

How to install Real Time Kernel patch on Linux(Ubuntu 16.04)

1. Install Ubuntu on Virtual box or Real hardware.
2. Download a kernel

```
https://www.kernel.org/
```

- What means Number/version 4.13.15
 - 4: Kernel version
 - 13: Major revision
 - 15: Minor revision
- ! Kernel version and major revision number must be the same.
- Create Kernel directory and download kernel source into the directory.

```
mkdir Kernel && cd Kernel  
wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.13.15.tar.xz
```

3. Download RT-Patch into the Kernel directory.

```
https://wiki.linuxfoundation.org/realtime/preempt_rt_versions  
https://cdn.kernel.org/pub/linux/kernel/projects/rt/4.13/
```

Download it

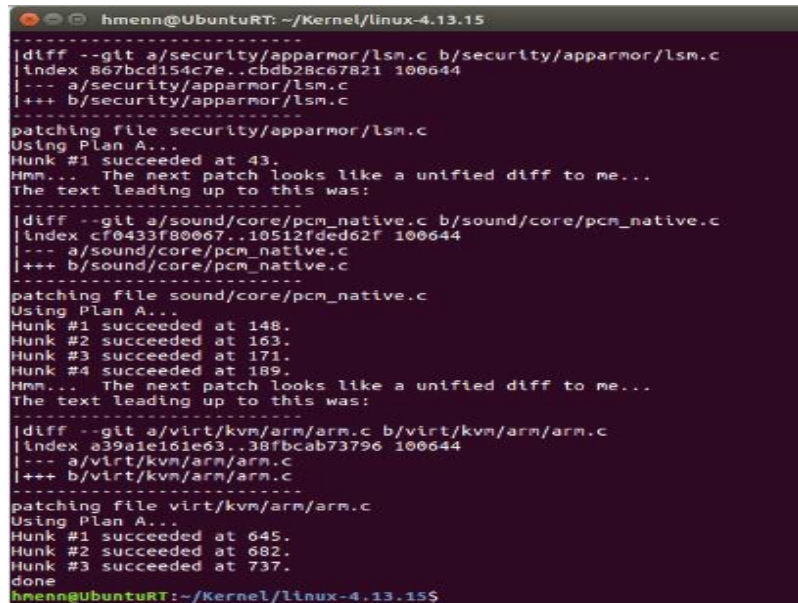
```
wget https://cdn.kernel.org/pub/linux/kernel/projects/rt/4.13/patch-  
4.13.13-rt5.patch.gz
```

4. Extract Kernel source and patch it.

```
tar xvf linux-4.13.15.tar.xz  
cd linux-4.13.15
```

Patch it

```
gzip -cd ../patch-4.13.13-rt5.patch.gz | patch -p1 --verbose
```



```
hmenng@UbuntuRT: ~/Kernel/linux-4.13.15
|diff --git a/security/apparmor/lsm.c b/security/apparmor/lsm.c
|index 867bcd154c7e..cbbdb28c67821 100644
|--- a/security/apparmor/lsm.c
|+++ b/security/apparmor/lsm.c
|-----
|patching file security/apparmor/lsm.c
|Using Plan A...
|Hunk #1 succeeded at 43.
|Hmm... The next patch looks like a unified diff to me...
|The text leading up to this was:
|-----
|diff --git a/sound/core/pcm_native.c b/sound/core/pcm_native.c
|index cf0433f80067..10512fded62f 100644
|--- a/sound/core/pcm_native.c
|+++ b/sound/core/pcm_native.c
|-----
|patching file sound/core/pcm_native.c
|Using Plan A...
|Hunk #1 succeeded at 148.
|Hunk #2 succeeded at 163.
|Hunk #3 succeeded at 171.
|Hunk #4 succeeded at 189.
|Hmm... The next patch looks like a unified diff to me...
|The text leading up to this was:
|-----
|diff --git a/virt/kvm/arm/arm.c b/virt/kvm/arm/arm.c
|index a39a1e161e63..38fbcab73796 100644
|--- a/virt/kvm/arm/arm.c
|+++ b/virt/kvm/arm/arm.c
|-----
|patching file virt/kvm/arm/arm.c
|Using Plan A...
|Hunk #1 succeeded at 645.
|Hunk #2 succeeded at 682.
|Hunk #3 succeeded at 737.
|done
hmenng@UbuntuRT:~/Kernel/linux-4.13.15$
```

