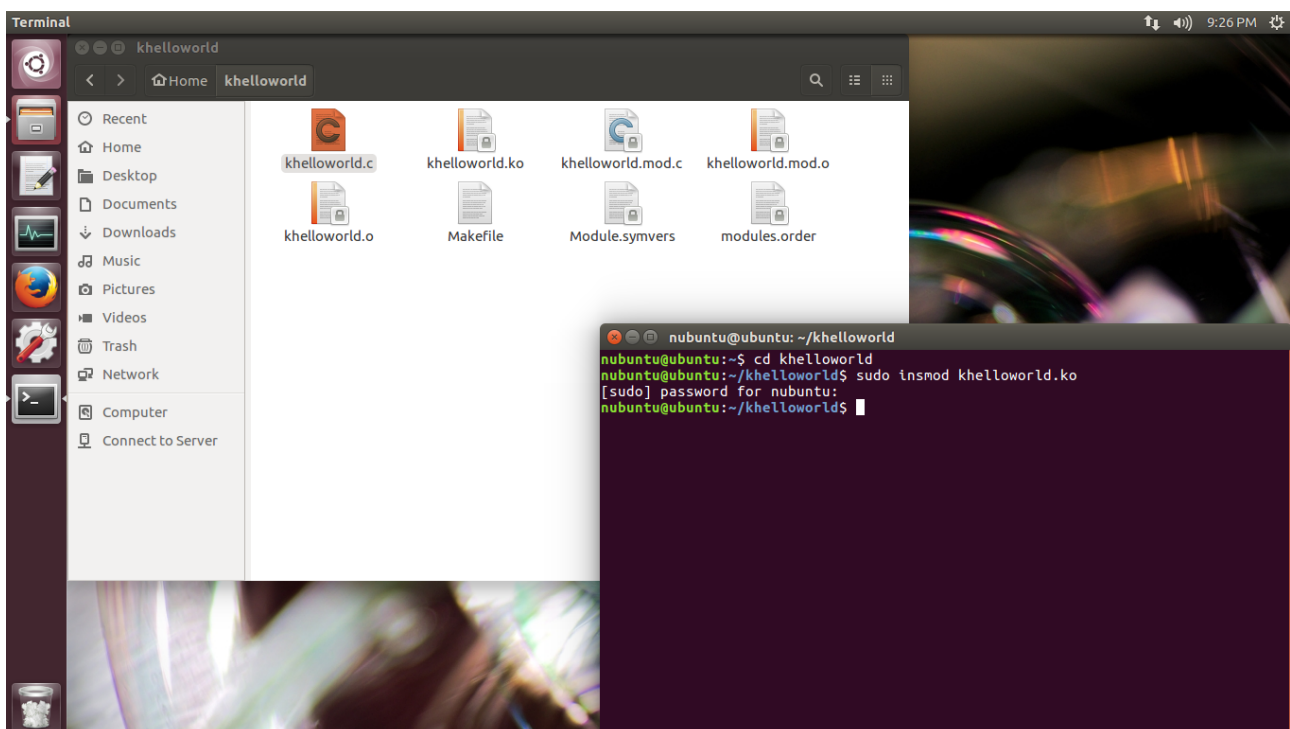
now lets compile the module:



Operating Systems Spring 2017



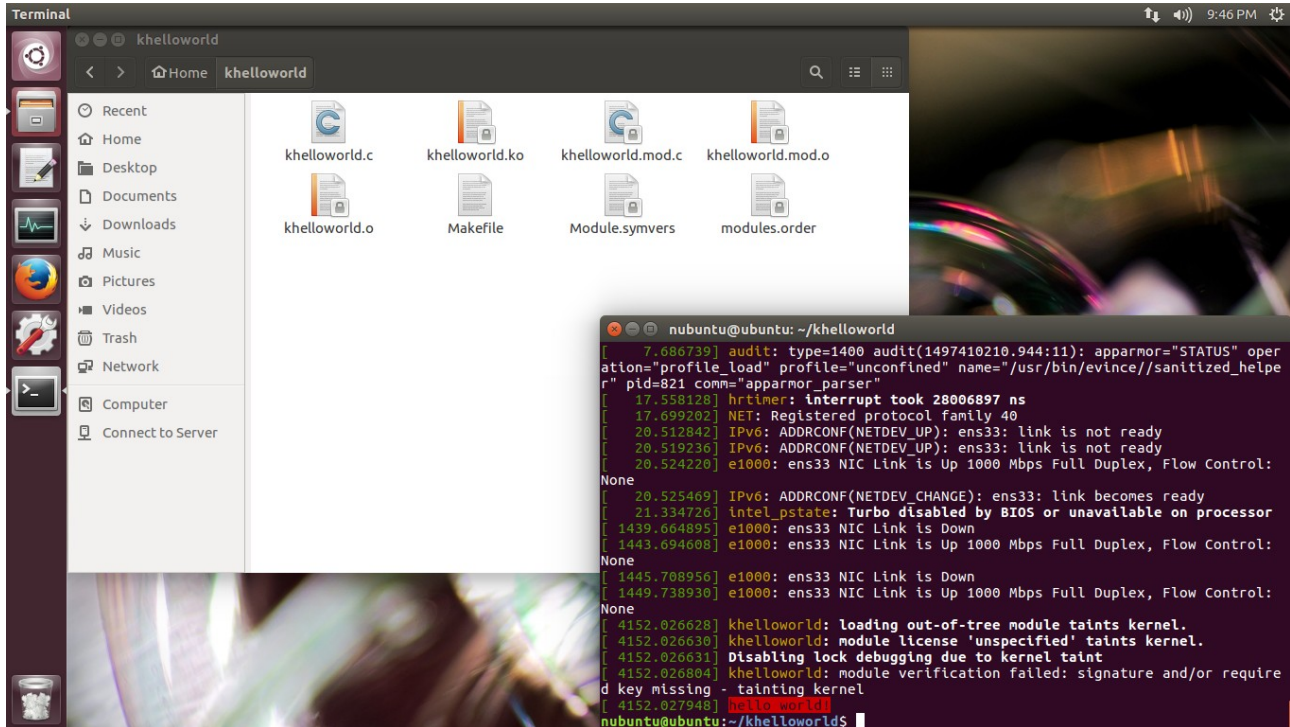
install the module:





