National University
of Computer & Emerging Sciences
Peshawar Campus

Problem Set:	Assignment 08	**Semester**:	Spring 2013
Points:	4		
Date Set:	May 21, 2013	**Due Date**:	May 29, 2013
Course:	CS206 Operating Systems	**Instructor**:	Nauman

Having compiled the complete kernel, we now want to include a kernel module at runtime. Our kernel module does not do anything useful (except demonstrate the process of writing and including it). Follow the steps below to complete the code.

1. Create a new directory in your home folder by the name `kern_mod_hello` – the name isn't important. It can be anything but make it descriptive.

2. Create a file called `hello.c` and include the following code in it.

```c
#include <linux/module.h>     /* Needed by all modules */
#include <linux/kernel.h>      /* Needed for KERN_INFO */

int init_module(void)
{
        printk(KERN_INFO "Hello world 1.\n");

        /*
         * A non 0 return means init_module failed; module can't be loaded.
         */

        return 0;
}

void cleanup_module(void)
{
        printk(KERN_INFO "Goodbye world 1.\n");
}
```

3. Create another file `Makefile` and put the follosing contents in it.

```
obj-m += hello.o

all:
  make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
  make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

   (The lines starting with `make` must have a tab character before this word. This is necessary as the commands below will not work as expected otherwise.)

4. You can build the kernel module simply by issuing the `make` command in the directory where the `Makefile` is located. Makefiles are used to automate build processes. That's how the make commands in the kernel work. You are advised to read a tutorial about this on your own time.

   Hint: if the make command fails and complains about missing header files, you may need to install the `linux-headers` package. This package provides the header files required for building kernel modules.

5. Now, do a directory listing to see the newly created files. The `hello.ko` file is your new and shiny kernel module.

6. Issue the following commands and observe their output:

   (a) `lsmod | grep hello`

   (b) Now open a new terminal and issue the following command `tail -f /var/log/syslog` (You can also view the syslog using the `dmesg` command)

   (c) Leave the syslog window open and issue the following command in the first window `sudo insmod hello.ko`

   (d) Observe the syslog. Now, issue the following command in the first window: `sudo rmmod hello.ko`

(e) View the module info using: `modinfo hello.ko`

Note: For submission, please include screenshots of the complete build/run process as well as the final 'syslog' output to demonstrate the complete methodology.