

Debugging Linux Kernel Source with Eclipse & QEMU in Fedora Core 11

Hyung Won Choi

<http://web.njit.edu/~hwc1027>

hwc1027@njit.edu

Acknowledgement

- These slides are based on Takis Blog.
 - <http://issaris.blogspot.com/2007/12/download-linux-kernel-sourcecode-from.html>
- All the credits go to the author of the Blog.

OS & Software Environment

- OS: Fedora Core 11
- Linux Kernel Source: 2.6.30.2
- IDE for Debugging: Eclipse-CDT
- Virtual Machine for Debugging: QEMU
- Compiler: GCC (version 4.4)

1. Install Eclipse/Eclipse-CDT

- Fedora Core 10/11
 - Install Fedora Eclipse packages
 - From CD/DVD or with yum.
- Other distribution
 - Download "Eclipse IDE for C/C++ Developers" from Eclipse site & Install.
 - <http://www.eclipse.org/downloads/>

2. Download Linux Kernel

- Obtain Linux Kernel source from Linux Kernel website
 - <http://www.kernel.org>
 - I downloaded 2.6.30.2 version (on 7/20/09)
 - <http://kernel.org/pub/linux/kernel/v2.6/linux-2.6.30.2.tar.bz2>

3. Untar Linux Kernel source

- Change to “root” user in a Terminal/Shell

```
$ su -
```

- Untar Linux Kernel source:

```
$ cp linux-2.6.30.2.tar.bz2 /usr/local/src/
```

```
$ cd /usr/local/src/
```

```
$ tar jxvf linux-2.6.30.2.tar.bz2
```

4. Configure with .config (1)

```
$ mkdir -p /mnt/build/linux-2.6
```

```
$ cp /boot/config-2.6.29.4-  
167.fc11.i686.PAE /mnt/build/linux-  
2.6/.config
```

```
$ cd /usr/local/src/linux-2.6.30.2/
```

```
$ make oldconfig O=/mnt/build/linux-2.6
```

```
...
```

4. Configure with .config (2)

Kernel compression mode

- > 1. Gzip (KERNEL_GZIP) (NEW)
- 2. Bzip2 (KERNEL_BZIP2) (NEW)
- 3. LZMA (KERNEL_LZMA) (NEW)

choice[1-3?]:

...

Strip assembler-generated symbols during link (STRIP_ASM_SYMS) [N/y/?] (NEW)

Support for extended (non-PC) x86 platforms (X86_EXTENDED_PLATFORM)
[Y/n/?]

Support non-standard 32-bit SMP architectures (X86_32_NON_STANDARD)
[N/y/?]

Paravirtualization layer for spinlocks (PARAVIRT_SPINLOCKS) [N/y/?]

...

// It asked many things. I entered “default” for all the questions.

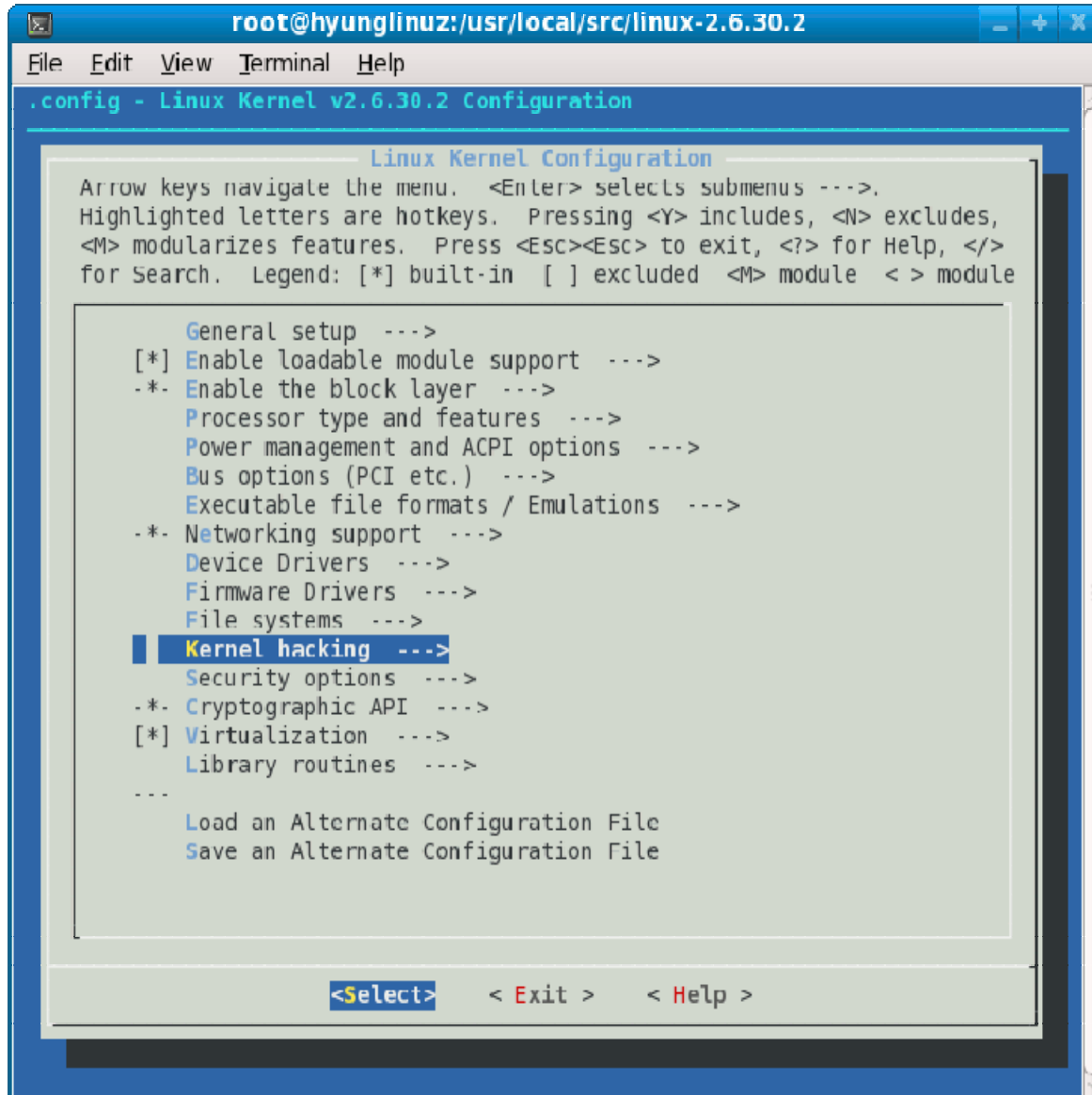
configuration written to .config

4. Configure with .config (3)

```
$ make menuconfig O=/mnt/build/linux-2.6
```

4. Configure with .config (3)

- Select “Kernel Hacking →”



```
root@hyunglinuz:/usr/local/src/linux-2.6.30.2
File Edit View Terminal Help
.config - Linux Kernel v2.6.30.2 Configuration

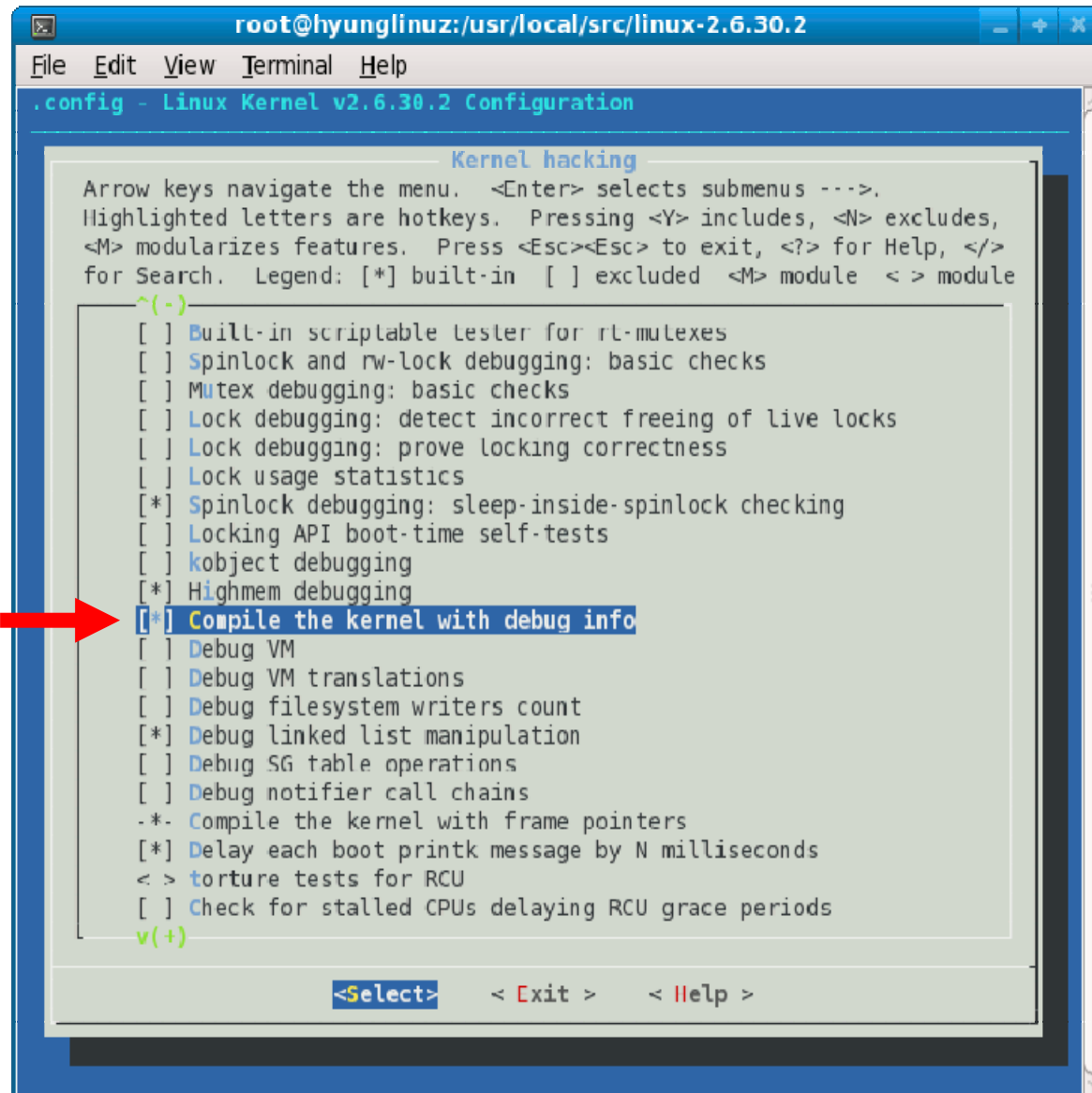
Linux Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module <> module

General setup --->
[*] Enable loadable module support --->
-* Enable the block layer --->
Processor type and features --->
Power management and ACPI options --->
Bus options (PCI etc.) --->
Executable file formats / Emulations --->
-* Networking support --->
Device Drivers --->
Firmware Drivers --->
File systems --->
Kernel hacking --->
Security options --->
-* Cryptographic API --->
[*] Virtualization --->
Library routines --->
...
Load an Alternate Configuration File
Save an Alternate Configuration File

<Select> < Exit > < Help >
```

4. Configure with .config (3)

- Enable “Compile the kernel with debug info”



```
root@hyunglinuz:/usr/local/src/linux-2.6.30.2
File Edit View Terminal Help
.config - Linux Kernel v2.6.30.2 Configuration

Kernel hacking
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < > module

^(-)
[ ] Built-in scriptable tester for rt-mutexes
[ ] Spinlock and rw-lock debugging: basic checks
[ ] Mutex debugging: basic checks
[ ] Lock debugging: detect incorrect freeing of live locks
[ ] Lock debugging: prove locking correctness
[ ] Lock usage statistics
[*] Spinlock debugging: sleep-inside-spinlock checking
[ ] Locking API boot-time self-tests
[ ] kobject debugging
[*] Highmem debugging
[*] Compile the kernel with debug info
[ ] Debug VM
[ ] Debug VM translations
[ ] Debug filesystem writers count
[*] Debug linked list manipulation
[ ] Debug SG table operations
[ ] Debug notifier call chains
-*- Compile the kernel with frame pointers
[*] Delay each boot printk message by N milliseconds
< > torture tests for RCU
[ ] Check for stalled CPUs delaying RCU grace periods
v(+)

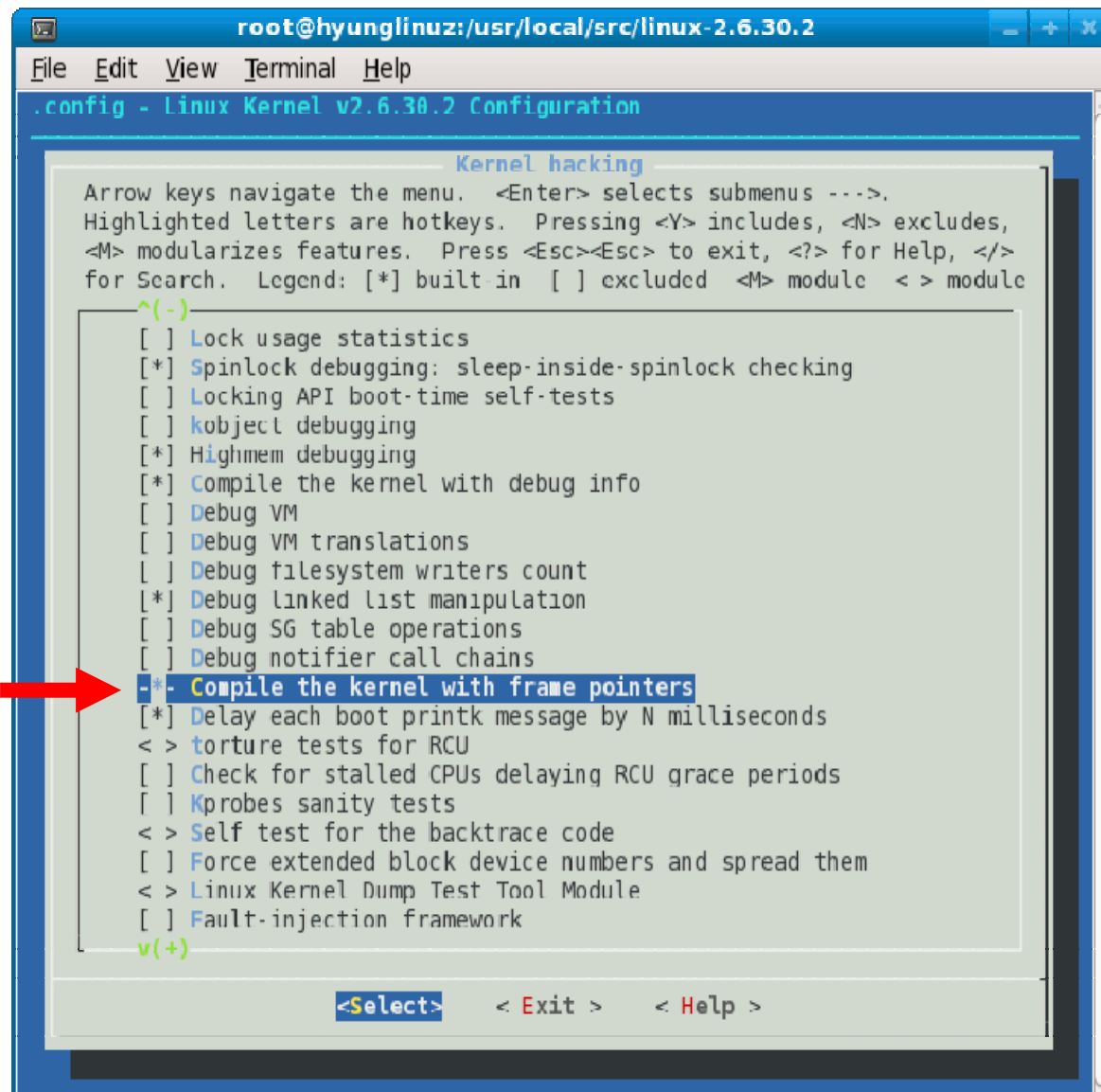
<Select> < Exit > < Help >
```

Already checked



4. Configure with .config (3)

- Enable “Compile the kernel with frame pointers”



```
root@hyunglinuz:/usr/local/src/linux-2.6.30.2
File Edit View Terminal Help
.config - Linux Kernel v2.6.30.2 Configuration

Kernel hacking
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module <> module
^(-)
[ ] Lock usage statistics
[*] Spinlock debugging: sleep-inside-spinlock checking
[ ] Locking API boot-time self-tests
[ ] kobject debugging
[*] Highmem debugging
[*] Compile the kernel with debug info
[ ] Debug VM
[ ] Debug VM translations
[ ] Debug filesystem writers count
[*] Debug linked list manipulation
[ ] Debug SG table operations
[ ] Debug notifier call chains
Already enabled → [*] Compile the kernel with frame pointers
[*] Delay each boot printk message by N milliseconds
< > torture tests for RCU
[ ] Check for stalled CPUs delaying RCU grace periods
[ ] Kprobes sanity tests
< > Self test for the backtrace code
[ ] Force extended block device numbers and spread them
< > Linux Kernel Dump Test Tool Module
[ ] Fault-injection framework
v(+)
```

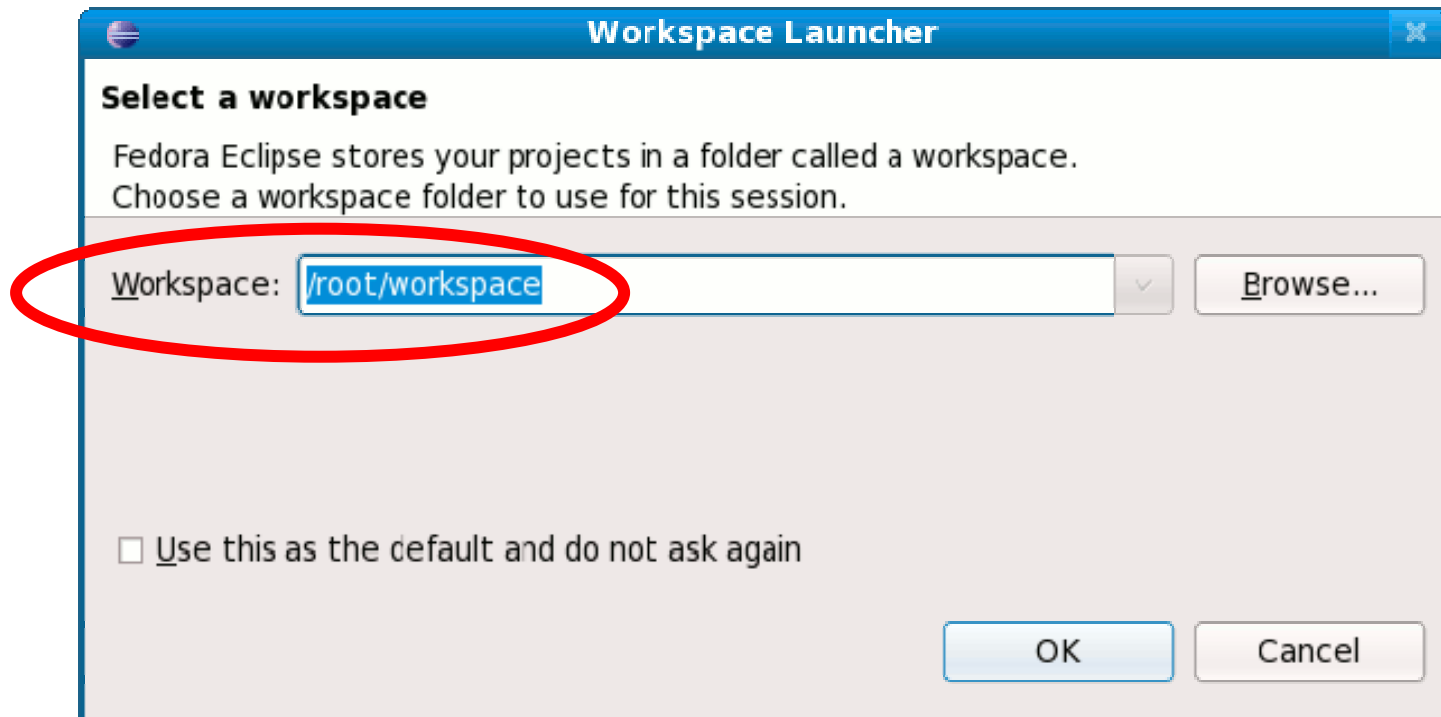
5. Run Eclipse (or Eclipse-cdt)

- In a terminal/shell:

```
$ eclipse
```

5. Run Eclipse (or Eclipse-cdt)

- “Select a workspace”: /root/workspace



5. Run Eclipse (or Eclipse-cdt)



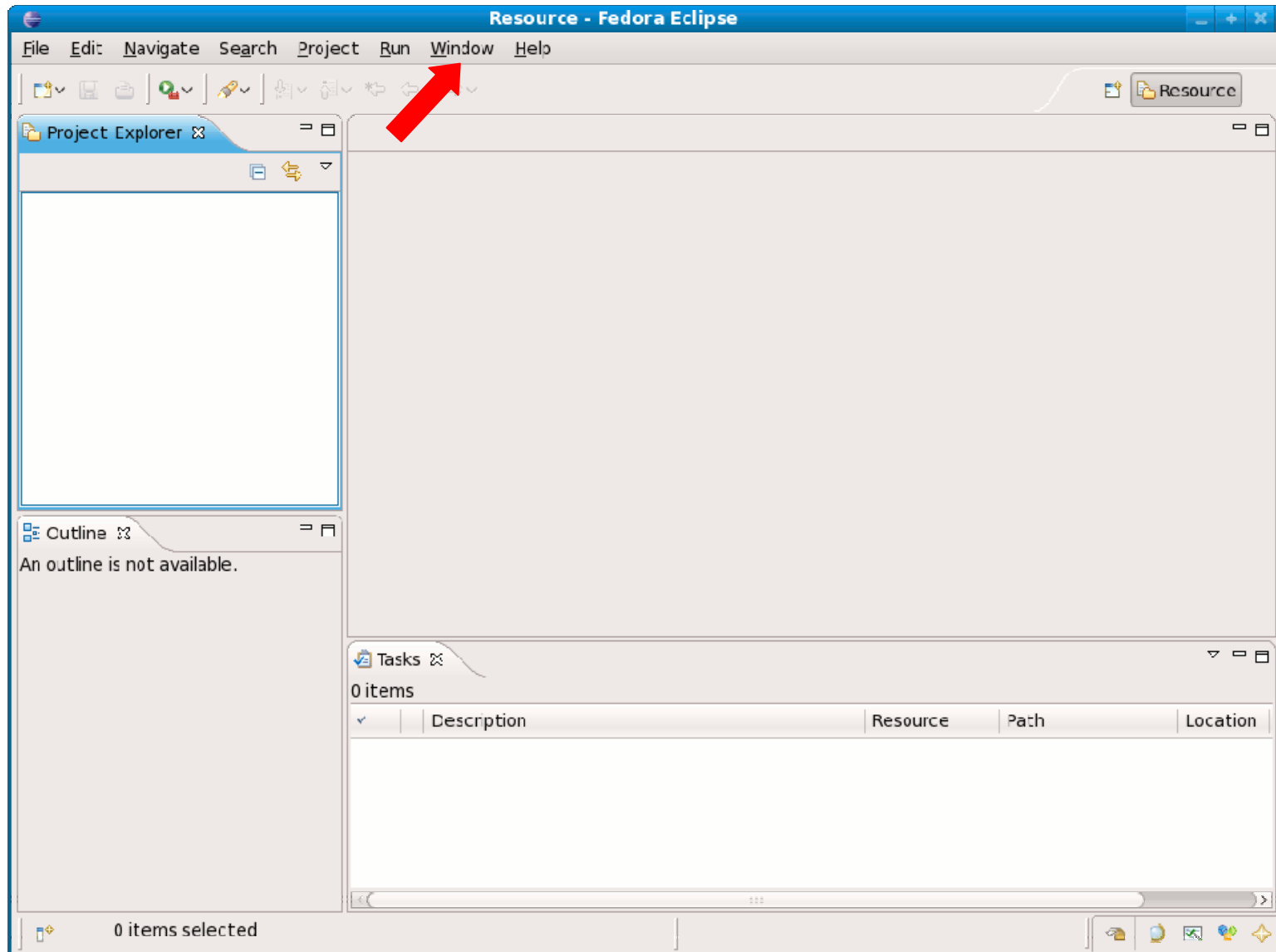
5. Run Eclipse (or Eclipse-cdt)