5. Install libncurses5-dev and libssl-dev to enable real time processing and compiling packages.

```
sudo apt-get install libncurses5-dev
sudo apt-get install libssl-dev
```

6. Configure Makefile before compiling kernel. This configure will open Real time preempt options.

```
make menuconfig
```

Select Fully Preemptible Kernel option from

1. Processor type and features
2. Preemption Model (Voluntary Kernel Preemption (Desktop))
3. Fully Preemptible Kernel (RT)

```

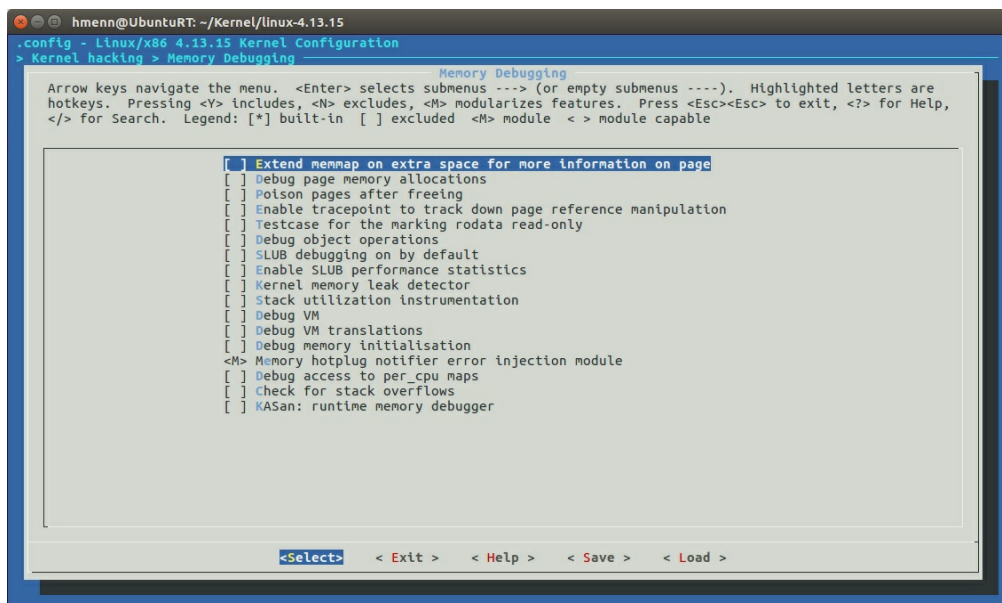
hmenn@UbuntuRT: ~/Kernel/linux-4.13.15
hmenn@UbuntuRT:~/Kernel/linux-4.13.15$ make menuconfig
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/mconf.o
SHIPPED scripts/kconfig/zconf.tab.c
SHIPPED scripts/kconfig/zconf.lex.c
SHIPPED scripts/kconfig/zconf.hash.c
HOSTCC scripts/kconfig/zconf.tab.o
HOSTCC scripts/kconfig/Lxdialog/checklist.o
HOSTCC scripts/kconfig/Lxdialog/util.o
HOSTCC scripts/kconfig/Lxdialog/inputbox.o
HOSTCC scripts/kconfig/Lxdialog/textbox.o
HOSTCC scripts/kconfig/Lxdialog/yesno.o
HOSTCC scripts/kconfig/Lxdialog/menubox.o
HOSTLD scripts/kconfig/mconf Kconfig
scripts/kconfig/mconf Kconfig
#
# using defaults found in /boot/config-4.10.0-28-generic
#
/boot/config-4.10.0-28-generic:5940:warning: symbol value 'm' invalid for SND_DESIGNWARE_PCM

*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.
hmenn@UbuntuRT:~/Kernel/linux-4.13.15$ make menuconfig

```

Go upper menu with ESC-ESC key. Deselect stack overflows if selected.

1. Kernel hacking --> [Enter]
2. Memory Debugging [Enter]
3. Check for stack overflows



Go upper

menu and save(RIGHT_ARROW) .config file.

7. Compile Kernel source file

```
make
```

NOTE: You can use "make -jX" option if you have thread support. X is number of core + 1. It will compile faster. For VirtualBox use this.

```
make -j2
```

Wait until compilation done. It can take 3-4 hour on VirtualBox.

