Kernel and driver programming assignment – 3 -- kernel module related
Note: read this assignment, hints provided at the end of this assignment -
plan you coding and testing – proceed with coding !!!

1. use hello.c and hello1.c to generate external kernel modules – load and test them – understand their dependencies – check if they work, as per their dependency rules – use the Makefile provided with the Samples !!!

2. once the above basic testing is done, do the following :
   As per what is given in chapter 17 of LKD/3, do the following :
   a) add our module related source files to the kernel source directory
      - you must create appropriate directory with kernel source directory
      - you must create appropriate Makefile in your kernel src project directory and and edit the parent directory's Makefile, as discussed in lecture and discussed in ch17 of LKD/3
      - you must create appropriate Kconfig in your kernel source directory and edit the parent directory's Kconfig
      - verify that the appropriate menu items/options are available via "make menuconfig"
   b) configure your module as a dynamic module, compile the kernel and test it
   c) configure your module as static module, compile the kernel and test it.

Kernel and driver programming assignment – 3 – kernel modules related

Hints for this assignment :

a) read chapter 2 of LDD/ 3

b) read chapter 17 of LKD / 3

c) read ch4 of embedded linux primer – read section 4.3 only – other sections may not make sense currently – many of the points given here will be Useful, when you understand Embedded Linux kernel architecture !!!

d) read chapter 8 of Embedded Linux primer – section 8.1.4 may be relevant your work

e) in addition, do not use their example code – let us use hello.c and hello1.c modules that we have

f) do not blindly copy what is given in the references – understand – make a plan – create / modify appropriate files as needed !!!

g) build your project directory to check whether your Kconfig and Makefile are effective – refer to Advanced build options of linux kernel in a Nutshell for more hints !!!

h) you must recompile the kernel and install it for testing !!!

i) use modprobe <nameofmodule> (without .ko extension) for testing dynamic modules – refer to ch8 of Embedded Linux primer

j) use initcall_debug as a command line parameter to the kernel, for testing static modules – this will show, when your static code's init methods are Invoked, during boot-up

Kernel and driver programming assignment – 3 - kernel modules related

Hints for this assignment :

f) the process of doing this assignment can be summarized as below using a traditional kernel developers' style:

```
while(1){
        step1 : add your source file(s) to the kernel source directory;
                make changes to relevant Kconfig(s) and Makefile(s)
        step2 : use make menuconfig to check whether your changes
                are updated and visible in the kernel configuration menu
                items
        step3 : select appropriate settings for your menu-item(module or
                static)
        step4 : recompile the kernel without errors – reboot
        step5 : load the module using modprobe or check if module
                is loaded, if it is statically built into the kernel
        step6 : if above steps worked properly, break ; otherwise,
                continue
}
```