- “Go to the Workbench”



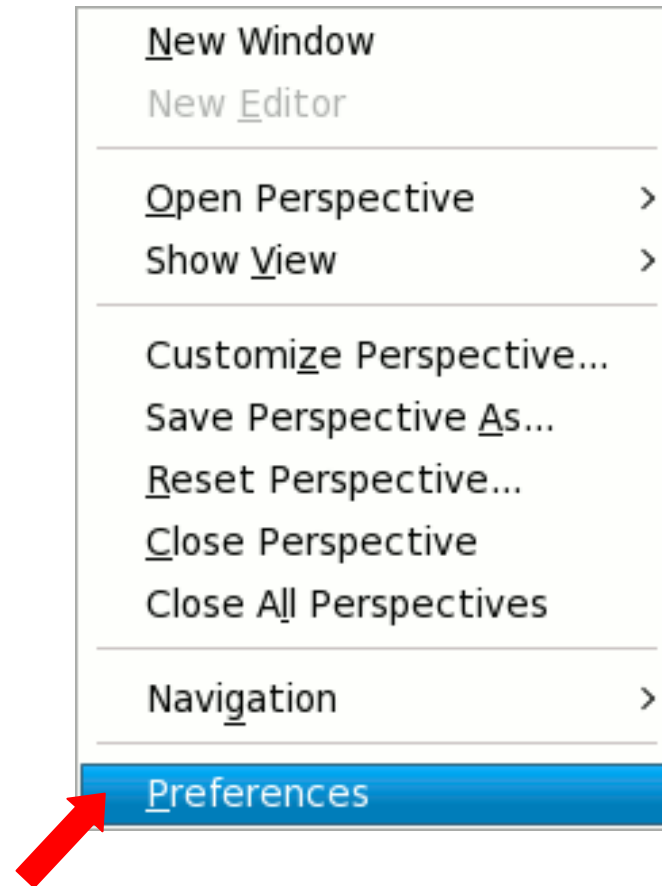
5. Run Eclipse (or Eclipse-cdt)

- Select “Window→ Preferences”:



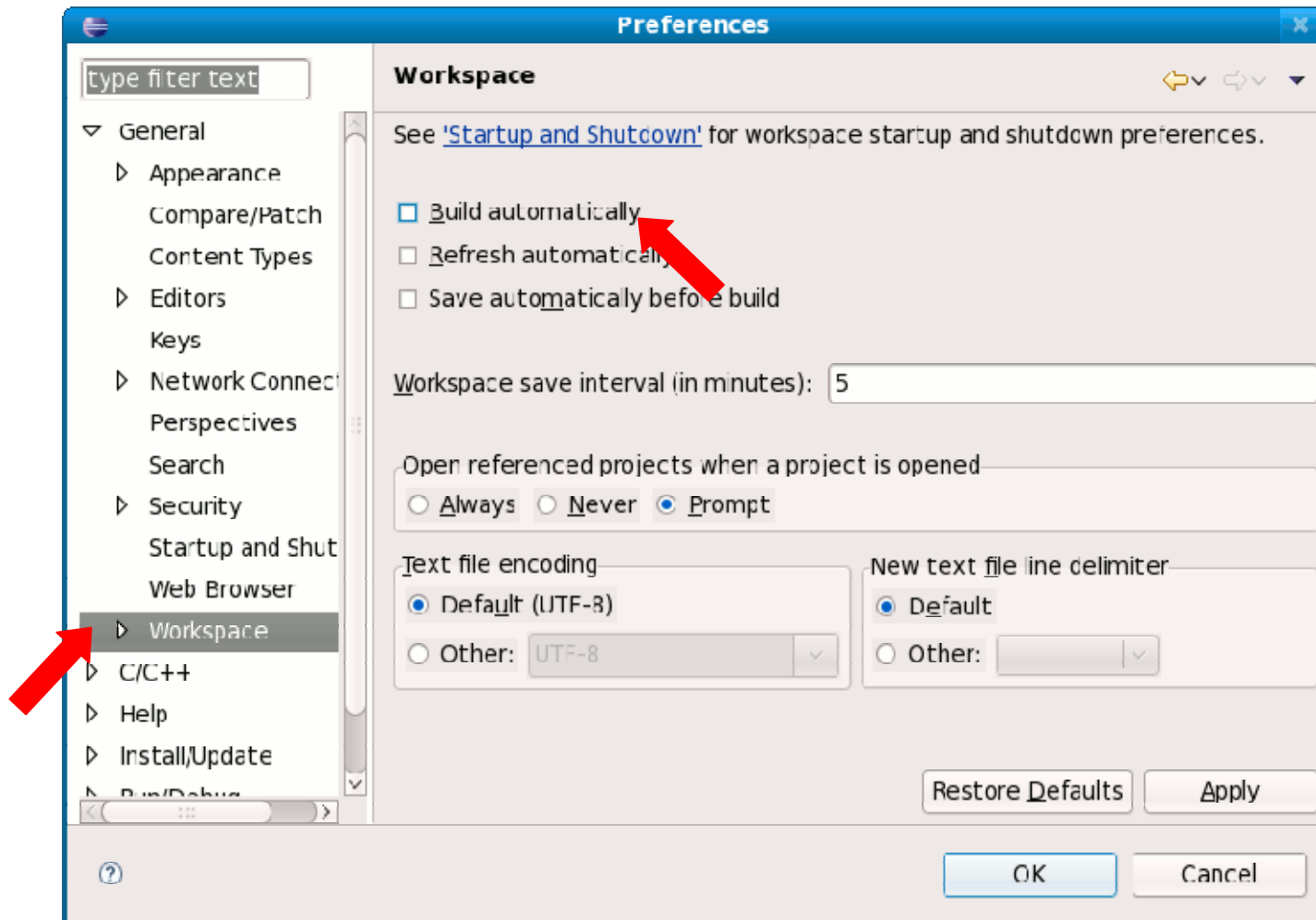
5. Run Eclipse (or Eclipse-cdt)

- Select “Window→ Preferences”:



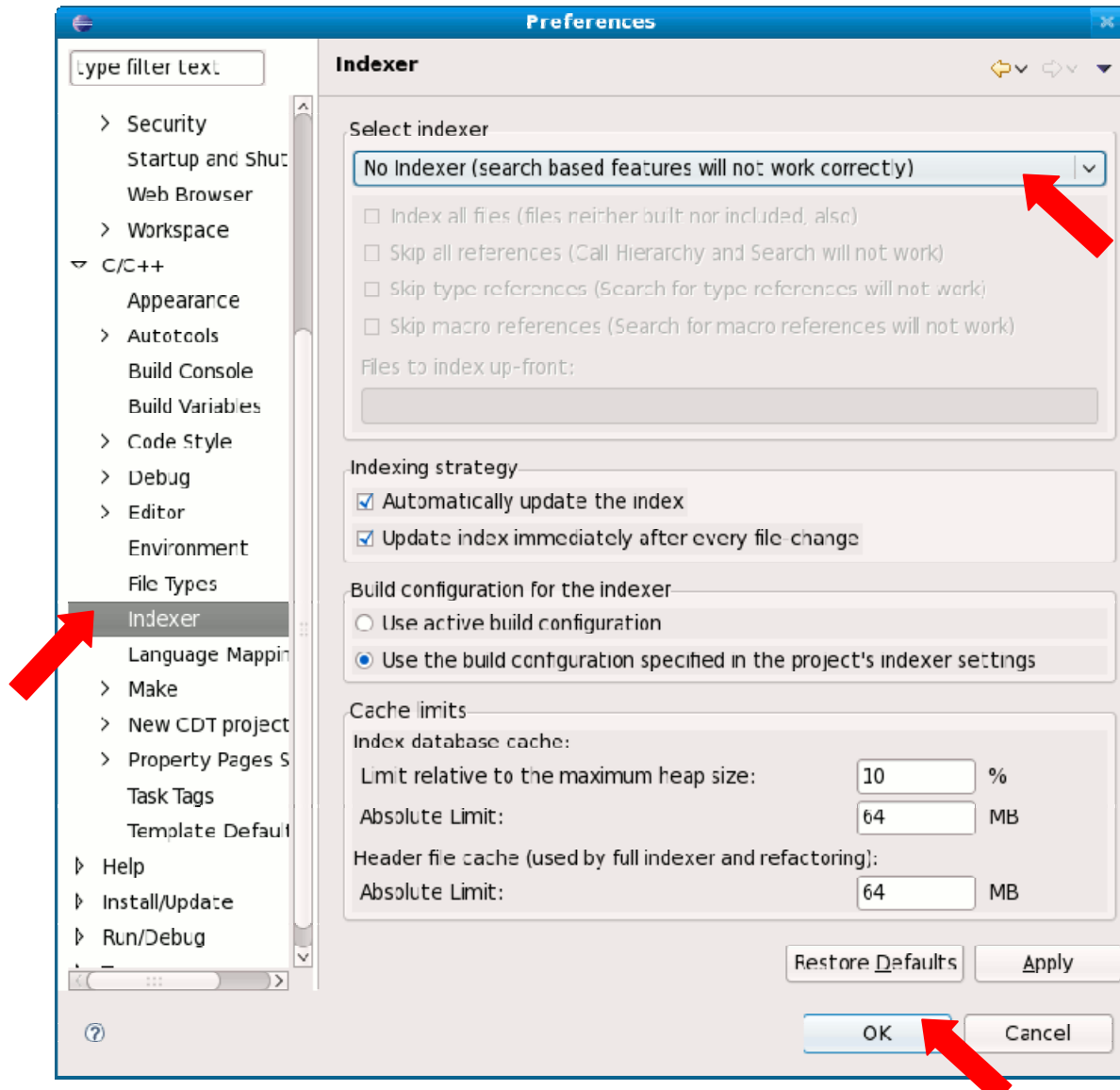
5. Run Eclipse (or Eclipse-cdt)

- Select “General → Workspace”
- Deselect “Build automatically” & “Apply”



5. Run Eclipse (or Eclipse-cdt)

- Select “C/C++ → Indexer”

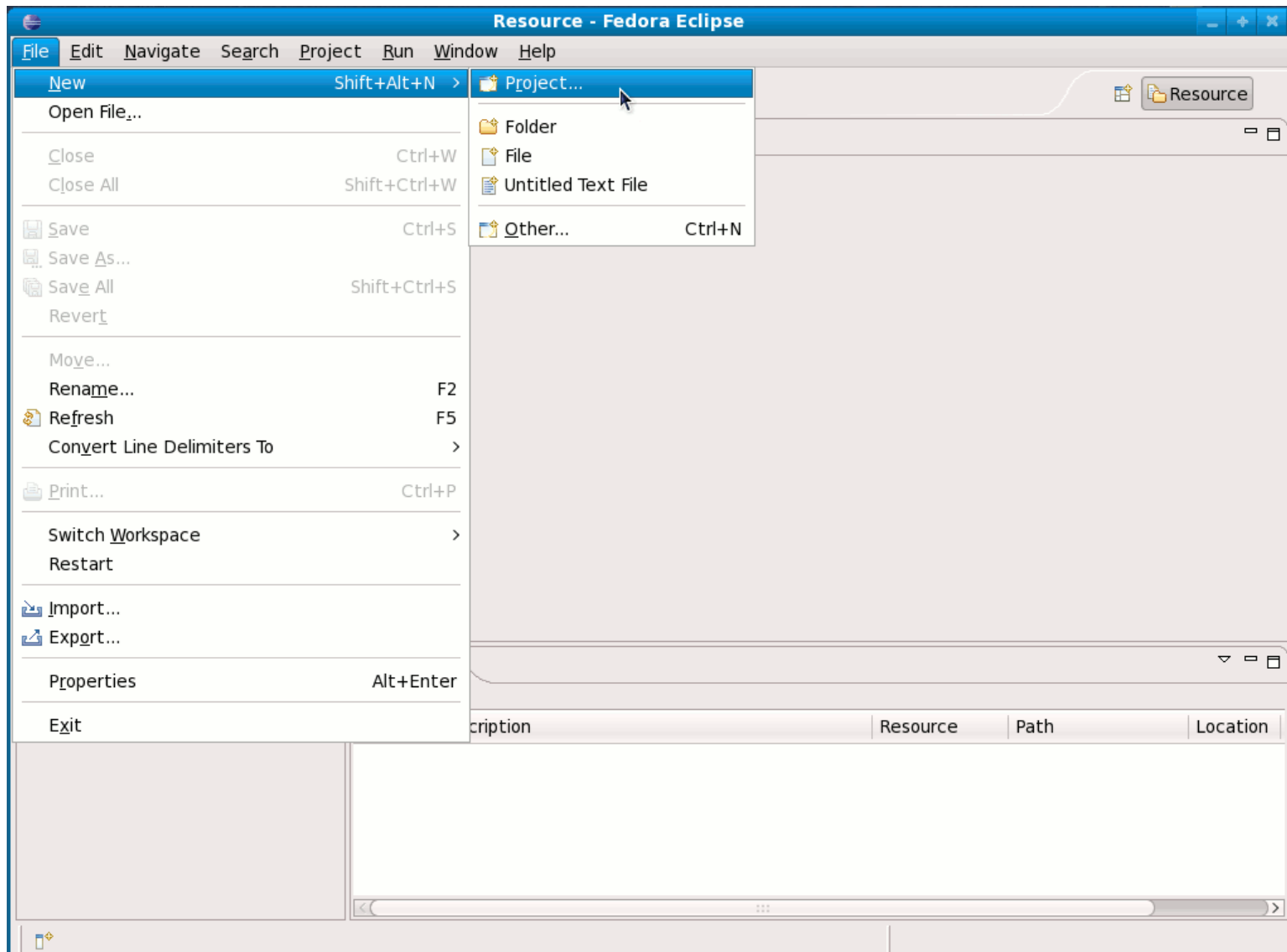


Switch "Fast C/C++ Indexer" to "No Indexer"

Click "OK"

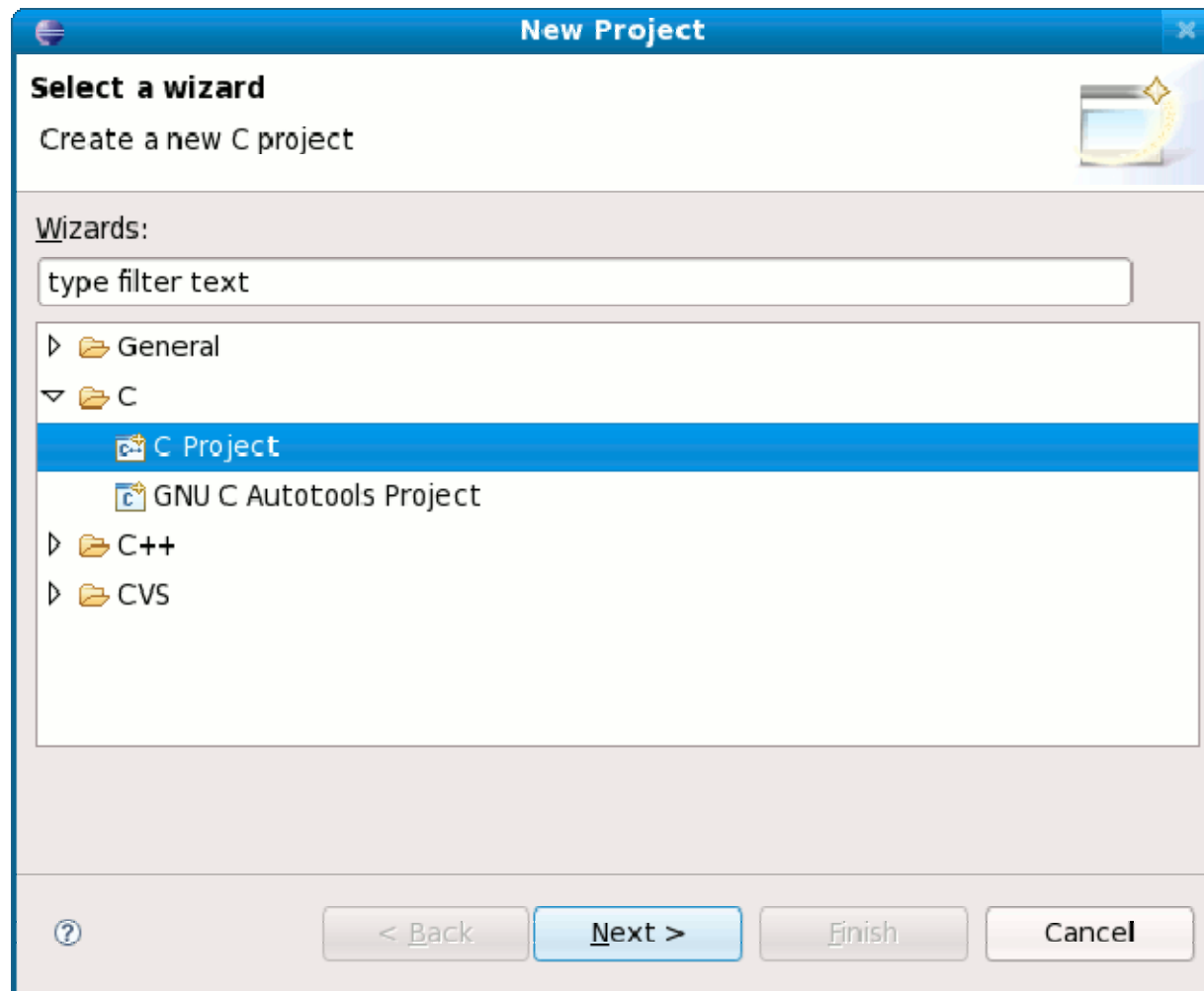
6. New Project

- “File→ New→ Project..” on Eclipse menu



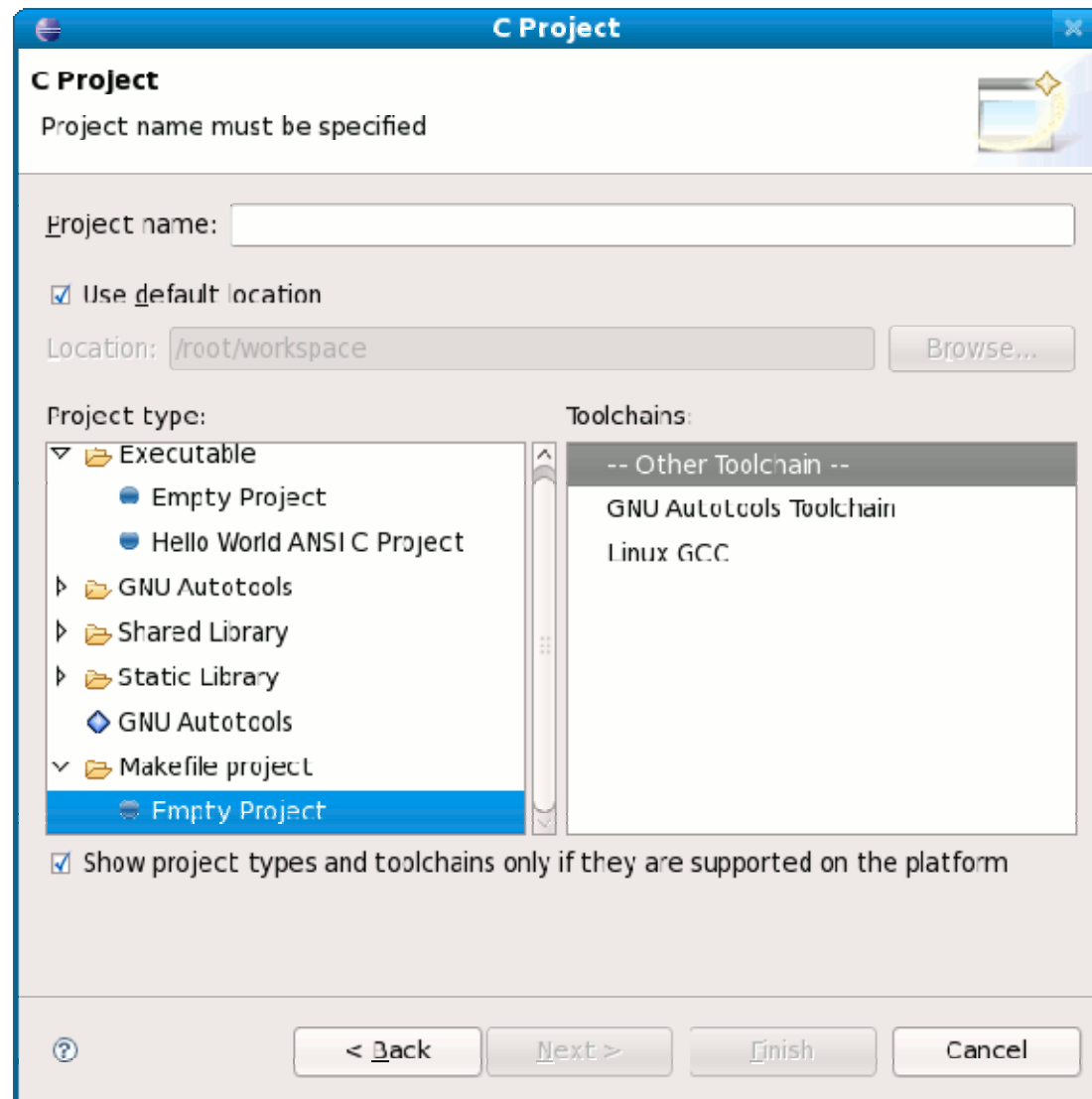
6. New Project

- Select “C→C Project” & click “Next”



