

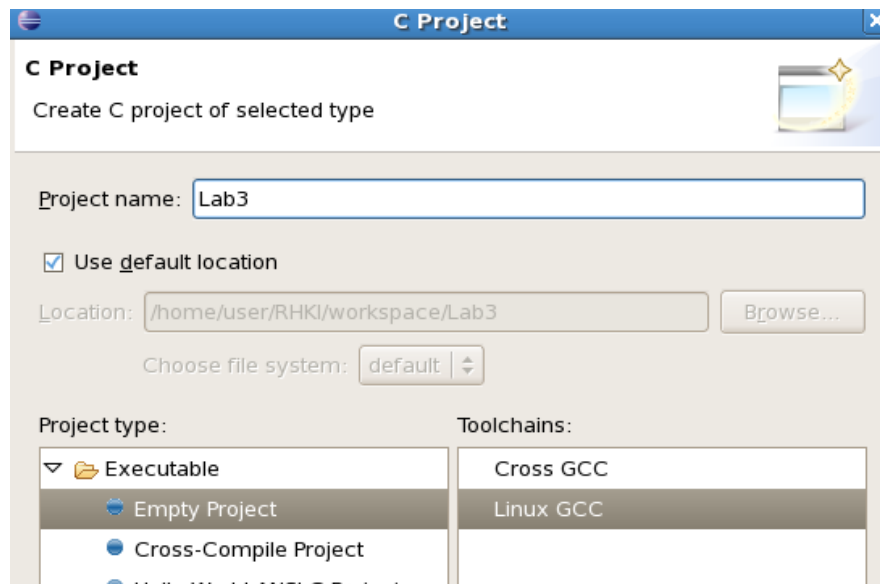
# Compiling a Linux Kernel Module

***Objective: Configure and compile a Linux Kernel module using Eclipse. For this lab we will be compiling a “hello world” Linux Kernel module to test our development environment. The code for this module will be provided.***



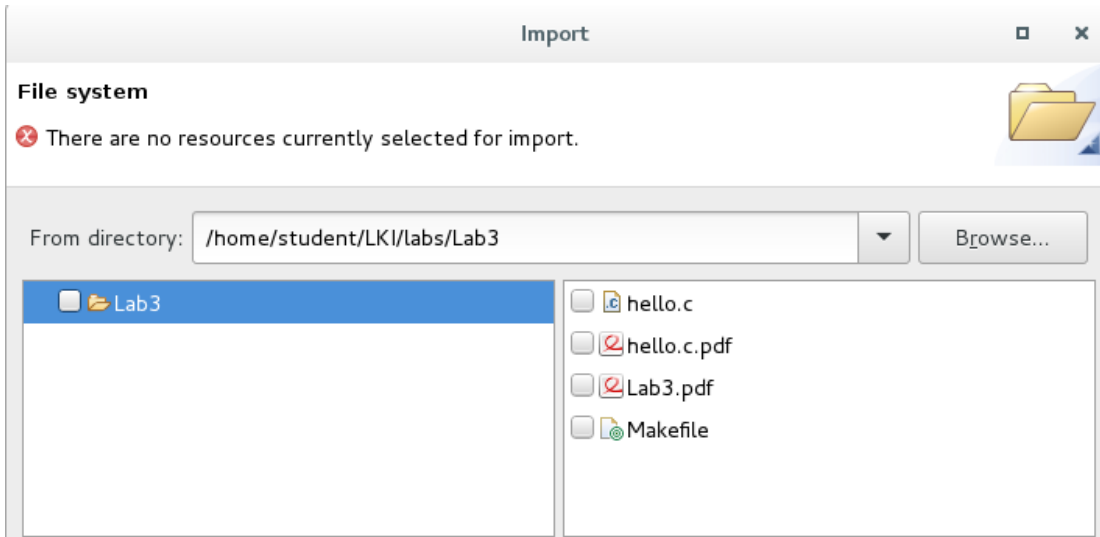
**File(s) for this lab:**

1. Start Eclipse: Applications->Programming->Eclipse.
2. In the Project Explorer, right-click and select New, C-Project. In the dialogue window call the Project Name “Lab 3” and selection Linux GCC for the toolchain:



3. Click next and keep both “Debug” and “Release” configurations. Click “Finish”.

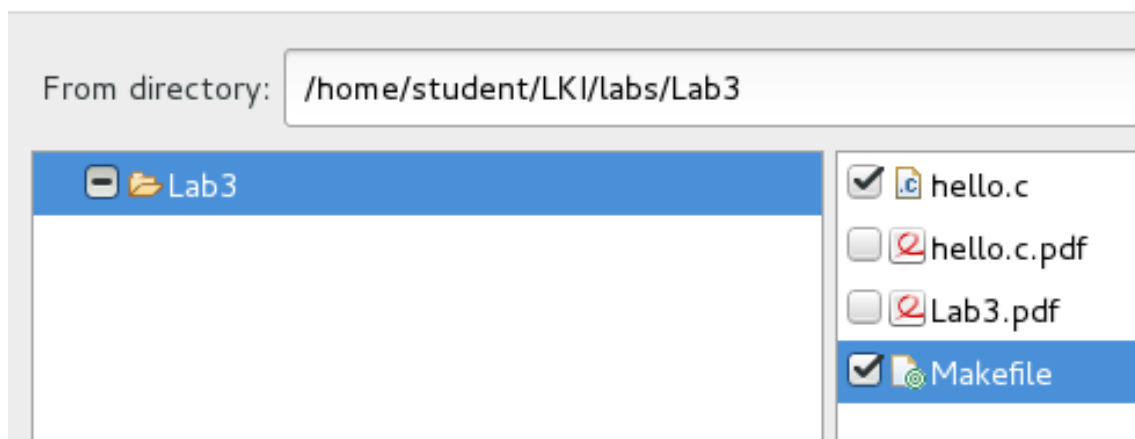
- Right click on the “Lab3” project in Project Explorer, select Import, General, Filesystem. You will be importing existing code from the folder “Lab3” located under your LKI/labs path.



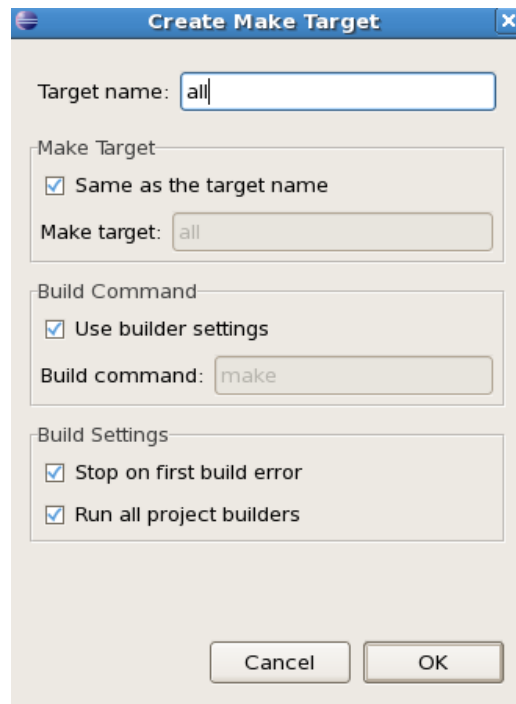
- Choose both the “Makefile” and “hello.c” and click finish.

### File system

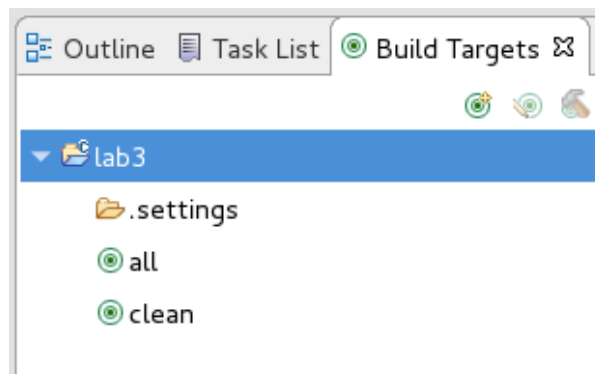
Import resources from the local file system.