Operating Systems Spring 2017

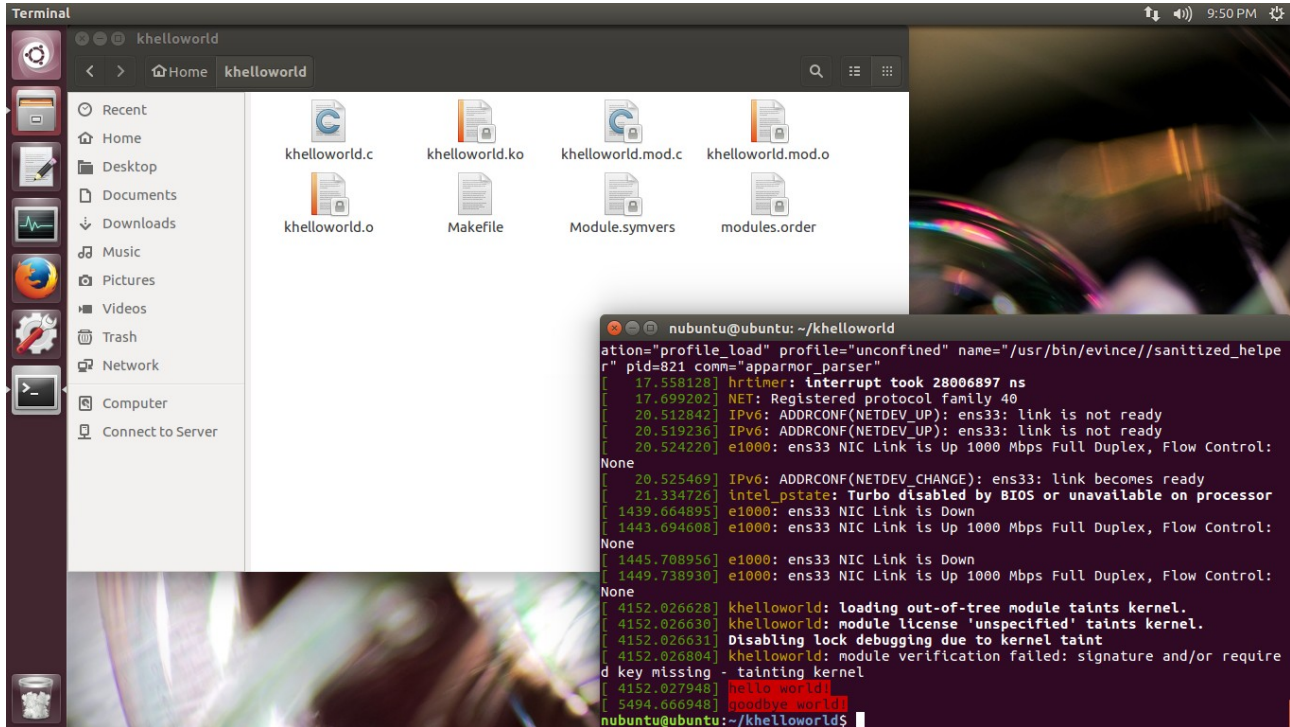
now checking the dmesg:



now you can remove the module with `sudo rmmod khelloworld.ko` and check dmesg:



Operating Systems Spring 2017



End of part one