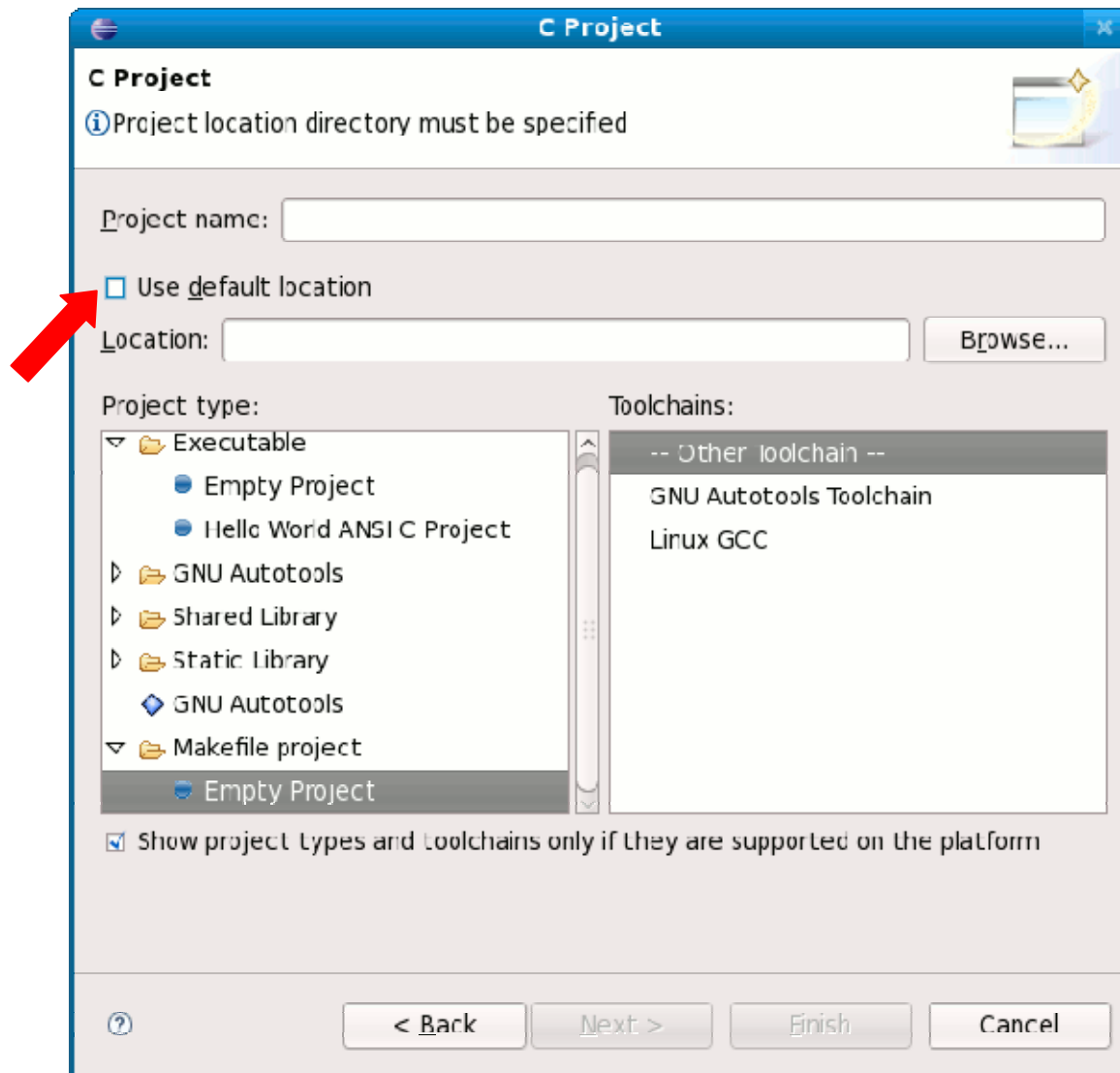
6. New Project

- Select “Makefile project” → “Empty Project”



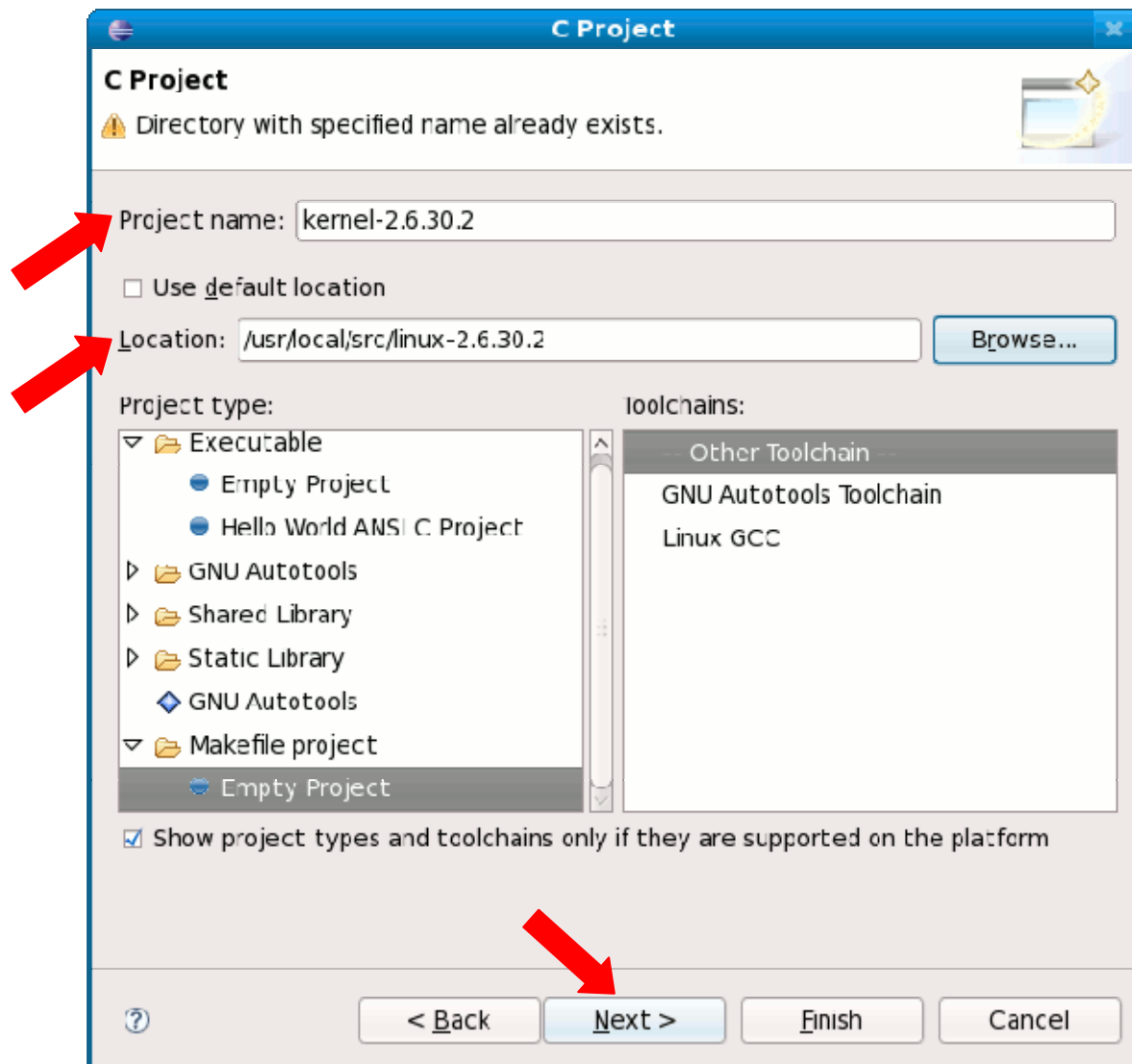
6. New Project

- Uncheck “Use default location” checkbox



6. New Project

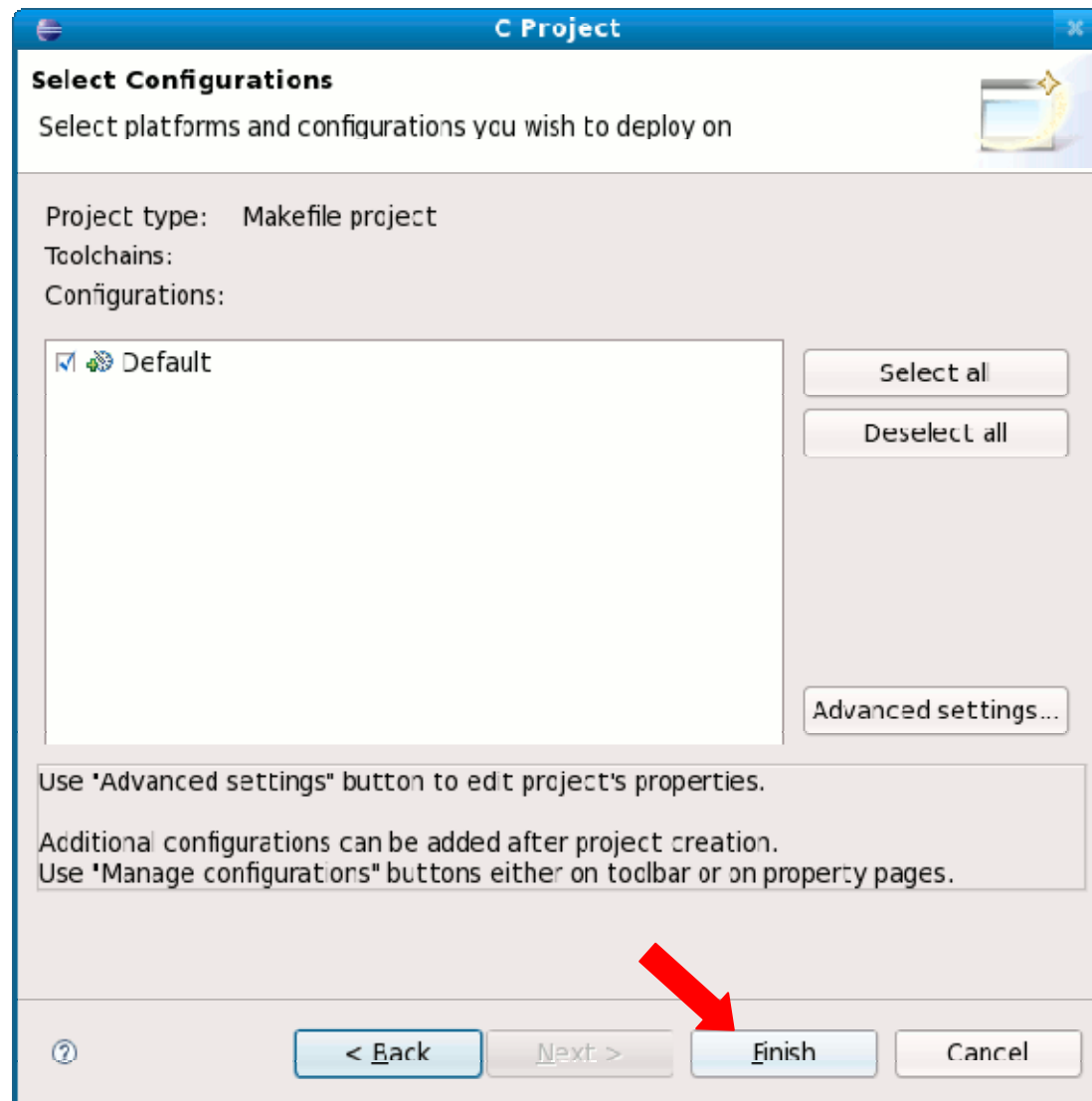
- Enter the project name in “Project name”
- Enter “/usr/local/src/linux-2.6.30.2” into “Location”



Click “Next”

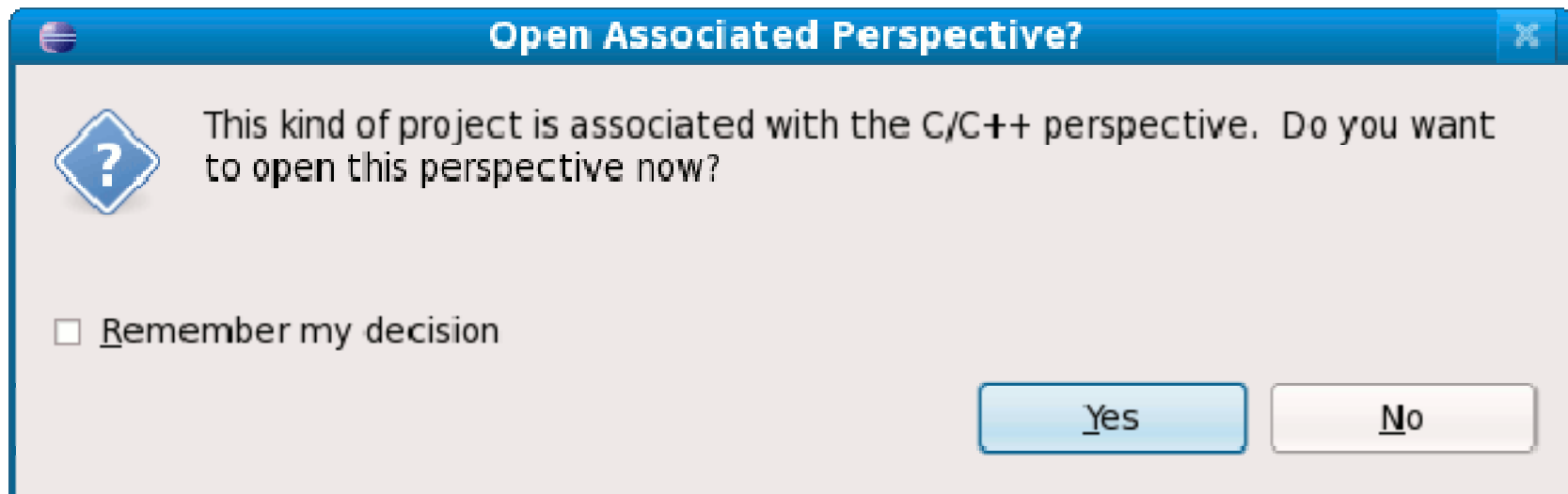
6. New Project

- Click “Finish”



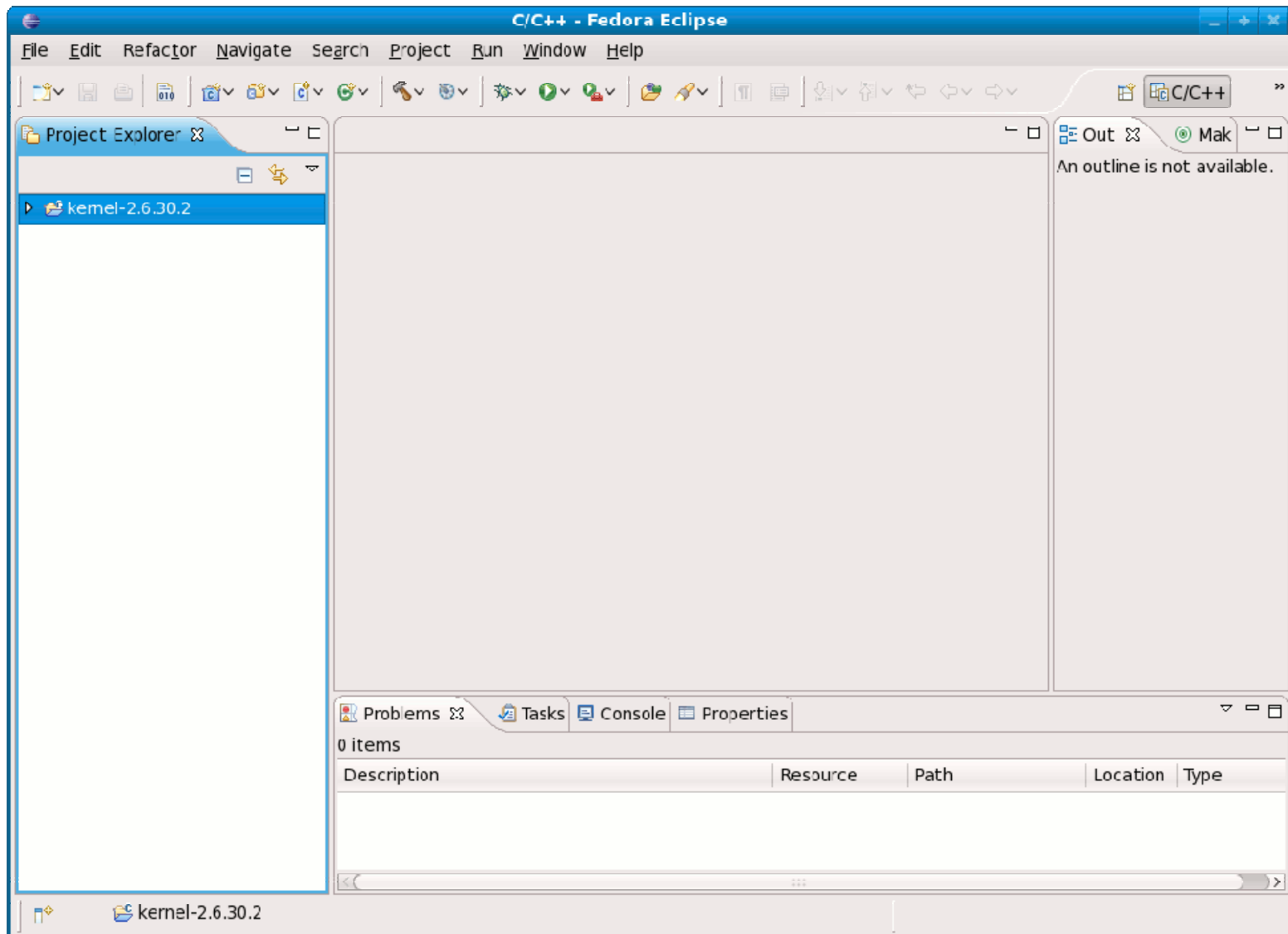
6. New Project

- Answer “Yes”



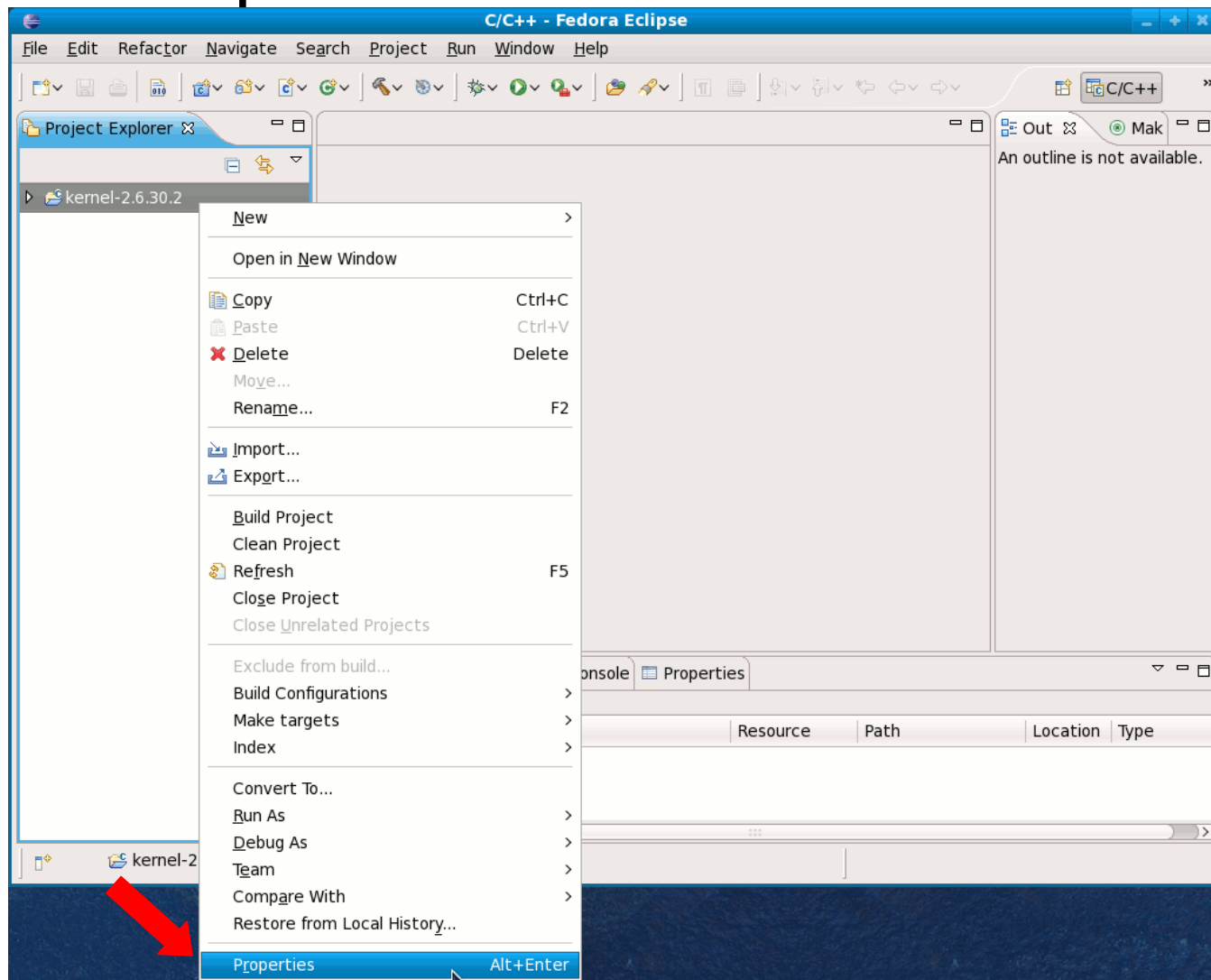
6. New Project

- New Project is created.