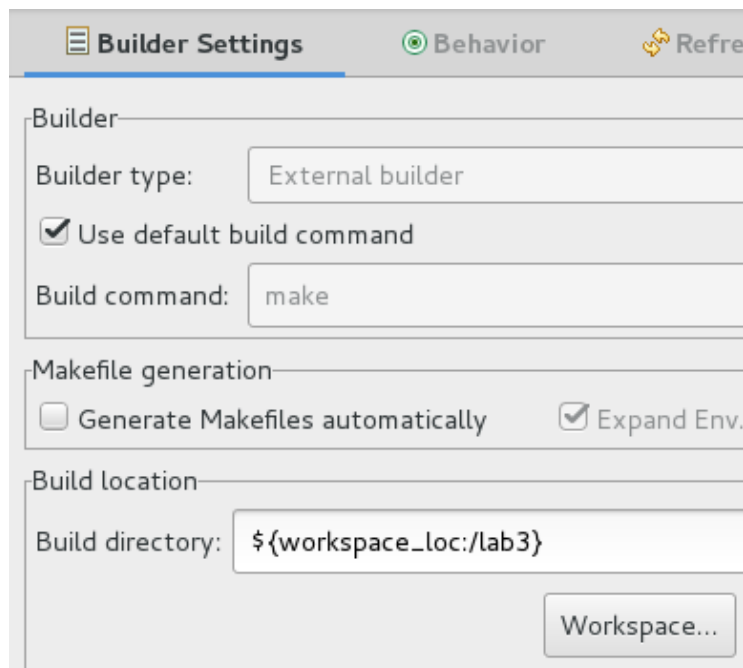
6. The Lab3 project is now added to Eclipse. In the right side of the main window, select the “Build Targets” tab and Select “New”.
7. Call the target name “all” and leave the other settings unmodified.



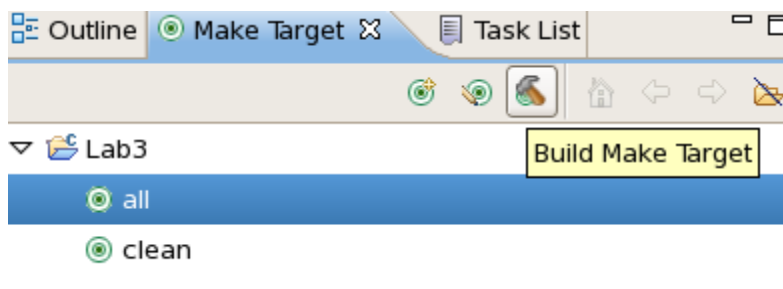
8. Create another target name called “clean”.
9. You should have 2 targets now in the “Make Target” tab window.



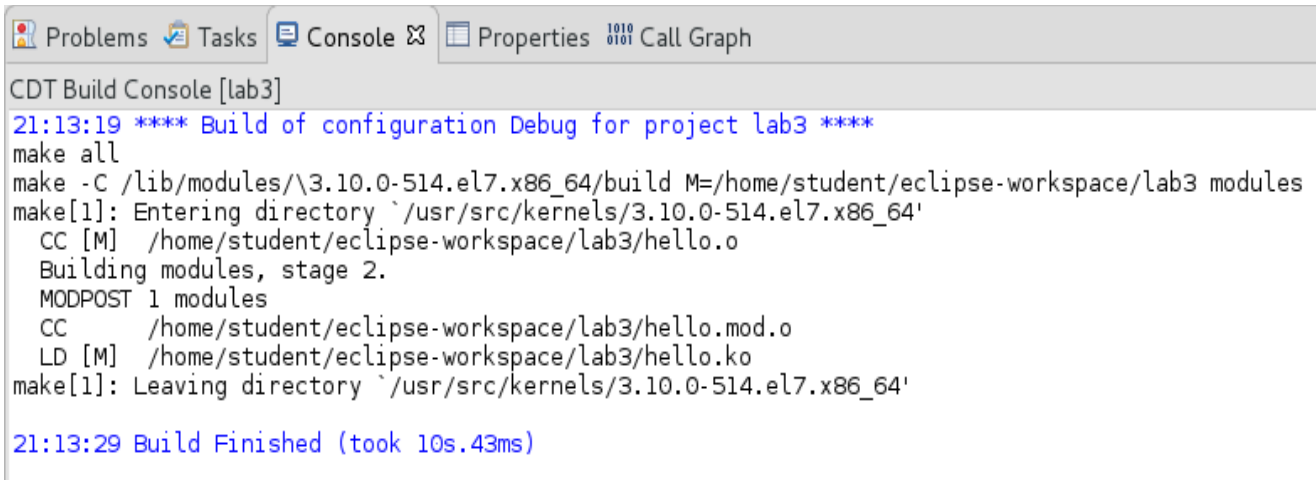
10. One final thing we need to do is modify the build path to our “Lab3” folder. To do that right-click on the Lab3 project in the Project Explorer and select Properties. Select *C/C++Build*
11. Uncheck the “Generate Makefile automatically” option and change the build directory to “Lab3”. Apply the changes and click OK