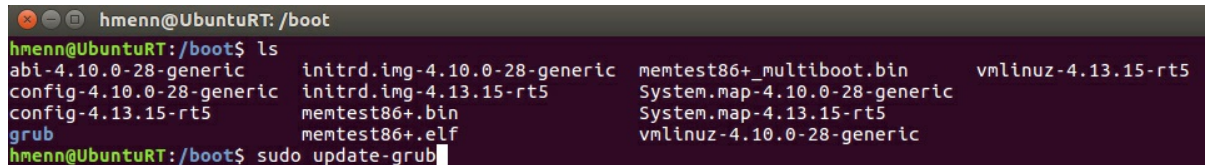
8. Make modules and install

```
sudo make modules_install -j2
sudo make install -j2
```

9. Check new kernel file and update grub boot loader to start Linux with new RT-Kernel.

```
cd /boot
ls
```

- You will see new kernel.



```
hmen@UbuntuRT: /boot
hmen@UbuntuRT:/boot$ ls
abi-4.10.0-28-generic      initrd.img-4.10.0-28-generic  memtest86+_multiboot.bin    vmlinuz-4.13.15-rt5
config-4.10.0-28-generic  initrd.img-4.13.15-rt5      System.map-4.10.0-28-generic
config-4.13.15-rt5       memtest86+.bin              System.map-4.13.15-rt5
grub                     memtest86+.elf              vmlinuz-4.10.0-28-generic
hmen@UbuntuRT:/boot$ sudo update-grub
```

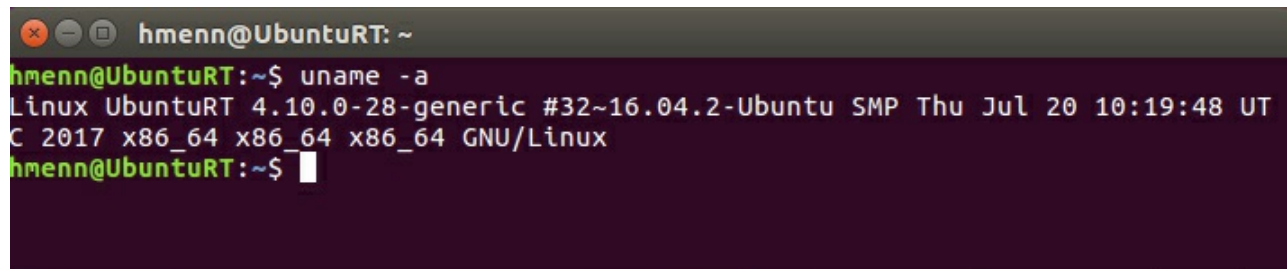
- Now update grub and reboot machine.

```
sudo update-grub
sudo reboot
```

10. Check kernel version to be sure.

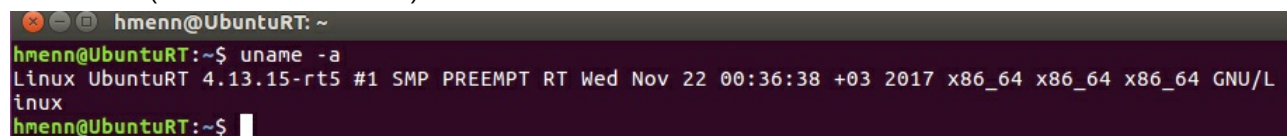
```
uname -a
```

- Old Kernel(Ubuntu SMP 4.10.0)



```
hmen@UbuntuRT: ~
hmen@UbuntuRT:~$ uname -a
Linux UbuntuRT 4.10.0-28-generic #32~16.04.2-Ubuntu SMP Thu Jul 20 10:19:48 UT
C 2017 x86_64 x86_64 x86_64 GNU/Linux
hmen@UbuntuRT:~$
```

- New Kernel(PREEMPT RT 4.13.15)



```
hmen@UbuntuRT: ~
hmen@UbuntuRT:~$ uname -a
Linux UbuntuRT 4.13.15-rt5 #1 SMP PREEMPT RT Wed Nov 22 00:36:38 +03 2017 x86_64 x86_64 x86_64 GNU/L
inux
hmen@UbuntuRT:~$
```