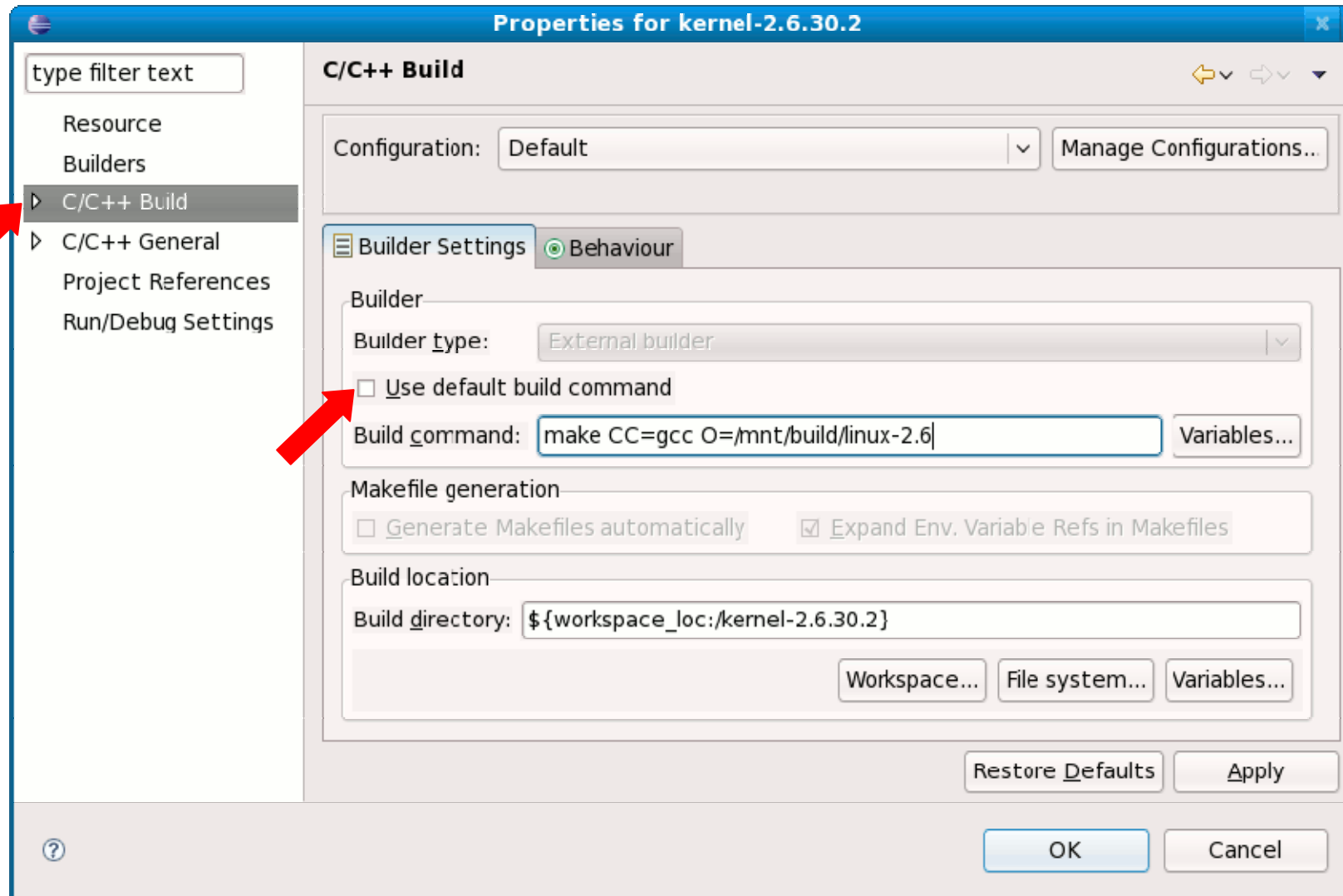
7. Configuring Project

- Click the right button (mouse) on the project.
- Select “Properties”



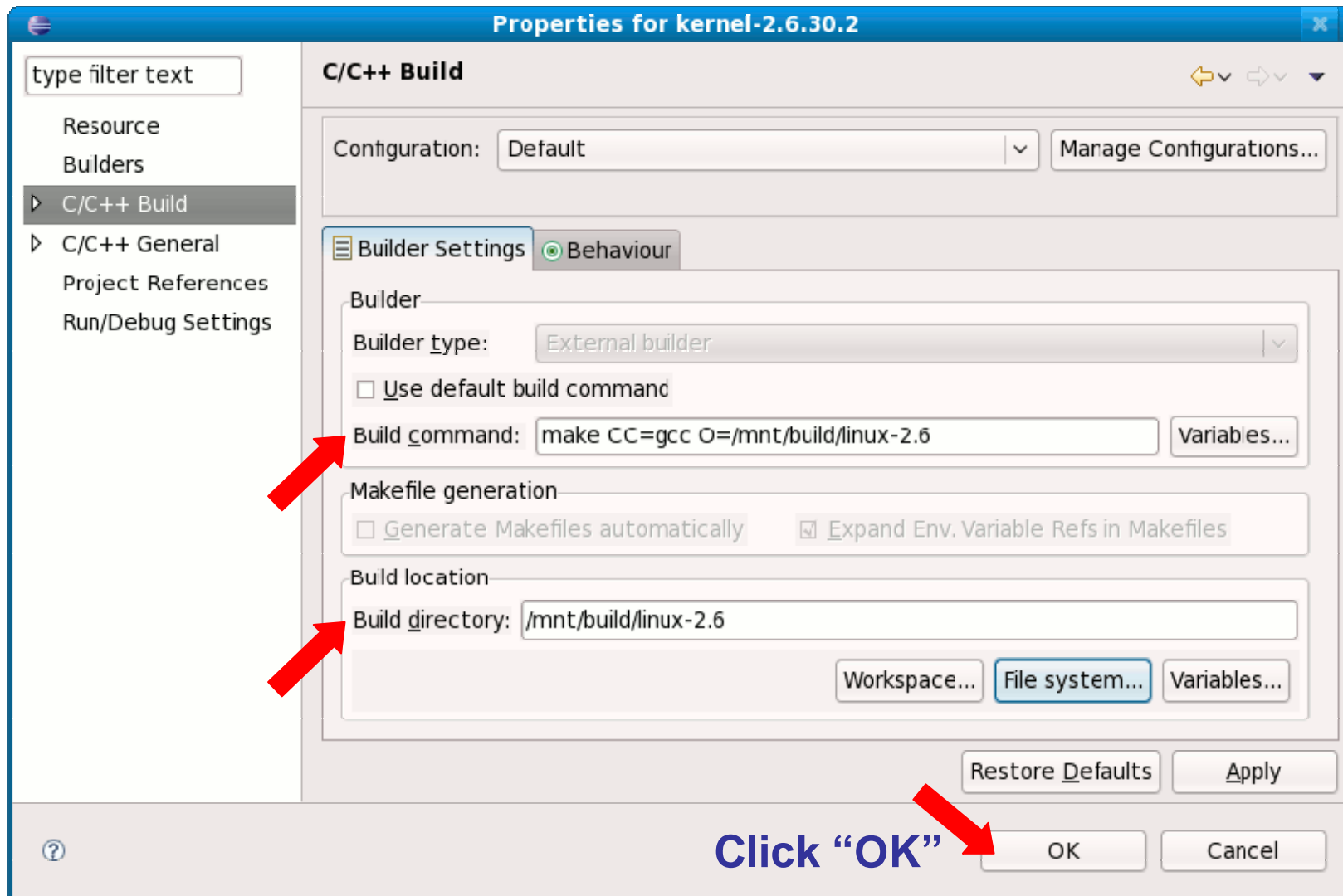
7. Configuring Project

- Select “C/C++ Builders”
- Uncheck “Use default build command”



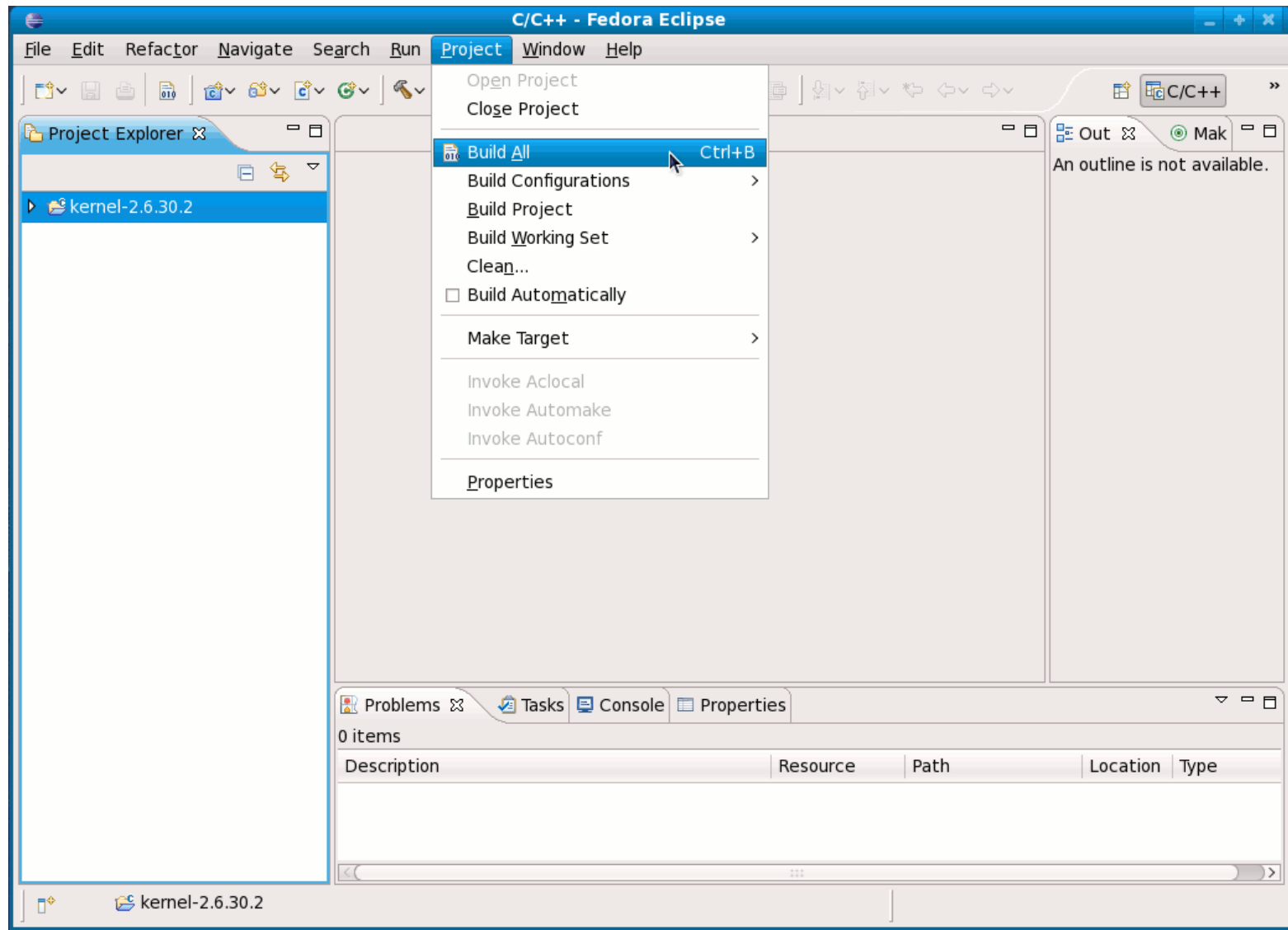
7. Configuring Project

- Enter “make CC=gcc O=/mnt/build/linux-2.6” in “Build command”
- Enter “/mnt/build/linux-2.6” in “Build directory” by “File system..”



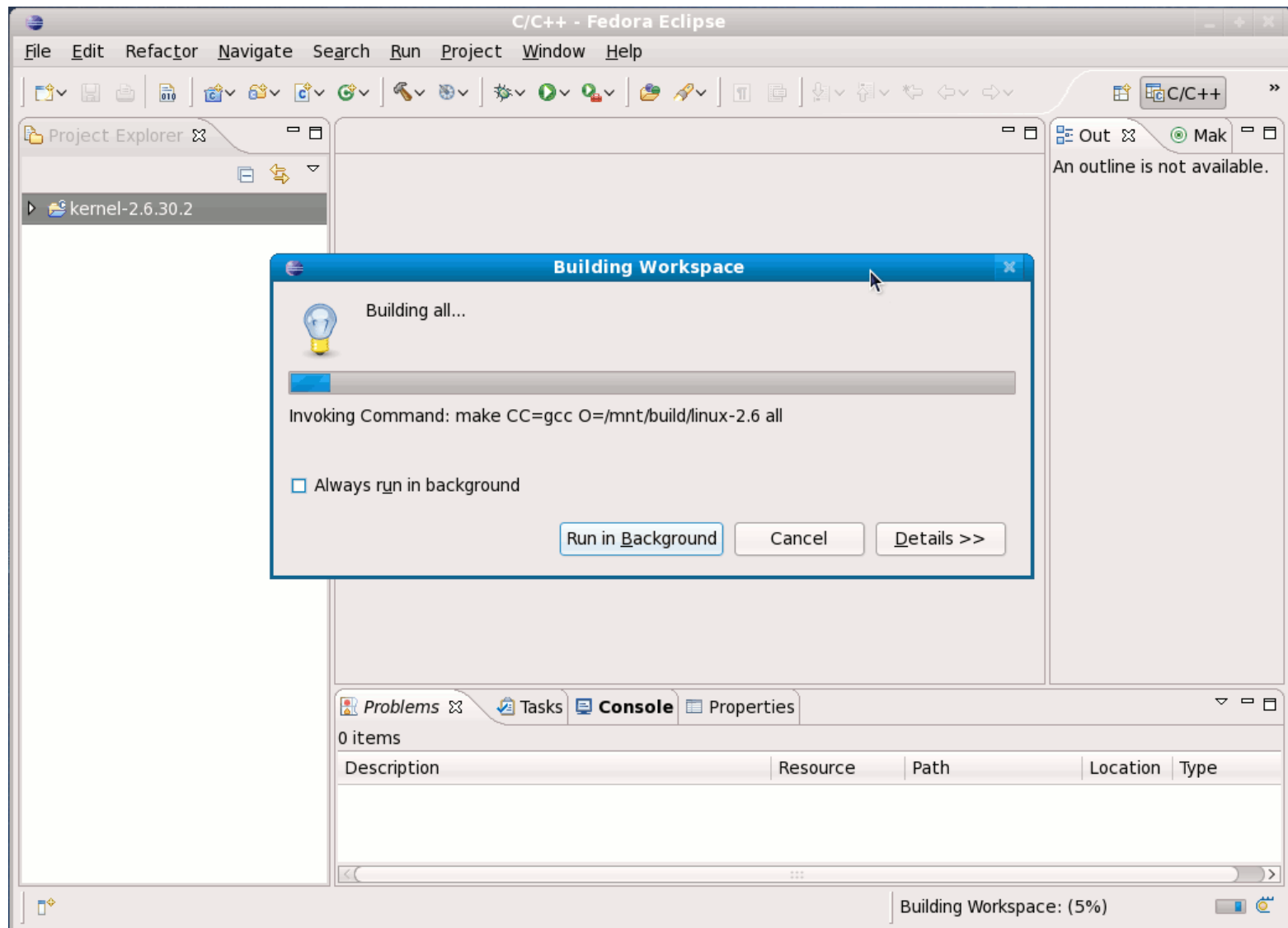
8. Build

- Select “Project→ Build all” on the menu



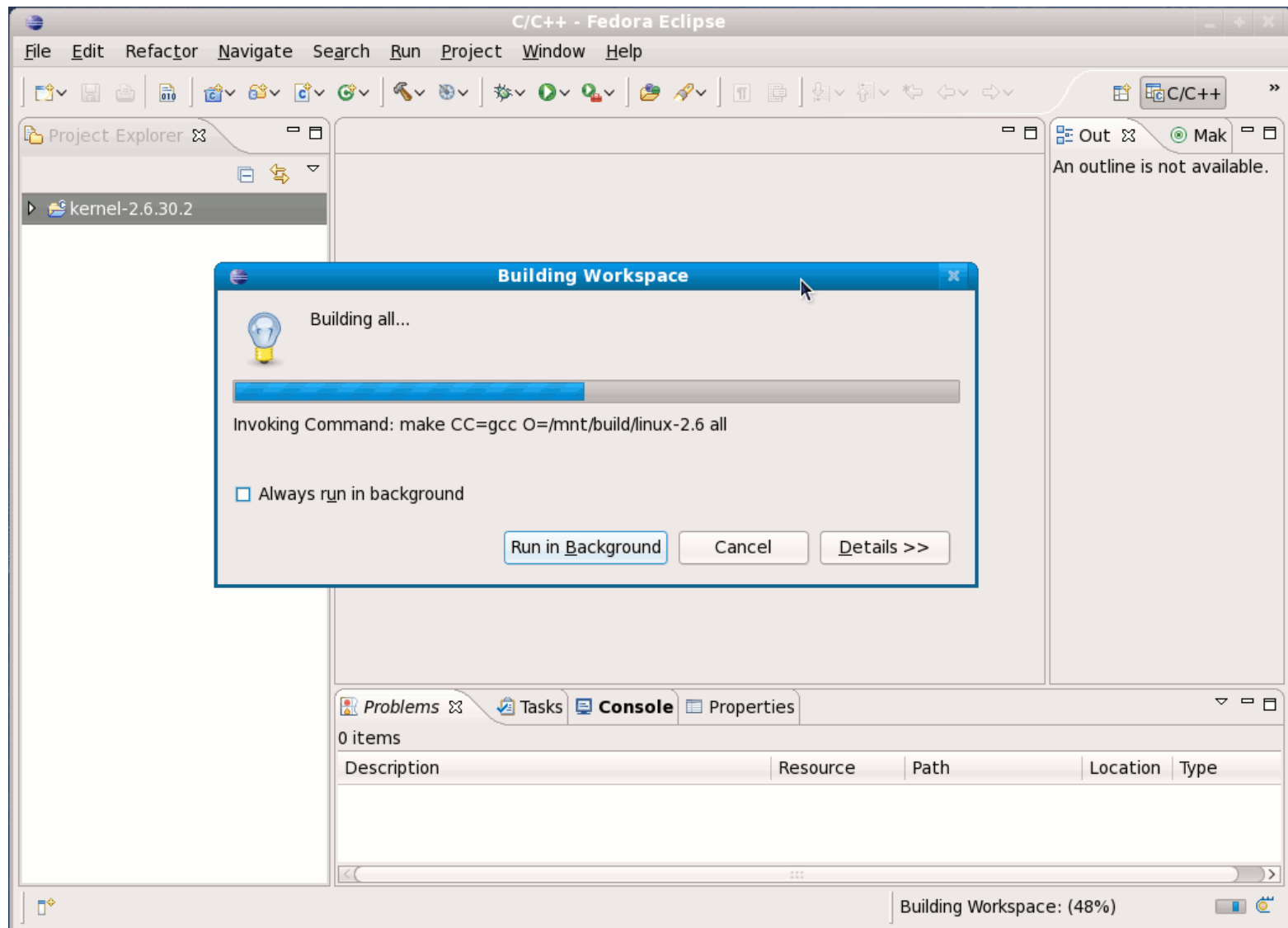
8. Build

- Shows the progress for building kernel



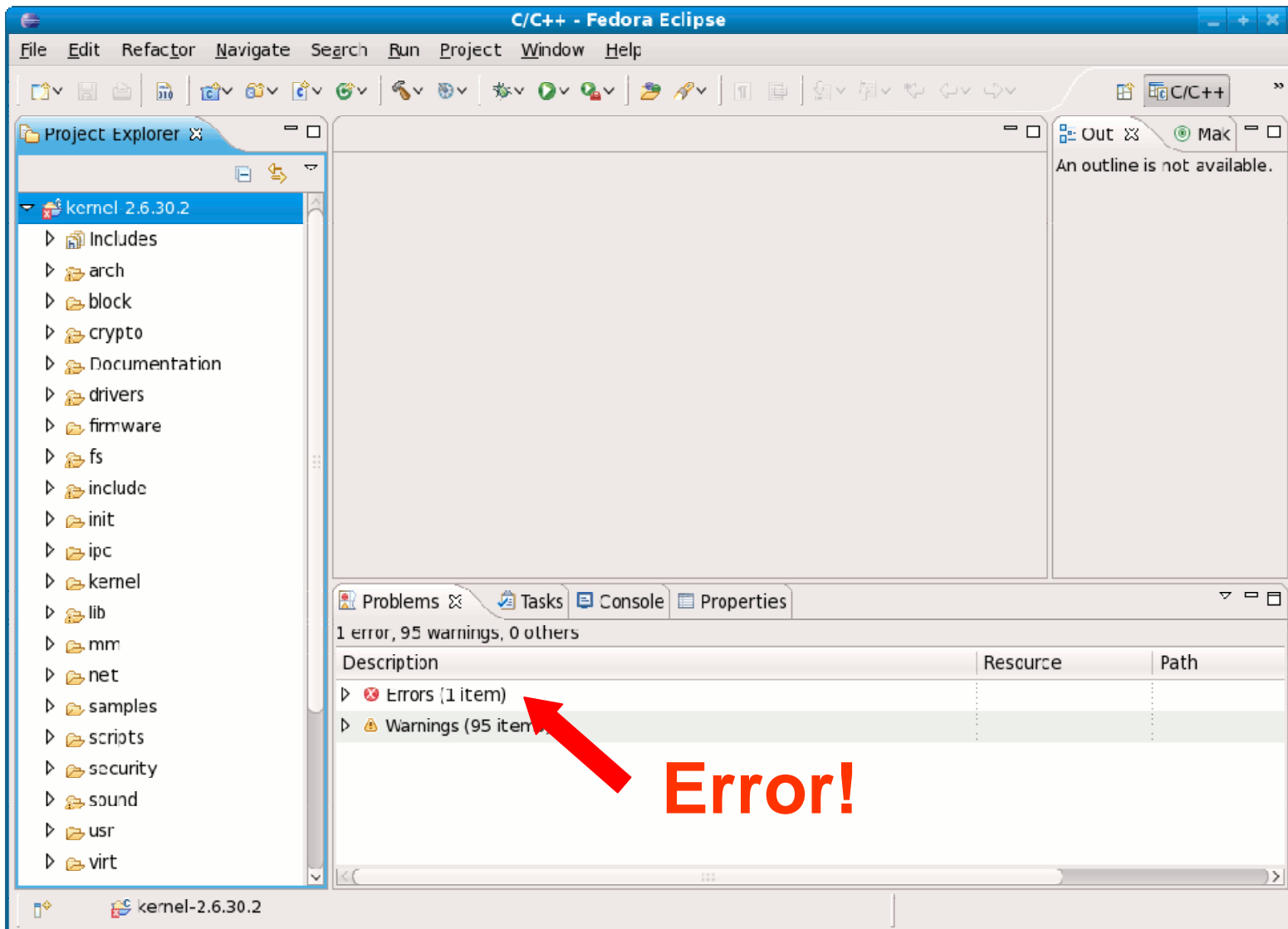
8. Build

- Shows the progress for building kernel



8. Build

- Found an error & Building was stopped.



8. Build

- “extern” (jffs2-user.h) caused an error.

The screenshot shows the Eclipse IDE interface. The main editor displays the file `jffs2-user.h` with the following code:

```
#undef cpu_to_jemode
#undef je16_to_cpu
#undef je32_to_cpu
#undef jemode_to_cpu
extern int target_endian;
#define t16(x) ({ _u16 __b = (x); (target_endian == __BYTE_ORDER)?
#define t32(x) ({ _u32 __b = (x); (target_endian == __BYTE_ORDER)?
#define cpu_to_je16(x) ((jint16_t){t15(x)})
#define cpu_to_je32(x) ((jint32_t){t32(x)})
#define cpu_to_jemode(x) ((jmode_t){t32(x)})
#define je16_to_cpu(x) (t16((x).v16))
#define je32_to_cpu(x) (t32((x).v32))
#define jemode_to_cpu(x) (t32((x).m))
```

A red arrow points to the line `extern int target_endian;`. The **Problems** window at the bottom shows the following error:

Description	Resource	Path
1 error, 95 warnings, 0 others		
Errors (1 item)		
/mnt/build/linux-2.6/usr/include/mtd/jffs2-user.h: extern's make no sense in userspace	kernel-2.6.30.2	
Warnings (95 items)		

The status bar at the bottom of the IDE displays the error message: `/mnt/build/linux-2.6/usr/include/mtd/jffs2-user.h: extern's make no sense in userspace`.