12. Finally, to build this module select “all” from the “Make Targets” and observe the console window for build information. This will show you syntax and library errors during the compilation process. If all was setup correctly, your module should build correctly.



## COMPILING A LINUX KERNEL MODULE

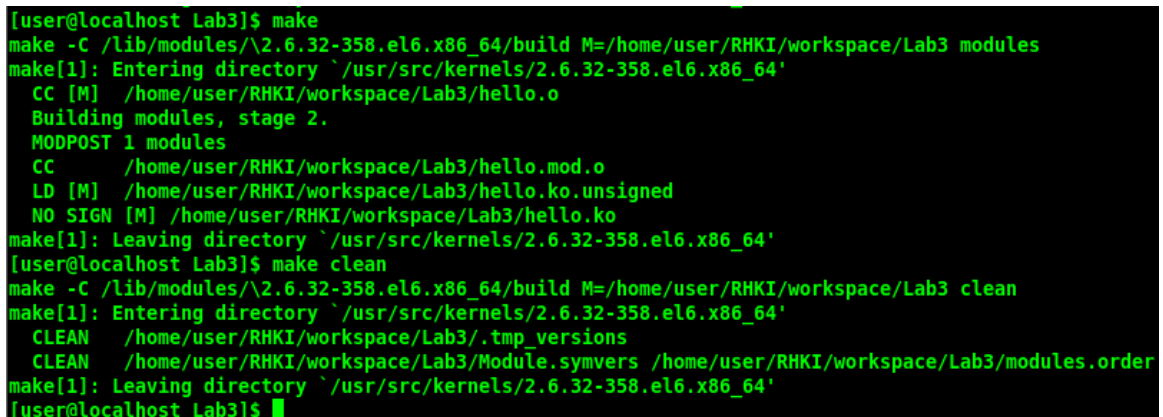


```
Problems Tasks Console Properties 1010 0101 Call Graph
CDT Build Console [lab3]
21:13:19 **** Build of configuration Debug for project lab3 ****
make all
make -C /lib/modules/\3.10.0-514.el7.x86_64/build M=/home/student/eclipse-workspace/lab3 modules
make[1]: Entering directory `/usr/src/kernels/3.10.0-514.el7.x86_64'
CC [M] /home/student/eclipse-workspace/lab3/hello.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/student/eclipse-workspace/lab3/hello.mod.o
LD [M] /home/student/eclipse-workspace/lab3/hello.ko
make[1]: Leaving directory `/usr/src/kernels/3.10.0-514.el7.x86_64'

21:13:29 Build Finished (took 10s.43ms)
```

13. Execute a “clean” to ensure that works as well to complete this lab.

*Note: You DO NOT have to use Eclipse if you prefer the command line method. The obvious advantage of Eclipse is the integration of the IDE, Syntax checking and interactive inclusion of functions and variables. You may choose to instead use vi or similar editor:*



```
[user@localhost Lab3]$ make
make -C /lib/modules/\2.6.32-358.el6.x86_64/build M=/home/user/RHKI/workspace/Lab3 modules
make[1]: Entering directory `/usr/src/kernels/2.6.32-358.el6.x86_64'
CC [M] /home/user/RHKI/workspace/Lab3/hello.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/user/RHKI/workspace/Lab3/hello.mod.o
LD [M] /home/user/RHKI/workspace/Lab3/hello.ko.unsigned
NO SIGN [M] /home/user/RHKI/workspace/Lab3/hello.ko
make[1]: Leaving directory `/usr/src/kernels/2.6.32-358.el6.x86_64'
[user@localhost Lab3]$ make clean
make -C /lib/modules/\2.6.32-358.el6.x86_64/build M=/home/user/RHKI/workspace/Lab3 clean
make[1]: Entering directory `/usr/src/kernels/2.6.32-358.el6.x86_64'
CLEAN /home/user/RHKI/workspace/Lab3/.tmp_versions
CLEAN /home/user/RHKI/workspace/Lab3/Module.symvers /home/user/RHKI/workspace/Lab3/modules.order
make[1]: Leaving directory `/usr/src/kernels/2.6.32-358.el6.x86_64'
[user@localhost Lab3]$
```