8. Build

- Modify jffs2-user.h file: remove “extern”
 - target_endian is only used in jffs2-user.h.

The screenshot shows the Eclipse IDE interface. The main editor window displays the file `jffs2-user.h` with the following code:

```
#undef cpu_to_jemode
#undef je16_to_cpu
#undef je32_to_cpu
#undef jemode_to_cpu

/* remove "extern" */
/* extern int target_endian; */
int target_endian;

#define t16(x) ({ __u16 __b = (x); (target_endian==__BYTE_ORDER)?
#define t32(x) ({ __u32 __b = (x); (target_endian==__LITTLE_ENDIAN)?

#define cpu_to_je16(x) ((jint16_t){t16(x)})
#define cpu_to_je32(x) ((jint32_t){t32(x)})
#define cpu_to_jemode(x) ((jmode_t){t32(x)})

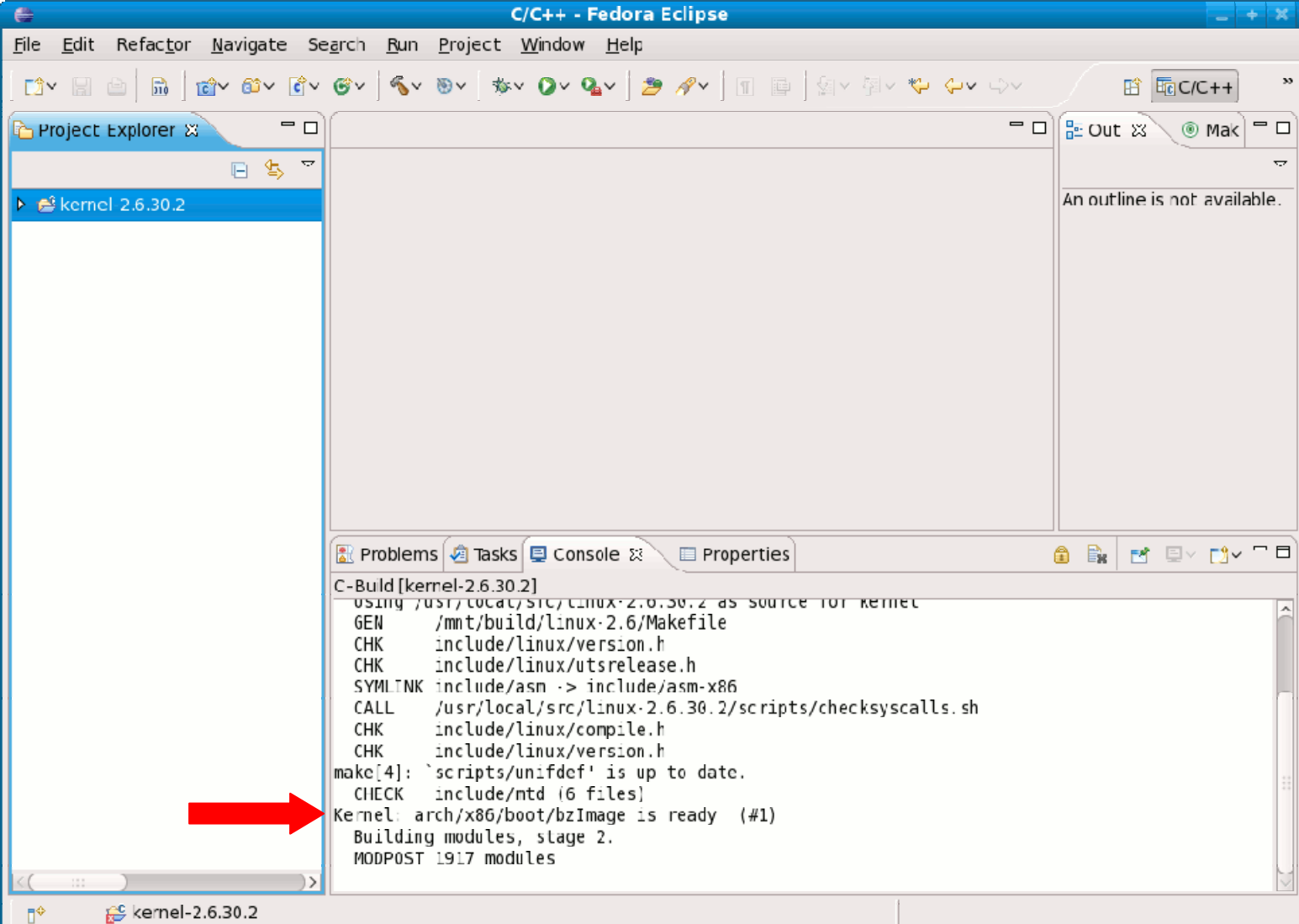
#define je16_to_cpu(x) (t16((x).v16))
```

A red arrow points to the line `int target_endian;`, indicating the removal of the `extern` keyword. The Project Explorer on the left shows the project structure for `kernel-2.6.30.2`. The Problems window at the bottom shows the following error:

Description	Resource	Path
Errors (1 item)		
/mnt/build/linux-2.6/usr/include/mtd/jffs2-user.h extern's make	kernel-2.6.30.2	
Warnings (95 items)		

8. Build

- Rebuild with “Project → Build all” & created a bzImage
- Kernel: arch/x86/boot/bzImage is ready (#1)



```
C/C++ - Fedora Eclipse
File Edit Refactor Navigate Search Run Project Window Help
Project Explorer
kernel 2.6.30.2
C-Build [kernel-2.6.30.2]
using /usr/local/src/linux-2.6.30.2 as source for kernel
GEN /mnt/build/linux-2.6/Makefile
CHK include/linux/version.h
CHK include/linux/utsrelease.h
SYMLINK include/asm -> include/asm-x86
CALL /usr/local/src/linux-2.6.30.2/scripts/checksyscalls.sh
CHK include/linux/compile.h
CHK include/linux/version.h
make[4]: `scripts/unifdef' is up to date.
CHECK include/ntd (6 files)
Kernel: arch/x86/boot/bzImage is ready (#1)
Building modules, stage 2.
MODPOST 1917 modules
```

9. QEMU Installation

- Install QEMU & Supporting software
- From Fedora 11 CD/DVD, install in order (or yum):
 - qemu-common-0.10-16.fc11.i586.rpm
 - qemu-img-0.10-16.fc11.i586.rpm
 - bochs-bios-2.3.8-0.6.git04387139e3b.fc11.noarch.rpm
 - etherboot-zroms-kvm-5.4.4-13.fc11.noarch.rpm
 - vgabios-0.6-0.5.b.fc11.noarch.rpm
 - qemu-system-x86-0.10-16.fc11.i586.rpm
 - qemu-kvm-0.10-16.fc11.i586.rpm (*optional*)
- Note: If you already installed, skip this.

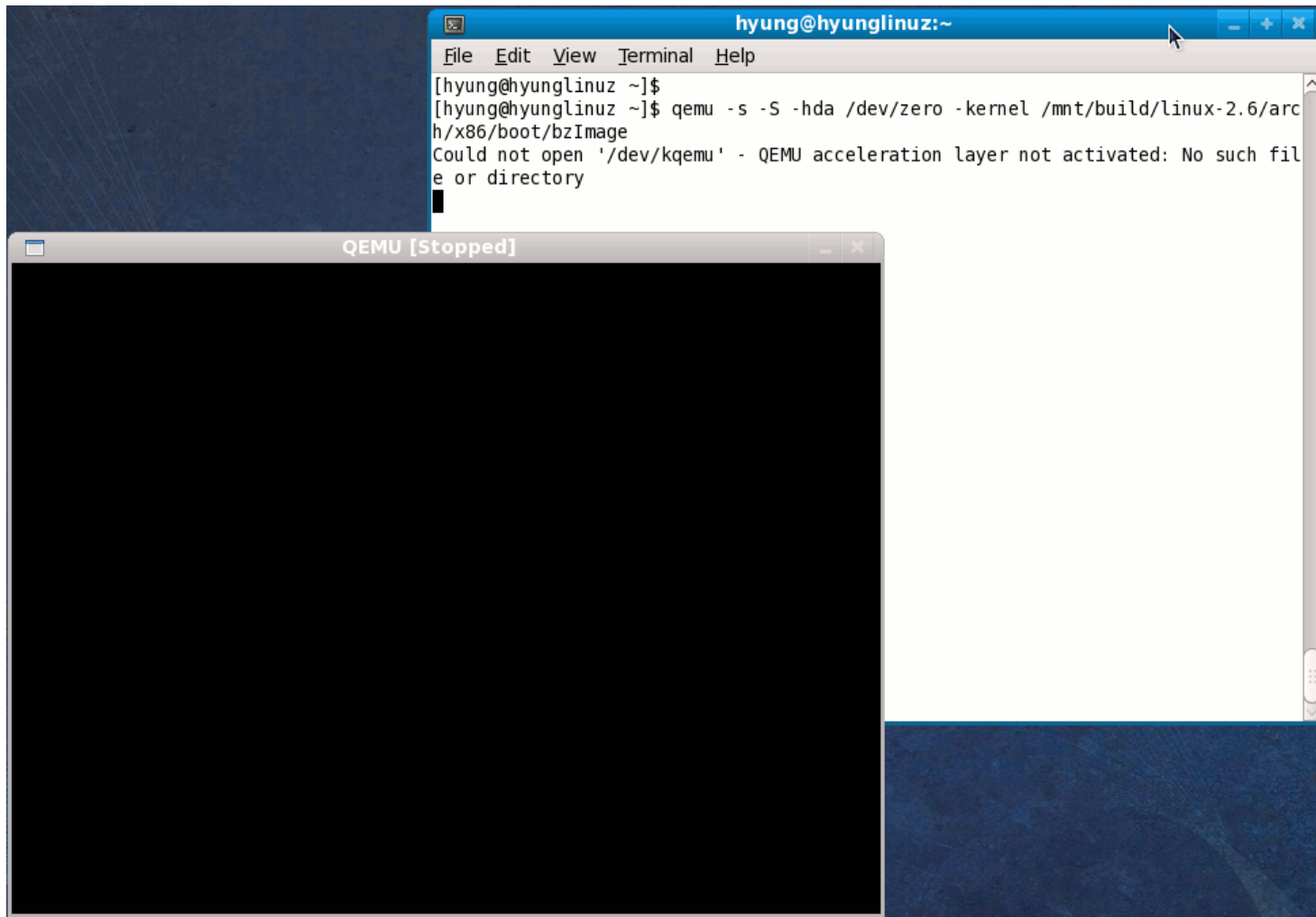
10. First Run QEMU

- In a terminal/shell

```
$ qemu -s -S -hda /dev/zero -kernel  
/mnt/build/linux-2.6/arch/x86/boot/bzImage
```

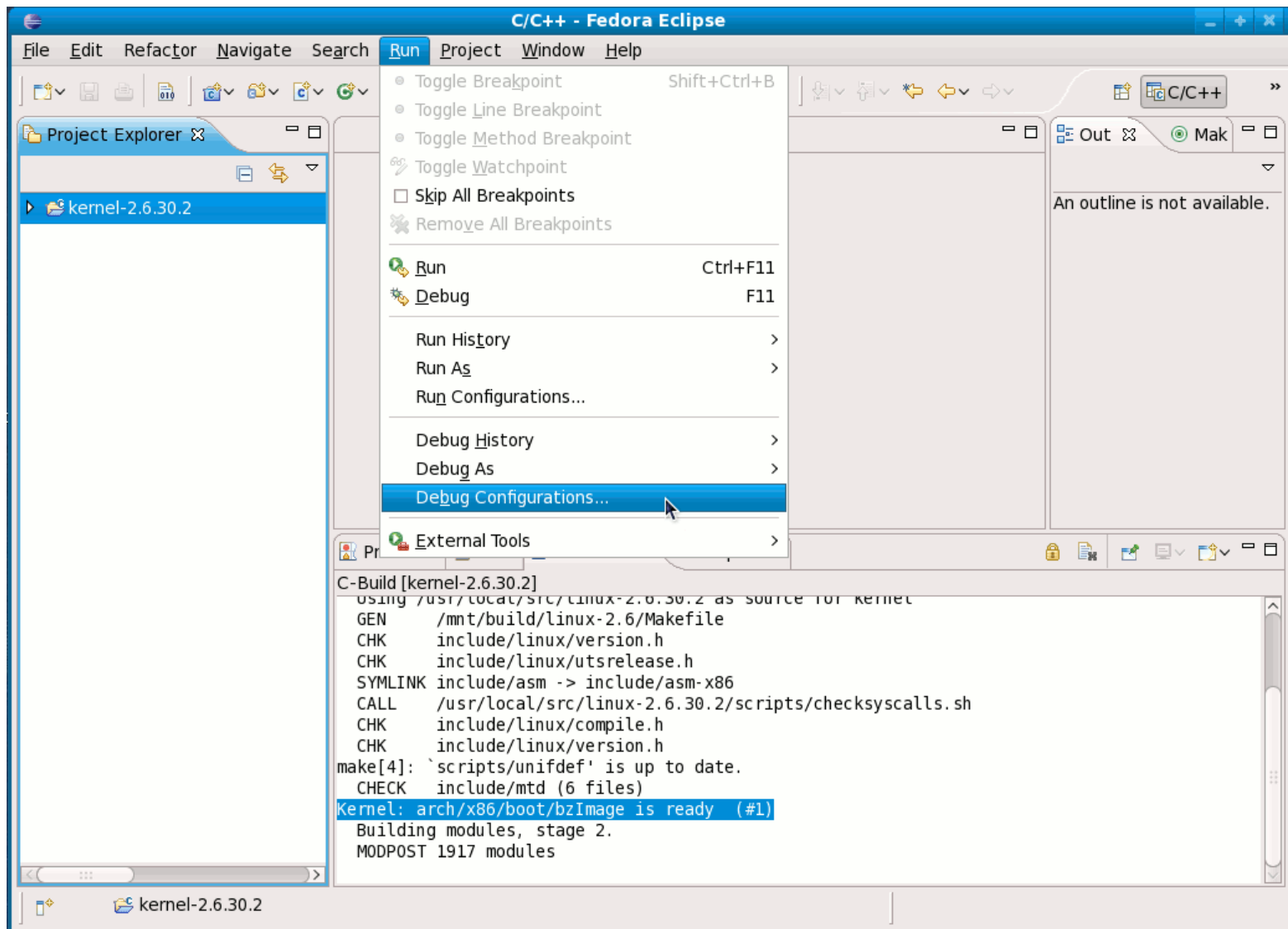

10. First Run QEMU

- Shows empty (blank) screen
 - Leave this QEMU screen



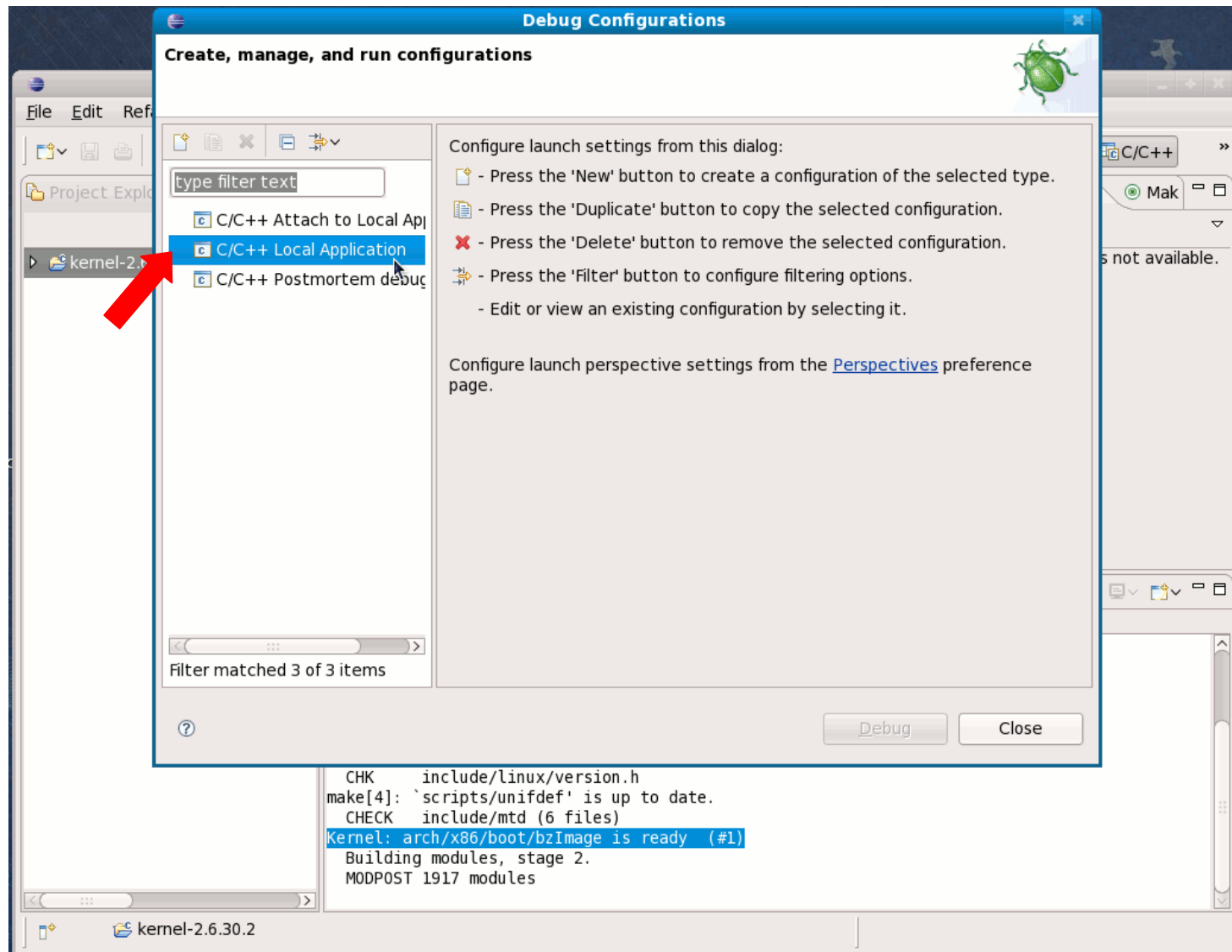
11. Eclipse Debug Configurations

- “Run→ Debug Configurations..”



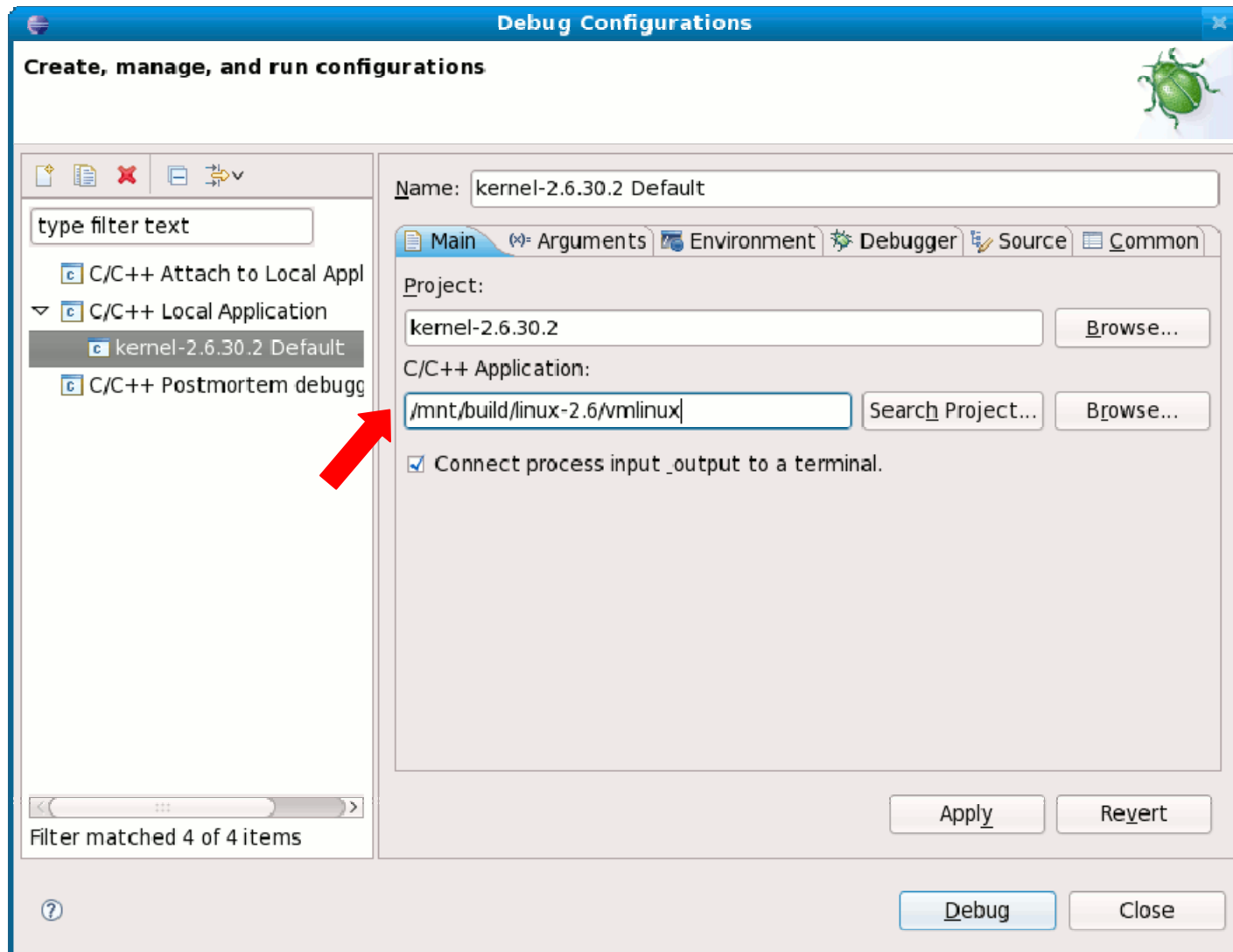
11. Eclipse Debug Configurations

- Double click “C/C++ Local Application”



11. Eclipse Debug Configurations

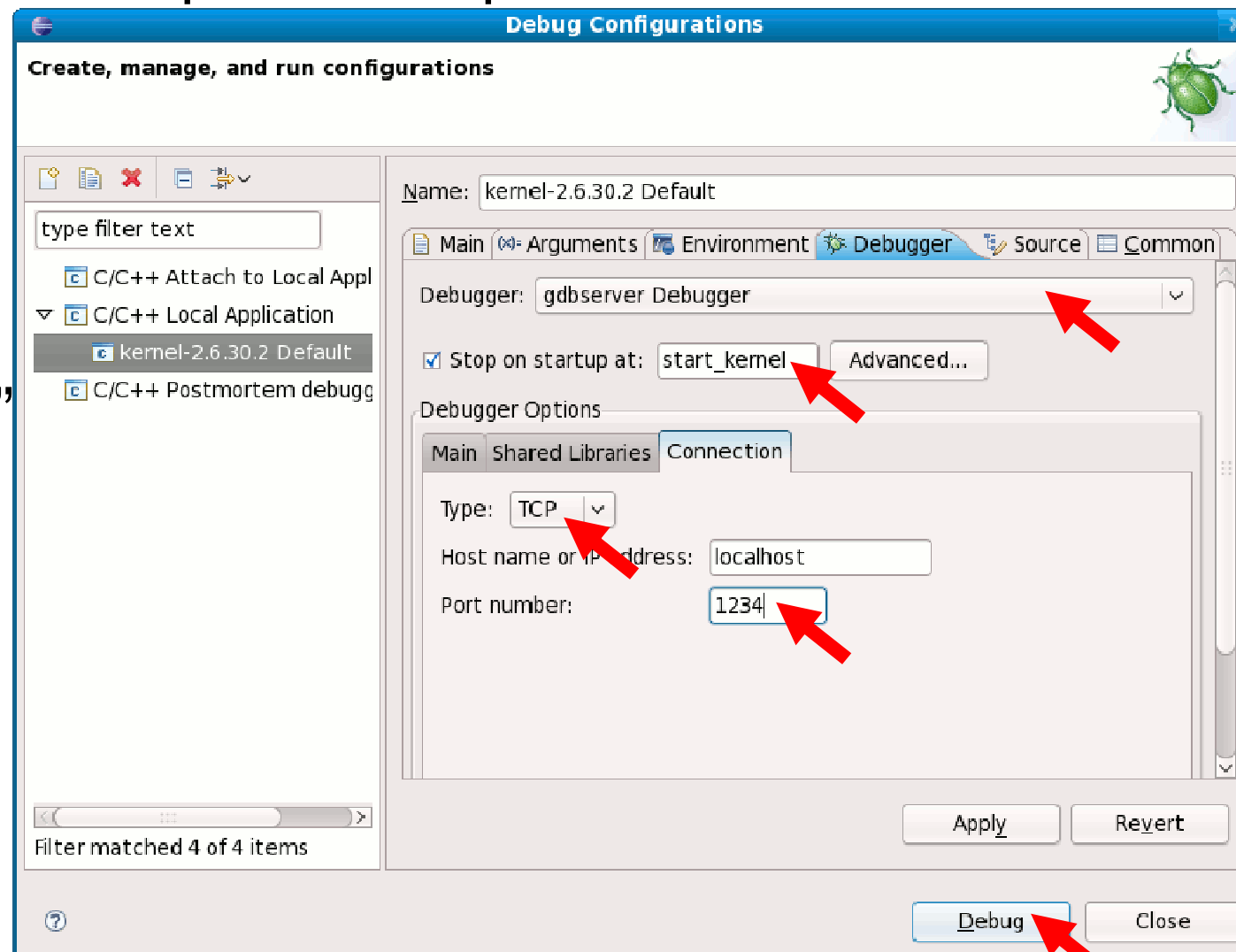
- Put “/mnt/build/linux-2.6/vmlinux” in “C/C++ Application”



11. Eclipse Debug Configurations

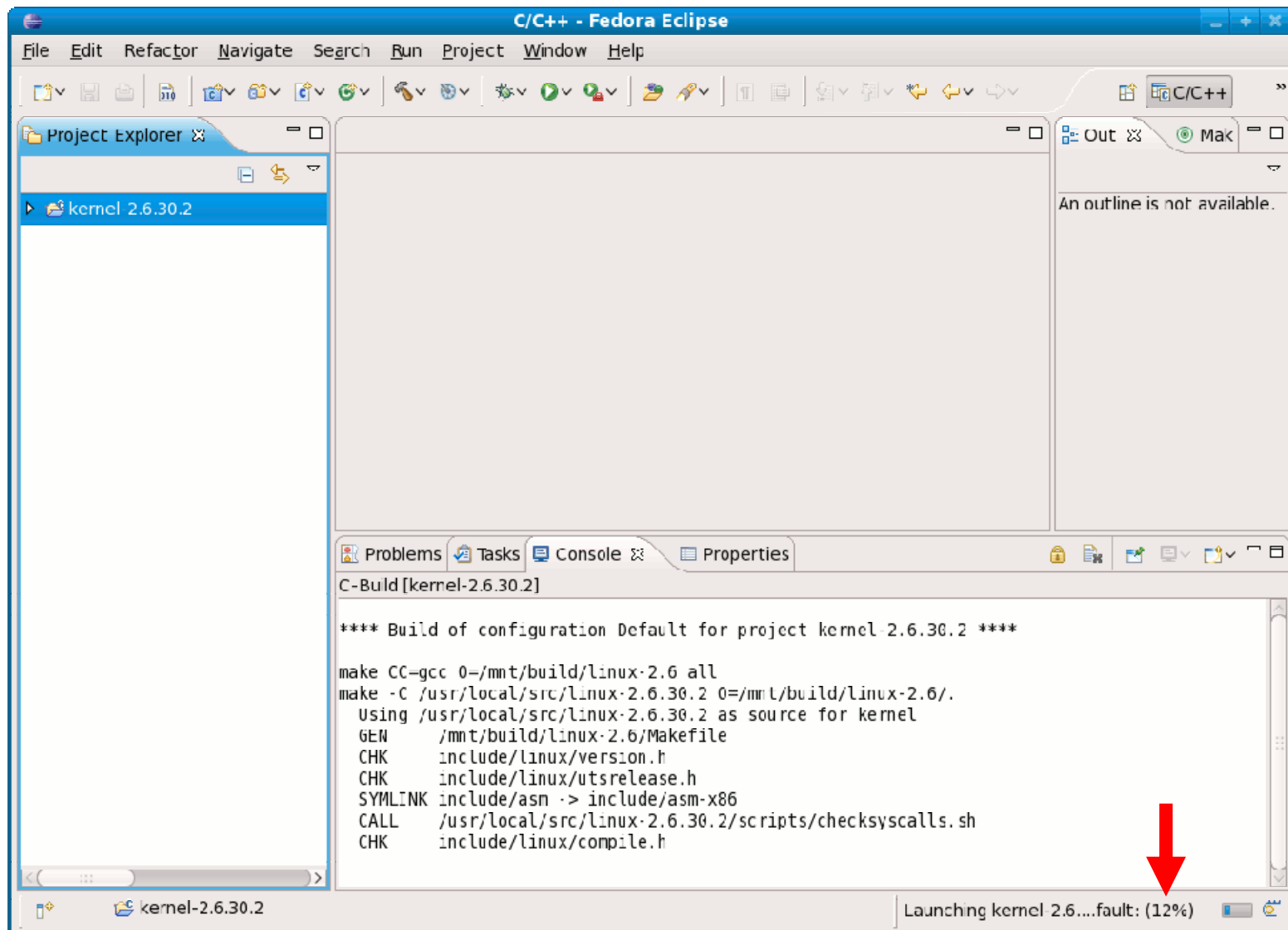
- In “Debugger” tab
 - “gdbserver Debugger” in “Debugger:”
 - “start_kernel” in “Stop on startup at:”

- “Connection”
 - Select “TCP” in “Type:” list
 - Put “1234” in “Port number”
- Click “Debug”



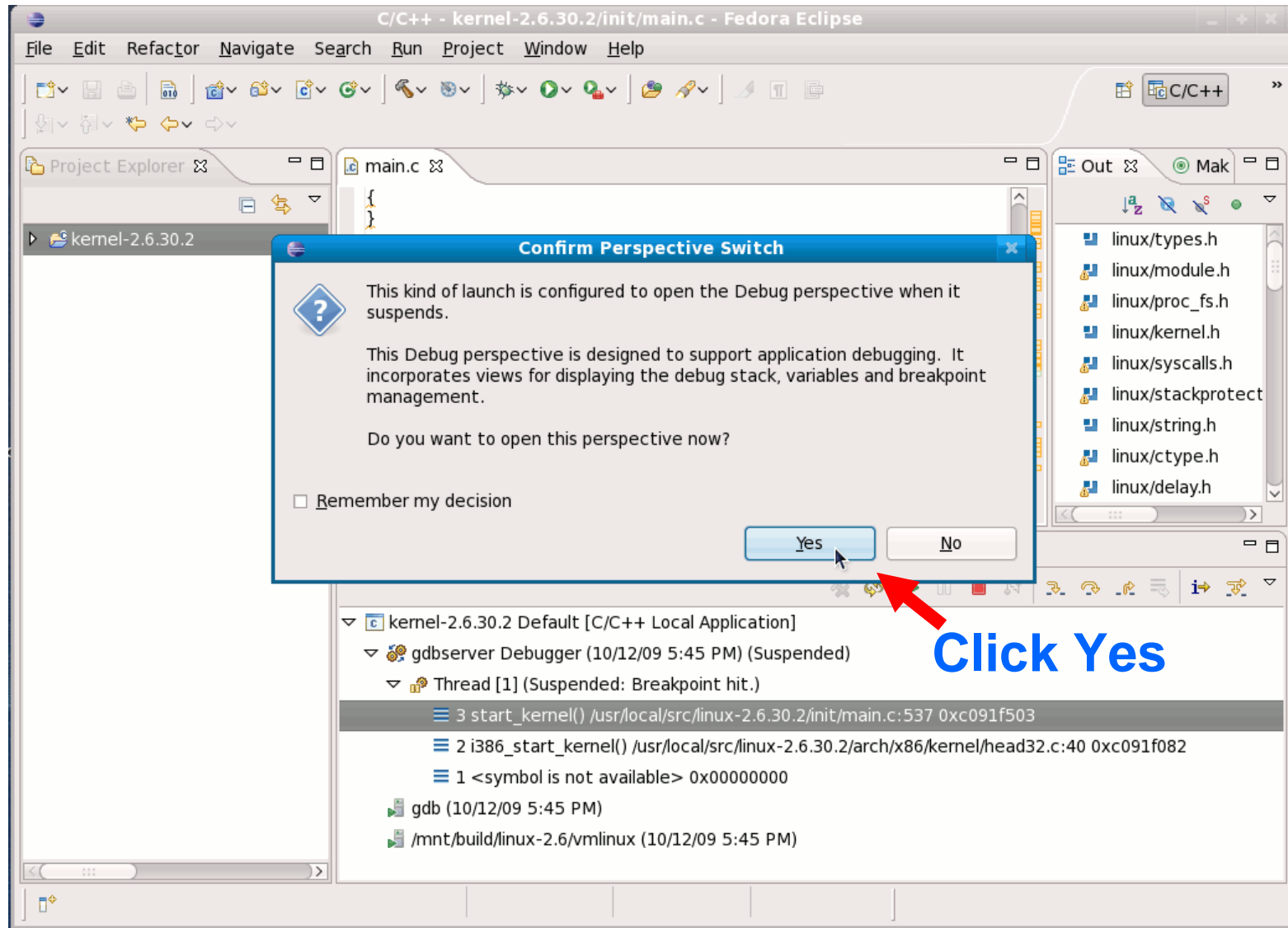
11. Eclipse Debug Configurations

- Eclipse compiles and links in progress



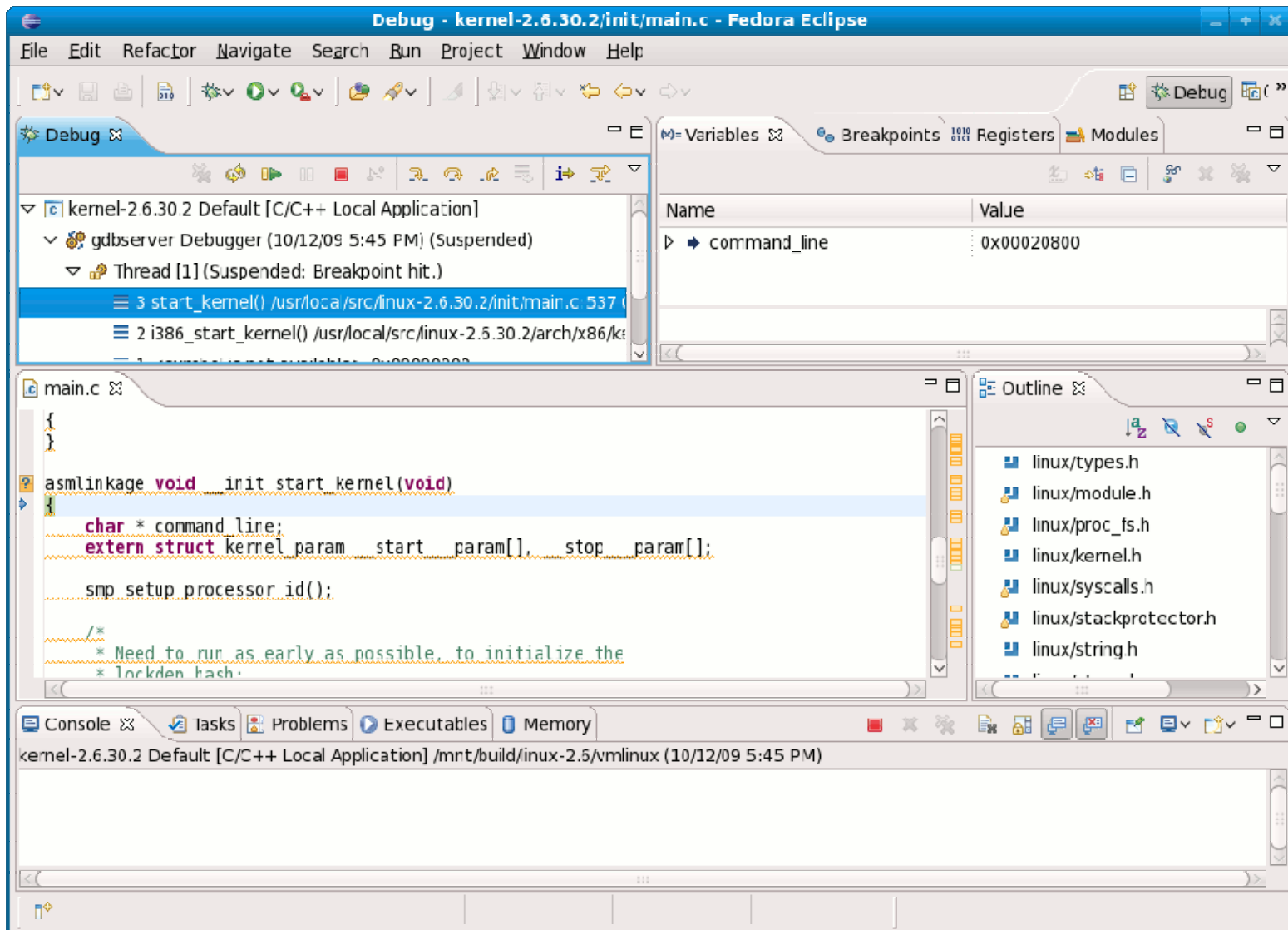
11. Eclipse Debug Configurations

- After a while, it opens “Confirm Perspective Switch”



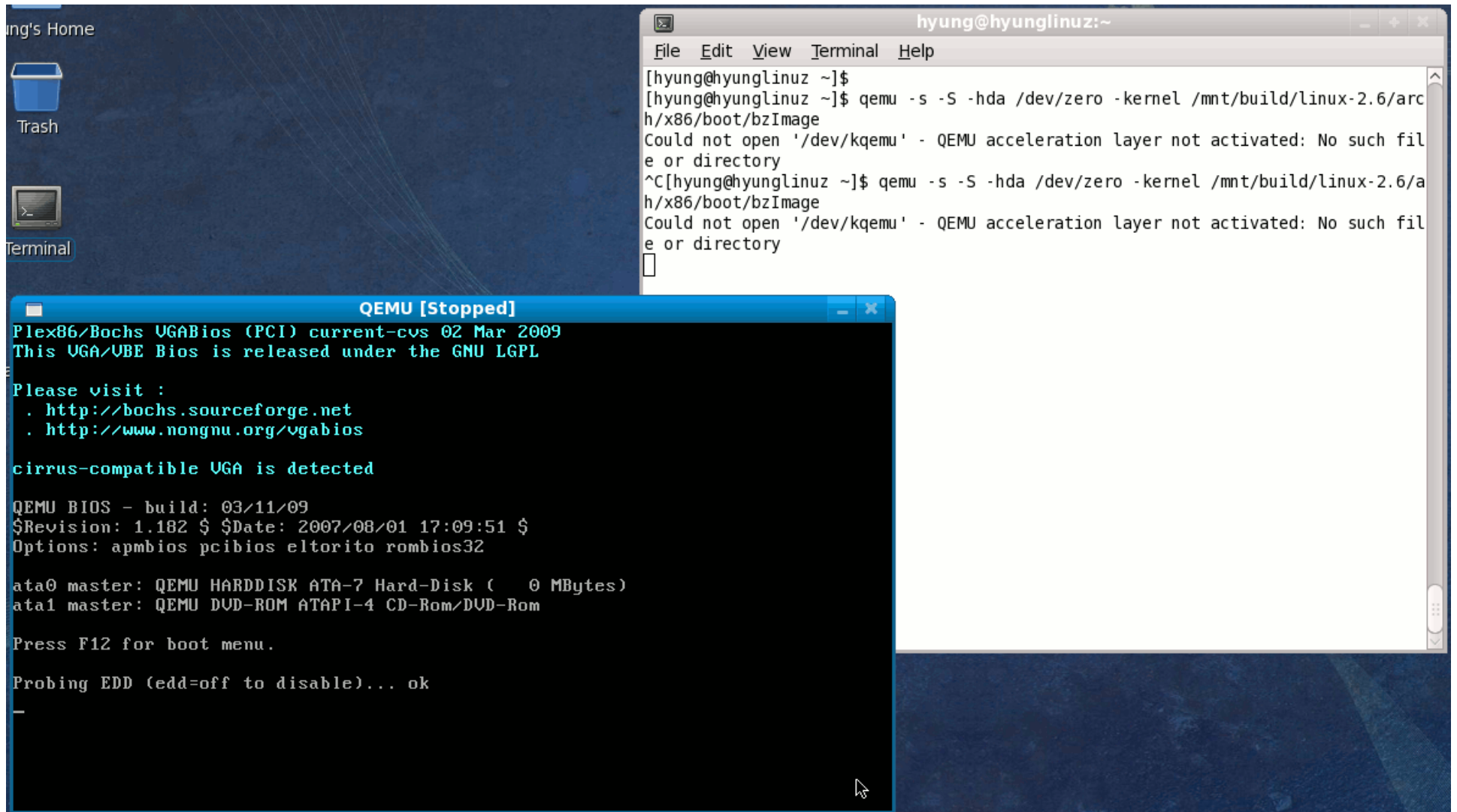
11. Eclipse Debug Configurations

- Shows the changed perspective



12. Back to QEMU screen

- Shows *some* outputs.



The screenshot shows a Linux desktop environment with a dark blue background. In the top-left corner, there is a 'Terminal' icon. A terminal window titled 'hyung@hyunglinux:~' is open, displaying the following text:

```
File Edit View Terminal Help
[hyung@hyunglinux ~]$
[hyung@hyunglinux ~]$ qemu -s -S -hda /dev/zero -kernel /mnt/build/linux-2.6/arch/x86/boot/bzImage
Could not open '/dev/kqemu' - QEMU acceleration layer not activated: No such file or directory
^C[hyung@hyunglinux ~]$ qemu -s -S -hda /dev/zero -kernel /mnt/build/linux-2.6/arch/x86/boot/bzImage
Could not open '/dev/kqemu' - QEMU acceleration layer not activated: No such file or directory

```

In the foreground, a window titled 'QEMU [Stopped]' is open, displaying the following text:

```
Plex86/Bochs UGABios (PCI) current-cvs 02 Mar 2009
This UGA/UBE Bios is released under the GNU LGPL

Please visit :
. http://bochs.sourceforge.net
. http://www.nongnu.org/ugabios

cirrus-compatible UGA is detected

QEMU BIOS - build: 03/11/09
$Revision: 1.182 $ $Date: 2007/08/01 17:09:51 $
Options: apmbios pcibios eltorito rombios32

ata0 master: QEMU HARDDISK ATA-7 Hard-Disk ( 0 MBytes)
ata1 master: QEMU DVD-ROM ATAPI-4 CD-Rom/DVD-Rom

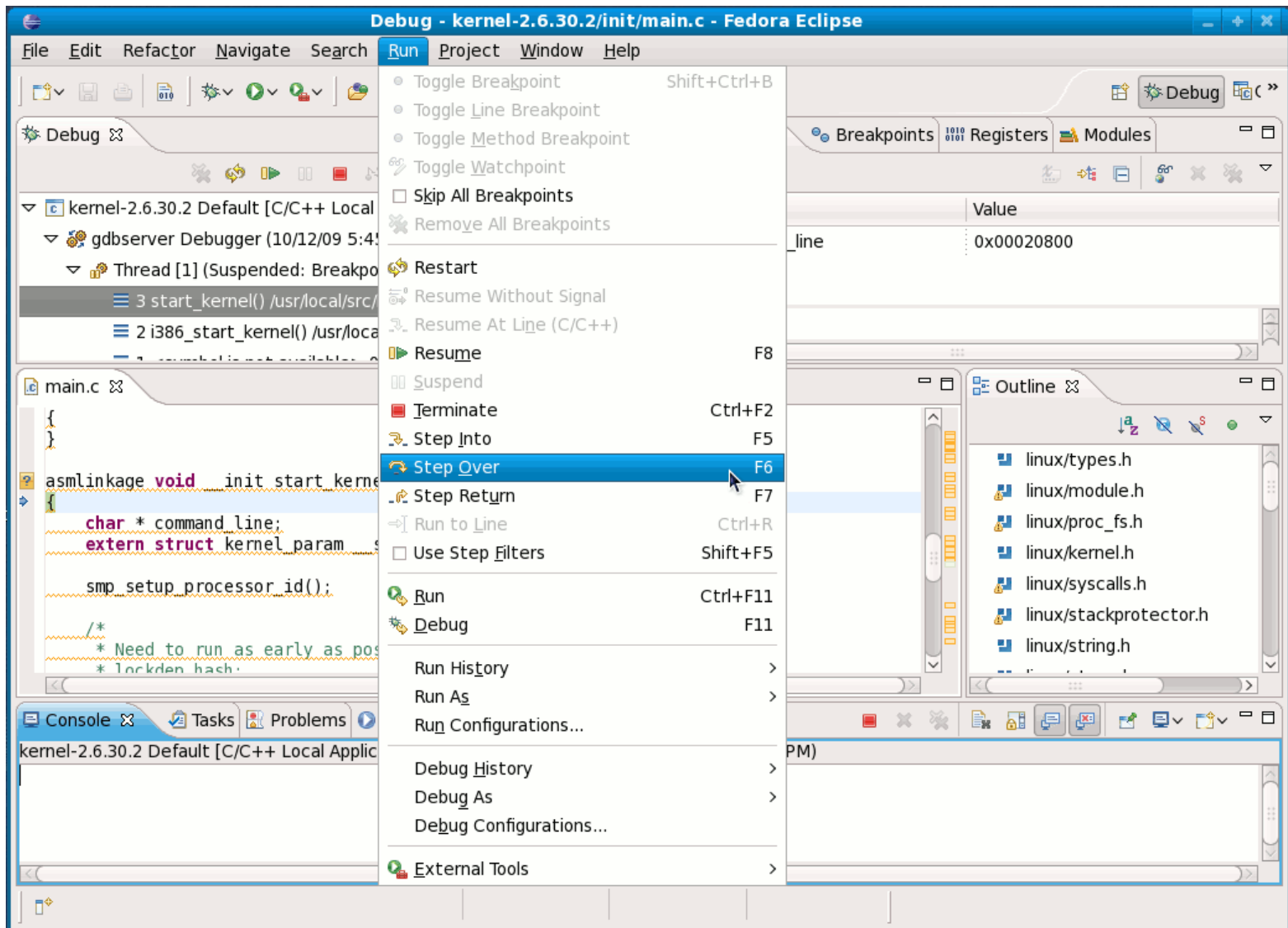
Press F12 for boot menu.

Probing EDD (edd=off to disable)... ok