14. Finally, make use of “cscope” throughout the week (and after) while reviewing the Linux kernel source tree. “cscope” is an interactive tool that allows you to browse the source tree use a curses based command line application.

In your “/home/student/LKI/apps/cscope” directory you will find the following files:

```
[user@localhost cscope]$ pwd
/home/user/RHKL/apps/cscope
[user@localhost cscope]$ ls -alrt
total 801628
drwxrwxr-x. 4 user user      86 Sep  8 18:07 .
-rwxrwxr-x. 1 user user     523 Sep  8 18:18 build-linux-cscope.sh
-rw-rw-r--. 1 user user  3245872 Sep  8 18:19 cscope.files
-rw-rw-r--. 1 user user 440214168 Sep  8 18:20 cscope.po.out
-rw-rw-r--. 1 user user  61038592 Sep  8 18:20 cscope.in.out
-rw-rw-r--. 1 user user 316358275 Sep  8 18:20 cscope.out
drwxrwxr-x. 2 user user    4096 Sep  8 18:20 .
[user@localhost cscope]$
```

To generate the database flat files used by “cscope” the following script is provided to you:

```
[user@localhost cscope]$ cat build-linux-cscope.sh
#!/bin/sh

# Automatically create cscope database for
# the linux source tree

LNK=/home/user/rpmbuild/BUILD/kernel-3.10.0-123.el7/linux-3.10.0-123.el7.x86_64
find $LNK
    -path "$LNK/arch/*" ! -path "$LNK/arch/x86*" -prune -o
    -path "$LNK/Documentation*" -prune -o
    -path "$LNK/scripts*" -prune -o
    -name ".*[chxs$]" -print > cscope.files

cscope -b -q -k
```

To run “cscope” with the local files as input, use the command “**cscope -d**”.

```
Find this C symbol:
Find this global definition:
Find functions called by this function:
Find functions calling this function:
Find this text string:
Change this text string:
Find this egrep pattern:
Find this file: module.h
Find files #including this file:
Find assignments to this symbol:
```

```
File: module.h

File
0 /home/user/rpmbuild/BUILD/kernel-3.10.0-123.el7/linux-3.10.0-123.el7.x86_64/arch/x86/include/asm/module.h
1 /home/user/rpmbuild/BUILD/kernel-3.10.0-123.el7/linux-3.10.0-123.el7.x86_64/arch/x86/un/asm/module.h
2 /home/user/rpmbuild/BUILD/kernel-3.10.0-123.el7/linux-3.10.0-123.el7.x86_64/drivers/edac/edac_module.h
3 /home/user/rpmbuild/BUILD/kernel-3.10.0-123.el7/linux-3.10.0-123.el7.x86_64/include/asm-generic/module.h
4 /home/user/rpmbuild/BUILD/kernel-3.10.0-123.el7/linux-3.10.0-123.el7.x86_64/include/drm/ttm/ttm_module.h
5 /home/user/rpmbuild/BUILD/kernel-3.10.0-123.el7/linux-3.10.0-123.el7.x86_64/include/linux/module.h
6 /home/user/rpmbuild/BUILD/kernel-3.10.0-123.el7/linux-3.10.0-123.el7.x86_64/include/trace/events/module.h
7 /home/user/rpmbuild/BUILD/kernel-3.10.0-123.el7/linux-3.10.0-123.el7.x86_64/include/uapi/linux/module.h
8 /home/user/rpmbuild/BUILD/kernel-3.10.0-123.el7/linux-3.10.0-123.el7.x86_64/tools/virtio/linux/module.h
```

Using “cscope” you can browse the Linux kernel source tree, find references to symbols, load files, etc. all from the command line. The default editor is “vi”.