```

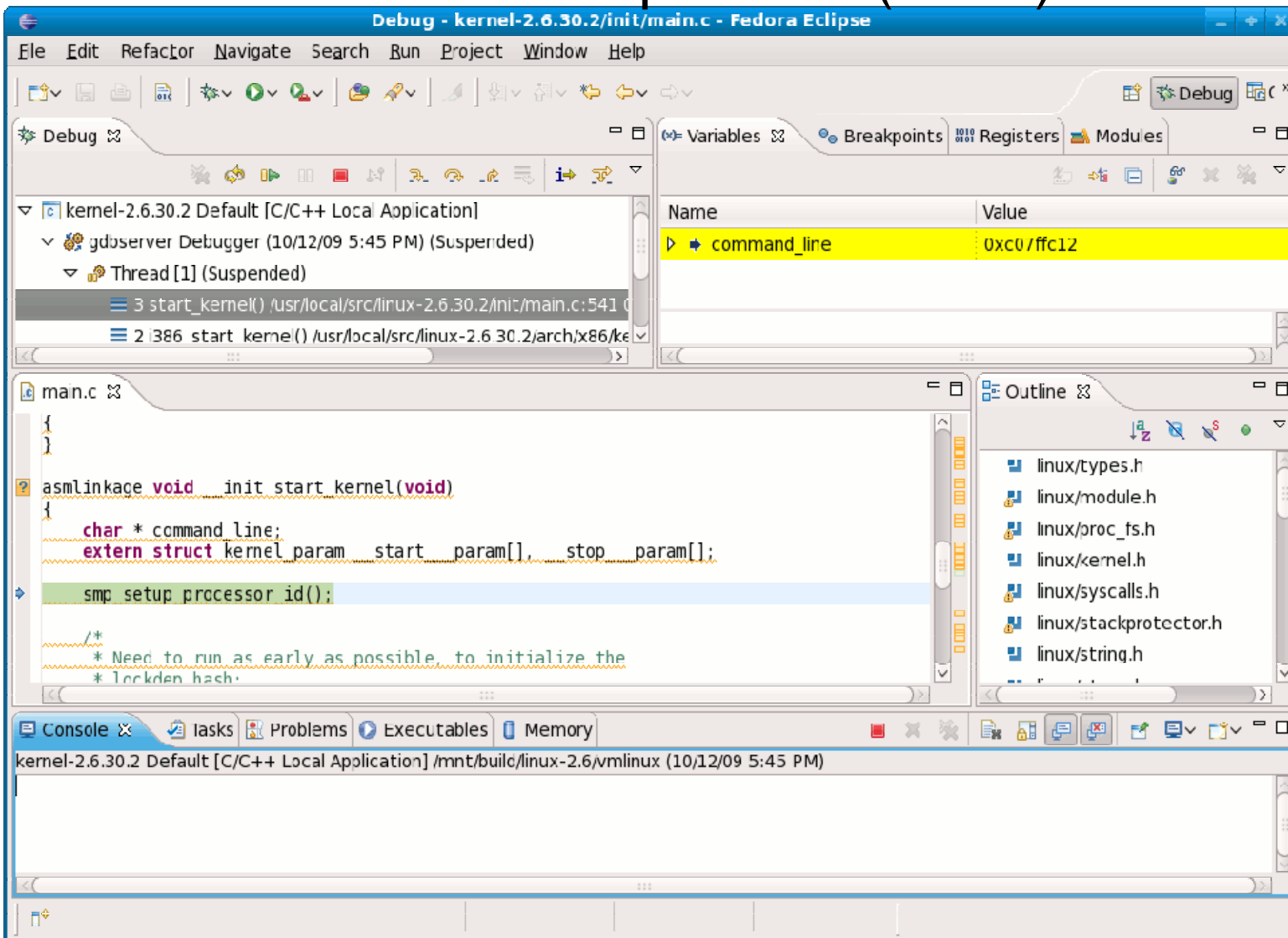
13. Line-by-line run in Eclipse

- In Eclipse, “Run→ Step over” (or F6)



13. Line-by-line run in Eclipse

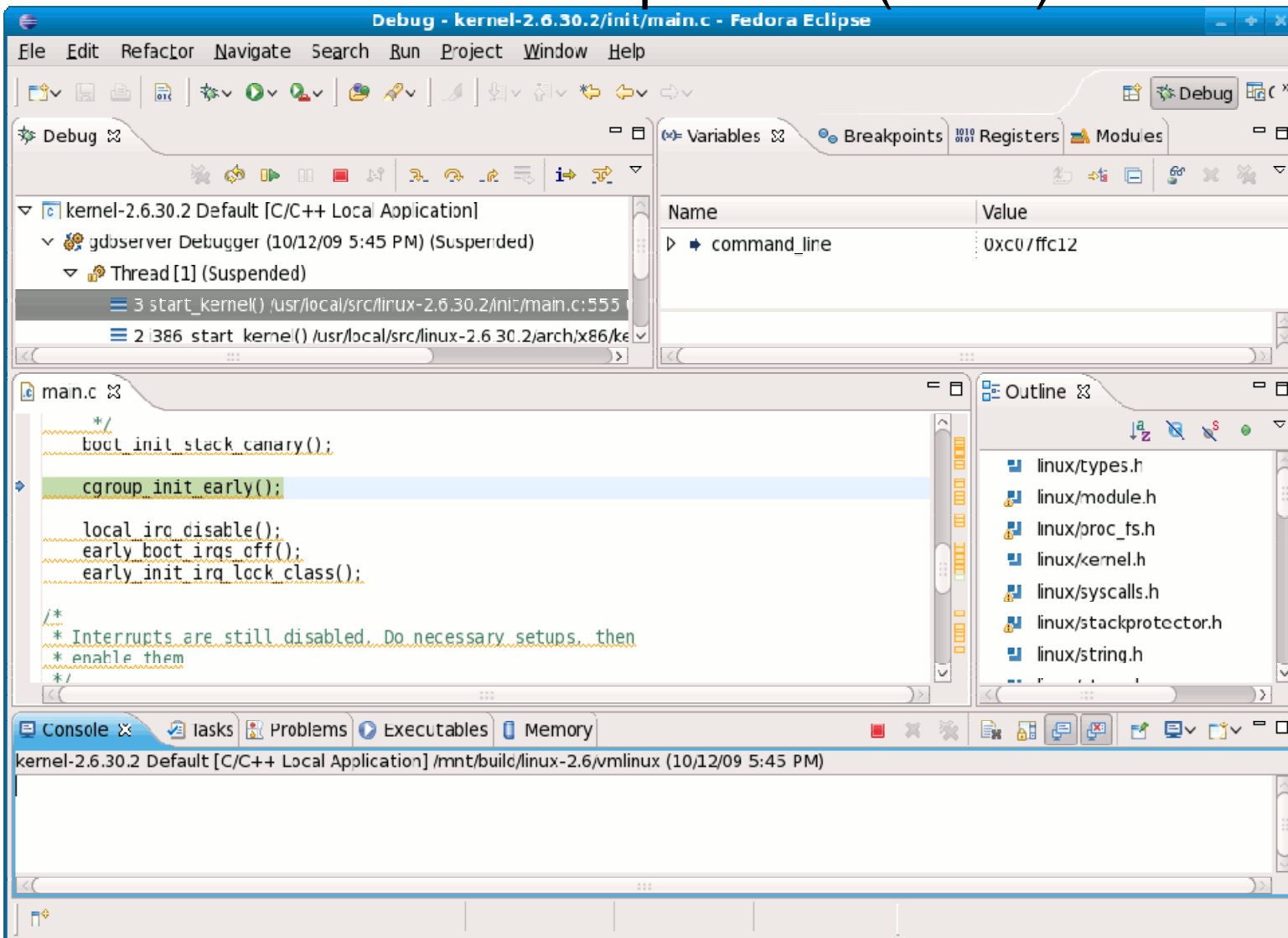
- In Eclipse, “Run → Step over” (or F6)
 - Several “Run → Step over”s (or F6)



Outputs on
QEMU screen

13. Line-by-line run in Eclipse

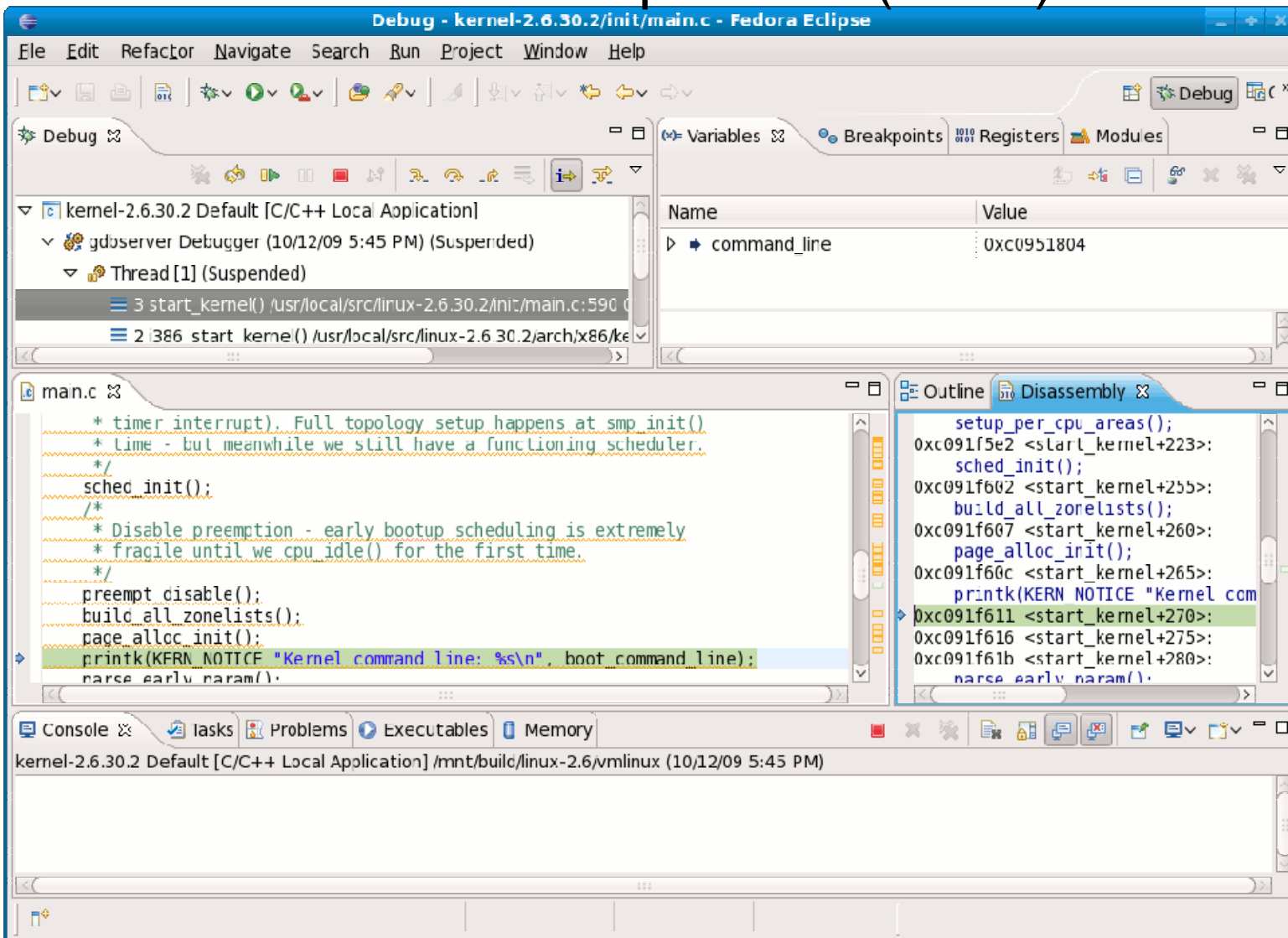
- In Eclipse, “Run → Step over” (or F6)
 - Several “Run → Step over”s (or F6)



Outputs on
QEMU screen

13. Line-by-line run in Eclipse

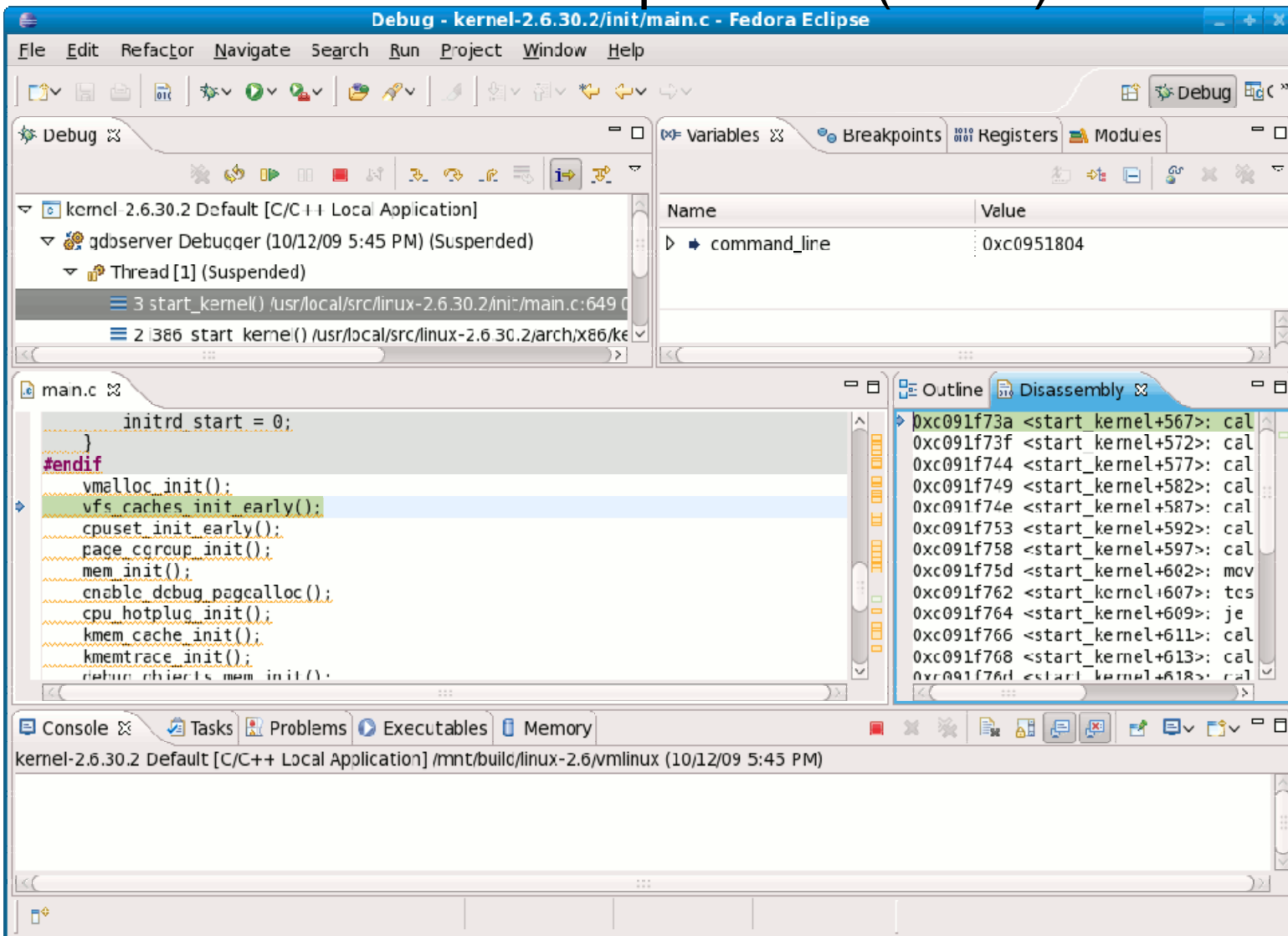
- In Eclipse, “Run → Step over” (or F6)
 - Several “Run → Step over”s (or F6)



Outputs on
QEMU screen

13. Line-by-line run in Eclipse

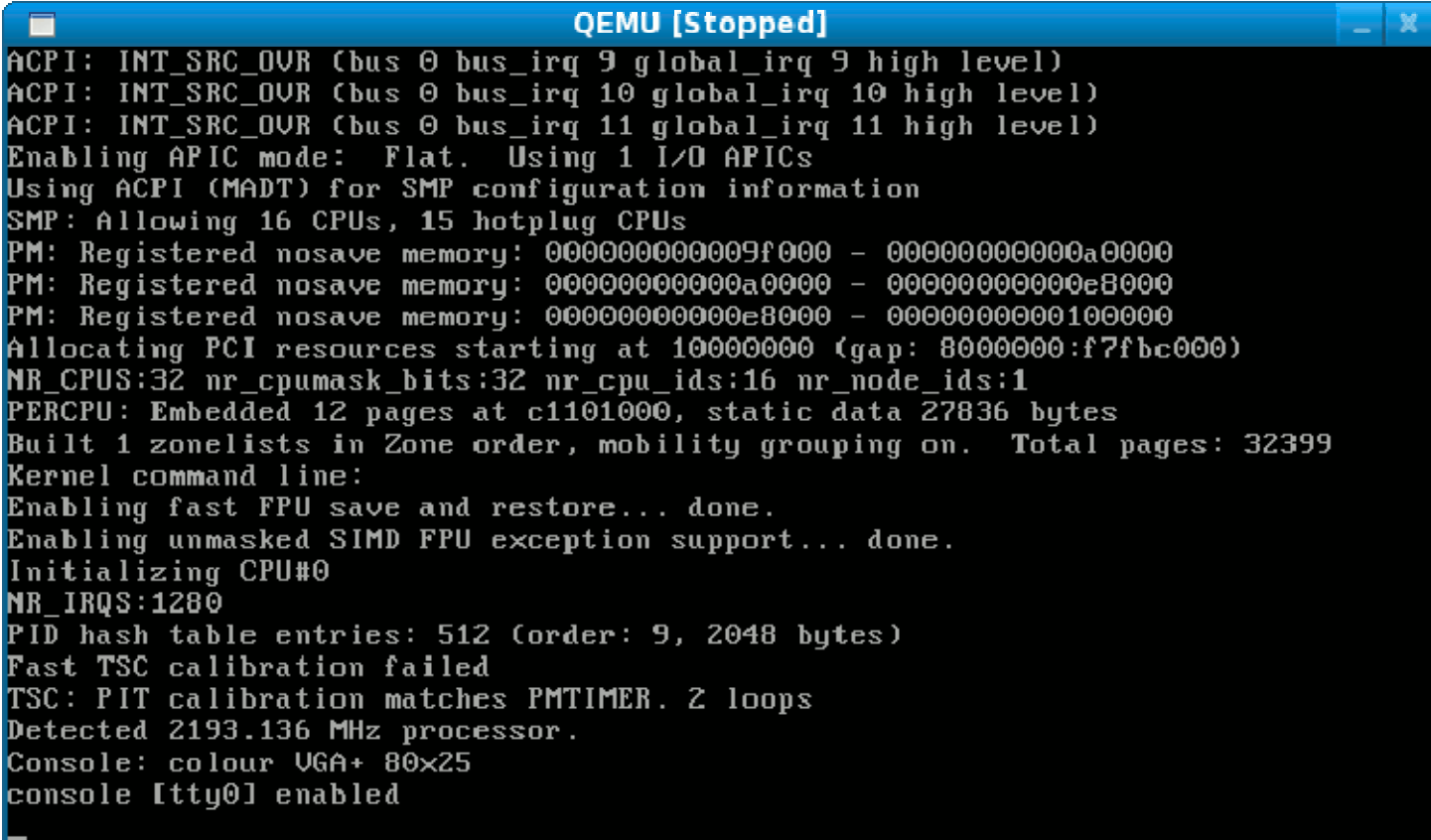
- In Eclipse, “Run → Step over” (or F6)
 - Several “Run → Step over”s (or F6)



Outputs on
QEMU screen

13. Line-by-line run in Eclipse

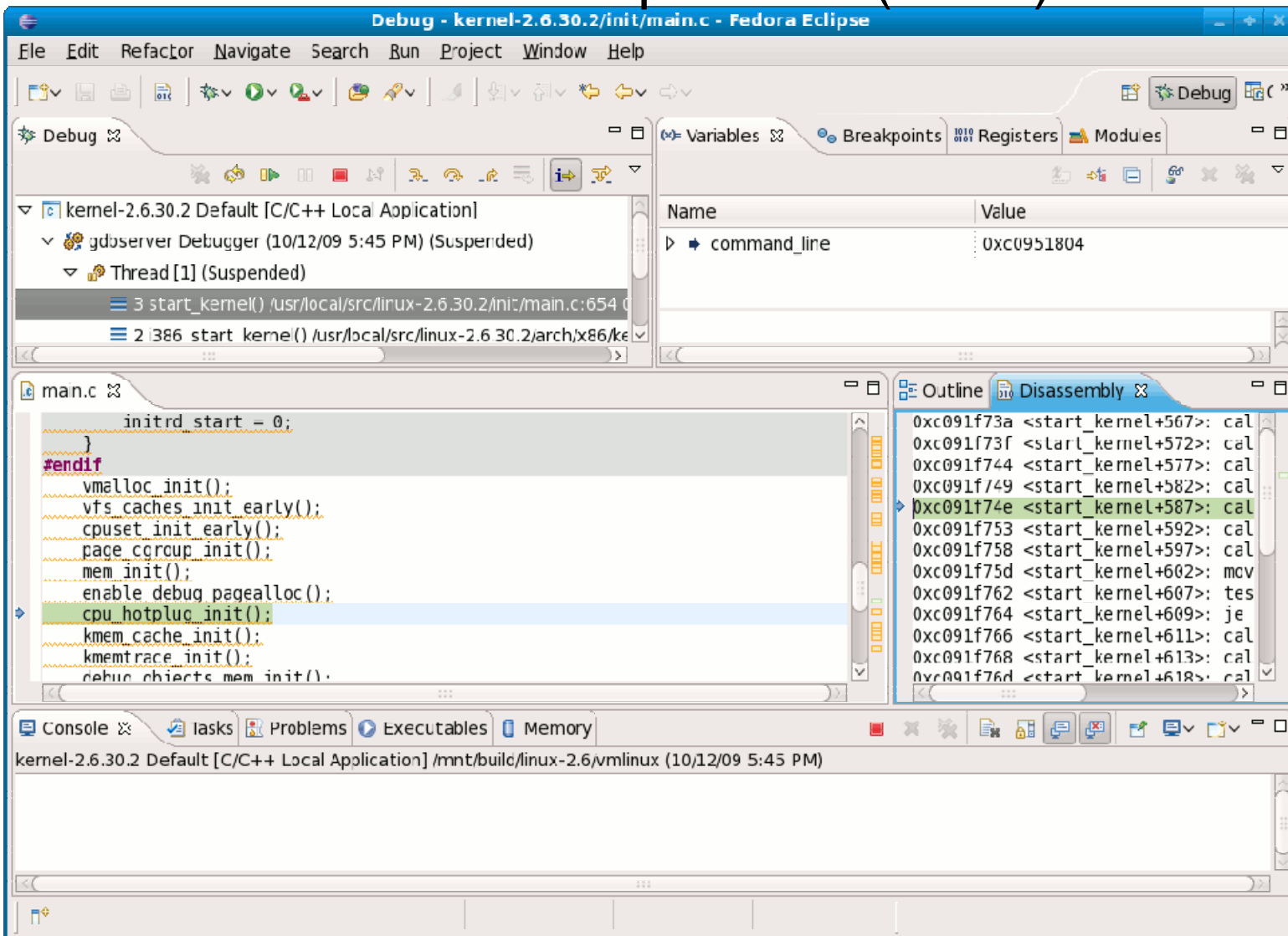
- In Eclipse, “Run → Step over” (or F6)
 - Several “Run → Step over”s (or F6)



```
QEMU [Stopped]
ACPI: INT_SRC_OVR (bus 0 bus_irq 9 global_irq 9 high level)
ACPI: INT_SRC_OVR (bus 0 bus_irq 10 global_irq 10 high level)
ACPI: INT_SRC_OVR (bus 0 bus_irq 11 global_irq 11 high level)
Enabling APIC mode: Flat. Using 1 I/O APICs
Using ACPI (MADT) for SMP configuration information
SMP: Allowing 16 CPUs, 15 hotplug CPUs
PM: Registered nosave memory: 0000000000009f000 - 000000000000a0000
PM: Registered nosave memory: 000000000000a0000 - 000000000000e8000
PM: Registered nosave memory: 000000000000e8000 - 00000000000100000
Allocating PCI resources starting at 10000000 (gap: 8000000:f7fbc000)
NR_CPUS:32 nr_cpumask_bits:32 nr_cpu_ids:16 nr_node_ids:1
PERCPU: Embedded 12 pages at c1101000, static data 27836 bytes
Built 1 zonelists in Zone order, mobility grouping on. Total pages: 32399
Kernel command line:
Enabling fast FPU save and restore... done.
Enabling unmasked SIMD FPU exception support... done.
Initializing CPU#0
NR_IRQS:1280
PID hash table entries: 512 (order: 9, 2048 bytes)
Fast TSC calibration failed
TSC: PIT calibration matches PMTIMER. 2 loops
Detected 2193.136 MHz processor.
Console: colour UGA+ 80x25
console [tty0] enabled
```

13. Line-by-line run in Eclipse

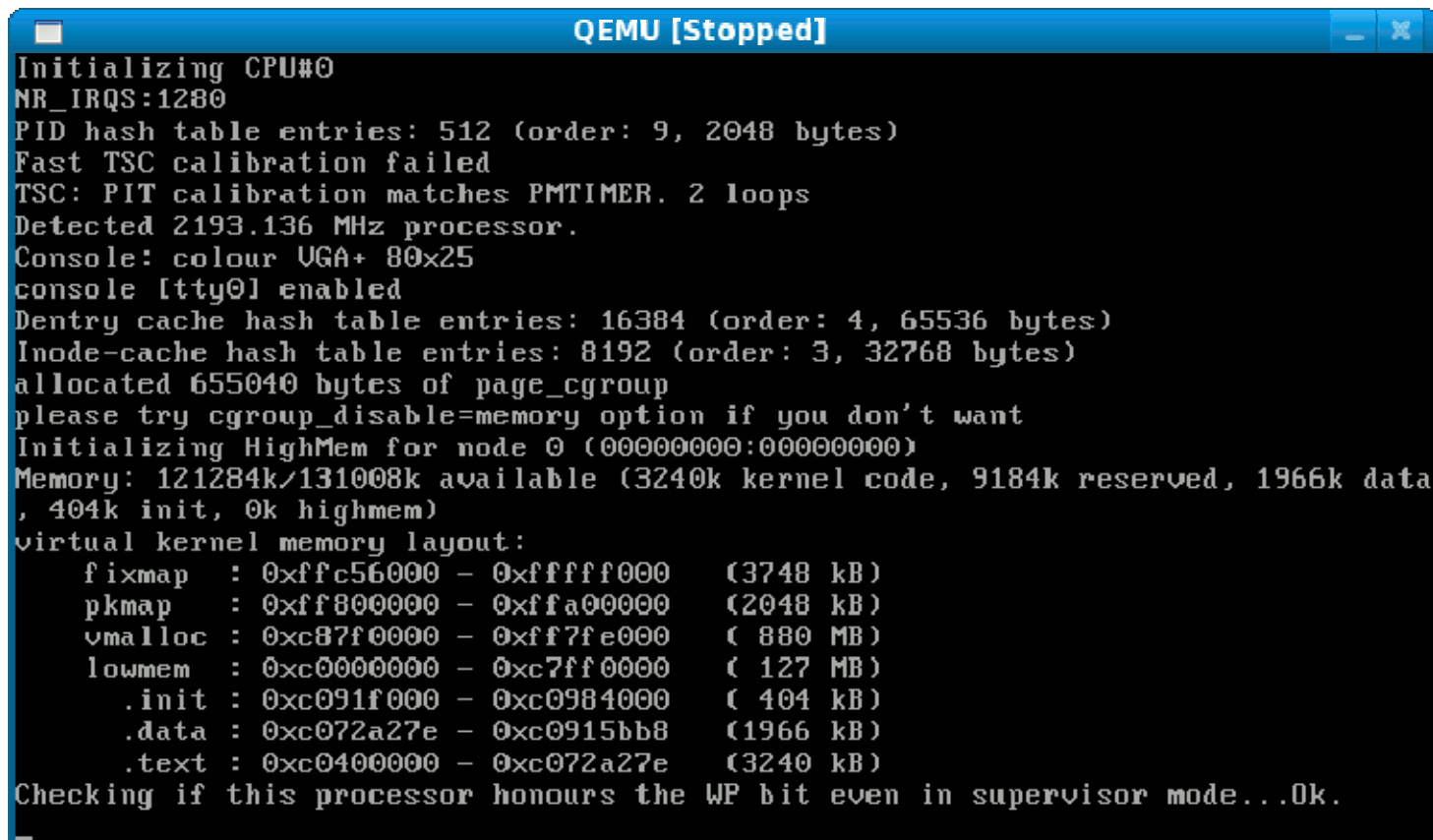
- In Eclipse, “Run → Step over” (or F6)
 - Several “Run → Step over”s (or F6)



Outputs on
QEMU screen

13. Line-by-line run in Eclipse

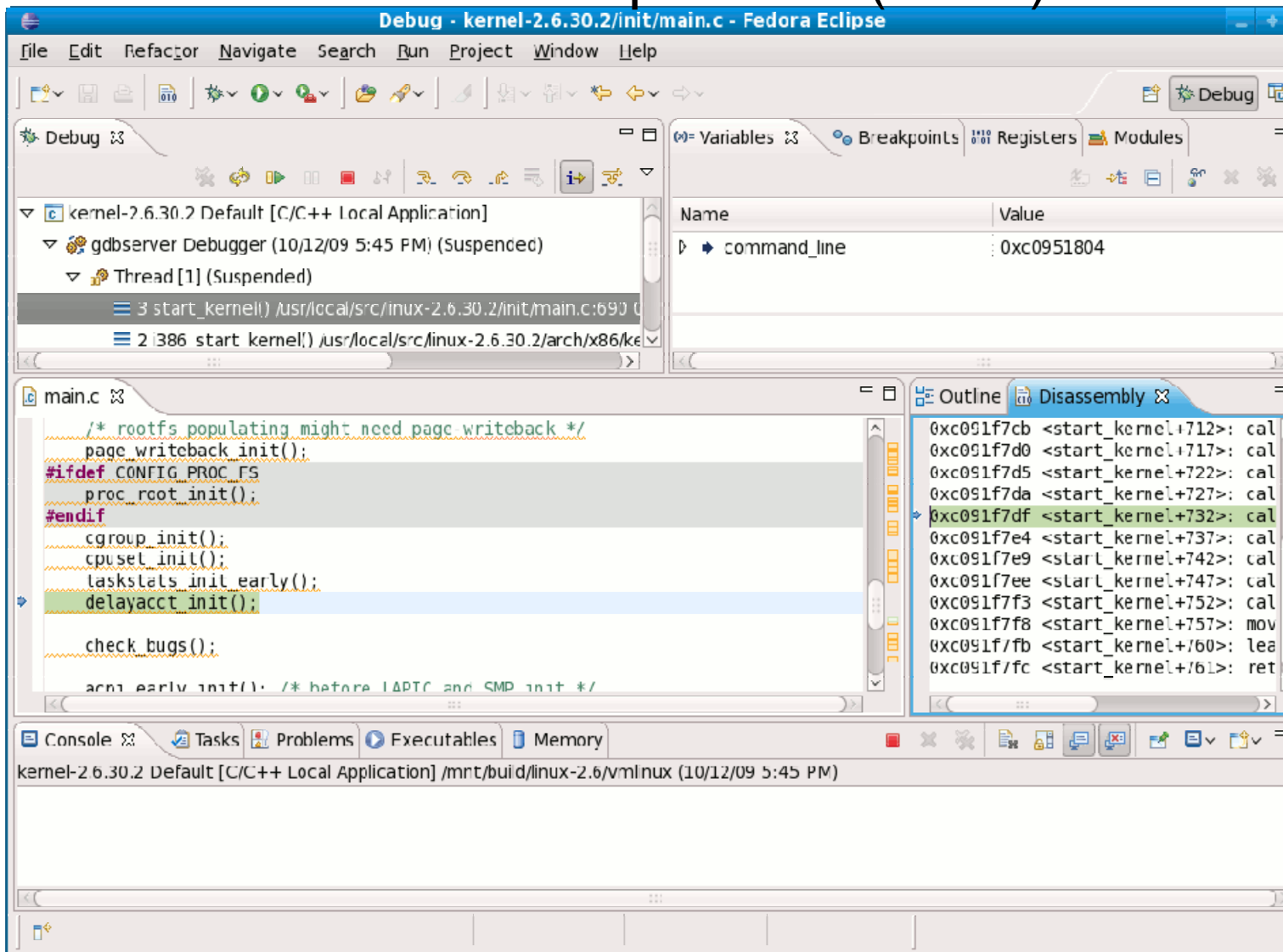
- In Eclipse, “Run → Step over” (or F6)
 - Several “Run → Step over”s (or F6)



```
QEMU [Stopped]
Initializing CPU#0
NR_IRQS:1280
PID hash table entries: 512 (order: 9, 2048 bytes)
Fast TSC calibration failed
TSC: PIT calibration matches PMTIMER. 2 loops
Detected 2193.136 MHz processor.
Console: colour UGA+ 80x25
console [tty0] enabled
Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)
Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)
allocated 655040 bytes of page_cgroup
please try cgroup_disable=memory option if you don't want
Initializing HighMem for node 0 (00000000:00000000)
Memory: 121284k/131008k available (3240k kernel code, 9184k reserved, 1966k data
, 404k init, 0k highmem)
virtual kernel memory layout:
   fixmap  : 0xffc56000 - 0xfffff000   (3748 kB)
   pkmap   : 0xff800000 - 0xffa00000   (2048 kB)
   vmalloc : 0xc87f0000 - 0xff7fe000   ( 880 MB)
   lowmem  : 0xc0000000 - 0xc7ff0000   ( 127 MB)
     .init  : 0xc091f000 - 0xc0984000   ( 404 kB)
     .data  : 0xc072a27e - 0xc0915bb8   (1966 kB)
     .text  : 0xc0400000 - 0xc072a27e   (3240 kB)
Checking if this processor honours the WP bit even in supervisor mode...Ok.
```

13. Line-by-line run in Eclipse

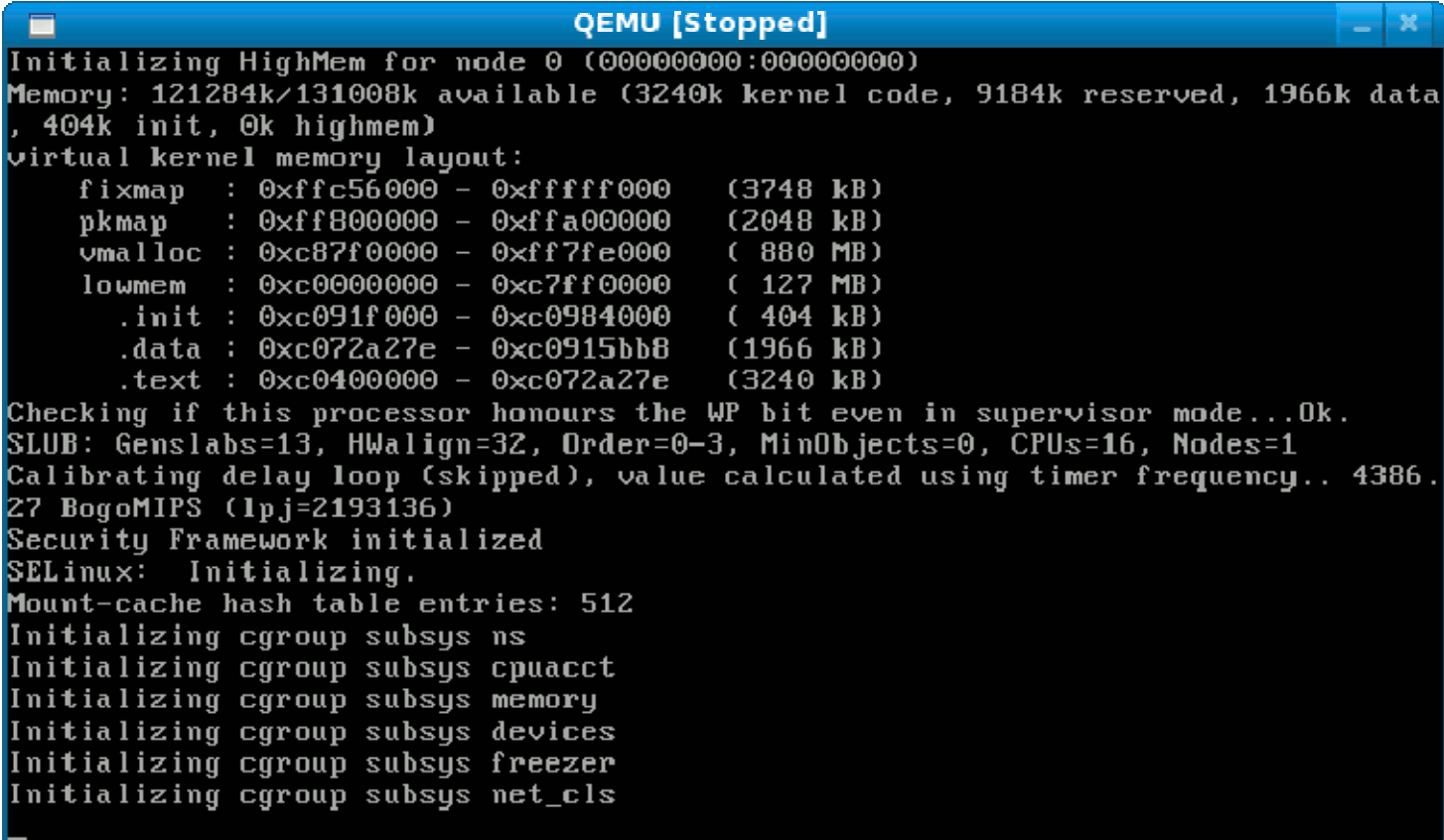
- In Eclipse, “Run → Step over” (or F6)
 - Several “Run → Step over”s (or F6)



Outputs on
QEMU screen

13. Line-by-line run in Eclipse

- In Eclipse, “Run → Step over” (or F6)
 - Several “Run → Step over”s (or F6)

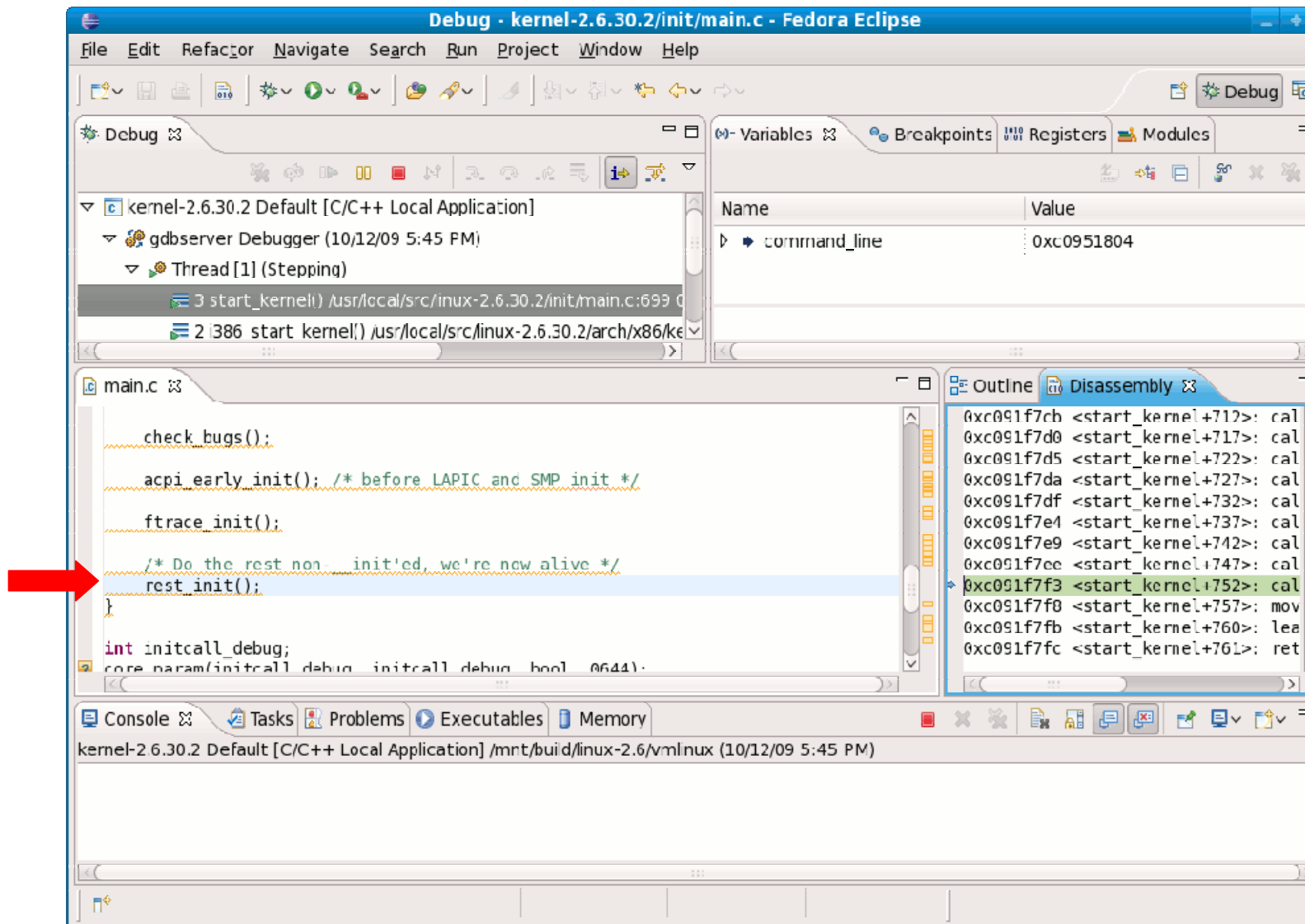


```
QEMU [Stopped]
Initializing HighMem for node 0 (00000000:00000000)
Memory: 121284k/131008k available (3240k kernel code, 9184k reserved, 1966k data
, 404k init, 0k highmem)
virtual kernel memory layout:
  fixmap   : 0xffc56000 - 0xfffff000   (3748 kB)
  pkmap   : 0xff800000 - 0xffa00000   (2048 kB)
  vmalloc : 0xc87f0000 - 0xff7fe000   ( 880 MB)
  lowmem  : 0xc0000000 - 0xc7ff0000   ( 127 MB)
  .init   : 0xc091f000 - 0xc0984000   ( 404 kB)
  .data   : 0xc072a27e - 0xc0915bb8   (1966 kB)
  .text   : 0xc0400000 - 0xc072a27e   (3240 kB)
Checking if this processor honours the WP bit even in supervisor mode...Ok.
SLUB: Genslabs=13, HWalig=32, Order=0-3, MinObjects=0, CPUs=16, Nodes=1
Calibrating delay loop (skipped), value calculated using timer frequency.. 4386.
27 BogoMIPS (lpj=2193136)
Security Framework initialized
SELinux: Initializing.
Mount-cache hash table entries: 512
Initializing cgroup subsys ns
Initializing cgroup subsys cpuacct
Initializing cgroup subsys memory
Initializing cgroup subsys devices
Initializing cgroup subsys freezer
Initializing cgroup subsys net_cls
```

14. Final QEMU screen

- After *rest_init()* run, QEMU console shows **kernel panic**.
 - Since it doesn't have a rootfile system
 - /dev/zero was assigned in the initial run.
- Can add a rootfile system later.

14. Final QEMU screen



14. Final QEMU screen

```
QEMU
sr0: scsi3-mmc drive: 4x/4x xa/form2 tray
Uniform CD-ROM driver Revision: 3.20
sr 1:0:0:0: Attached scsi generic sg1 type 5
md: Waiting for all devices to be available before autodetect
md: If you don't use raid, use raid=noautodetect
md: Autodetecting RAID arrays.
md: Scanned 0 and added 0 devices.
md: autorun ...
md: ... autorun DONE.
UFS: Cannot open root device "<NULL>" or unknown-block(8,6)
Please append a correct "root=" boot option; here are the available partitions:
0b00          1048575 sr0 driver: sr
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(8,6)
Pid: 1, comm: swapper Not tainted 2.6.30.2 #1
Call Trace:
[<c07246cb>] ? printk+0x14/0x19
[<c0724610>] panic+0x3e/0xe5
[<c091fbbf>] mount_block_root+0x1e6/0x1f5
[<c04b5936>] ? sys_mknod+0x18/0x1a
[<c091fc1a>] mount_root+0x4c/0x54
[<c091fd67>] prepare_namespace+0x145/0x16c
[<c091f330>] kernel_init+0x143/0x152
[<c091f1ed>] ? kernel_init+0x0/0x152
[<c0408e43>] kernel_thread_helper+0x7/0x10
```



15. End

- Now, you have an environment to debug Linux Kernel source code.
- All the credits go to Takis Blog.
 - <http://issaris.blogspot.com/2007/12/download-linux-kernel-sourcecode-from.html>

Thank you.