

1. Install –rt kernel

-rt kernel is still supported. You can find the list from the link below.
<https://mirrors.edge.kernel.org/pub/linux/kernel/projects/rt/>

However, since this page is maintained by kernel.org, operated by the Linux Kernel Organization, Inc., I believe this kernel is for vanilla linux kernel instead of Ubuntu linux kernel. In other words, we should patch it to vanilla linux kernel from
<https://mirrors.edge.kernel.org/pub/linux/kernel/>.

Before we get started, I would hope that your current kernel is –generic kernel. If not, please reboot and select –generic kernel.

Let's change to super user to avoid future permission issues.

```
sudo -s
```

1.1 Get the sources

First, we need to find our patch and the compatible Linux kernel edition.

<https://mirrors.edge.kernel.org/pub/linux/kernel/projects/rt/>



Index of /pub/linux/kernel/projects/rt/

../	08-Aug-2013 18:24	-
2.6.22/	08-Aug-2013 18:26	-
2.6.23/	08-Aug-2013 18:27	-
2.6.24/	08-Aug-2013 18:27	-
2.6.25/	08-Aug-2013 18:28	-
2.6.26/	08-Aug-2013 18:28	-
2.6.29/	04-Nov-2014 14:19	-
2.6.31/	08-Aug-2013 18:29	-
2.6.33/	19-Nov-2013 22:02	-
3.0/	23-Nov-2017 05:44	-
3.10/	08-Jun-2017 13:40	-
3.12/	13-Feb-2017 22:26	-
3.14/	17-Aug-2018 04:15	-
3.18/	23-Nov-2017 05:53	-
3.2/	16-Nov-2016 19:26	-
3.4/	19-Nov-2013 22:01	-
3.6/	04-Nov-2014 13:35	-
3.8/	13-Jul-2015 21:06	-
4.0/	29-Nov-2017 22:12	-
4.1/	17-Oct-2017 13:42	-
4.11/	17-Nov-2017 17:03	-
4.13/	31-Jul-2018 21:12	-
4.14/	03-Aug-2018 07:39	-
4.16/	07-Aug-2018 14:27	-
4.18/	16-Aug-2018 20:09	-
4.4/	30-Sep-2016 21:37	-
4.6/	23-Dec-2016 15:26	-
4.8/	06-Aug-2018 09:04	-
4.9/		



<https://mirrors.edge.kernel.org/pub/linux/kernel/projects/rt/4.16/>



Index of /pub/linux/kernel/projects/rt/4.16/

../		
incr/	03-Aug-2018 07:39	-
older/	03-Aug-2018 07:39	-
patch-4.16.18-rt12.patch.gz	03-Aug-2018 07:39	292K
patch-4.16.18-rt12.patch.sign	03-Aug-2018 07:39	566
patch-4.16.18-rt12.patch.xz	03-Aug-2018 07:39	225K
patches-4.16.18-rt12.tar.gz		455K
patches-4.16.18-rt12.tar.sign		566
patches-4.16.18-rt12.tar.xz		329K
sha256sums.asc		1261

```
wget https://mirrors.edge.kernel.org/pub/linux/kernel/projects/rt/4.16/patch-4.16.18-rt12.patch.xz
```

Next, we need to download the Linux kernel which goes with our patch – **linux-4.16.18**

linux-4.16.16.tar.gz	16-Jun-2018 07:46	152M
linux-4.16.16.tar.sign	16-Jun-2018 07:46	833
linux-4.16.16.tar.xz	16-Jun-2018 07:46	98M
linux-4.16.17.tar.gz	20-Jun-2018 19:04	152M
linux-4.16.17.tar.sign	20-Jun-2018 19:04	833
linux-4.16.17.tar.xz	20-Jun-2018 19:04	98M
linux-4.16.18.tar.gz	25-Jun-2018 23:59	152M
linux-4.16.18.tar.sign	25-Jun-2018 23:59	833
linux-4.16.18.tar.xz	25-Jun-2018 23:59	98M
linux-4.16.2.tar	2018 10:33	152M
linux-4.16.2.tar	2018 10:33	833
linux-4.16.2.tar	2018 10:33	98M
linux-4.16.3.tar	2018 06:57	152M
linux-4.16.3.tar	2018 06:57	833
linux-4.16.3.tar	2018 06:57	98M
linux-4.16.4.tar	2018 07:46	152M
linux-4.16.4.tar	2018 07:46	833
linux-4.16.4.tar	2018 07:46	98M
linux-4.16.5.tar	2018 09:03	152M
linux-4.16.5.tar	2018 09:03	833
linux-4.16.5.tar	2018 09:03	98M
linux-4.16.6.tar.gz	26-Apr-2018 09:03	152M
linux-4.16.6.tar.sign	29-Apr-2018 19:47	833
linux-4.16.6.tar.xz	29-Apr-2018 19:47	98M
linux-4.16.7.tar.gz	02-May-2018 15:04	152M

```
wget https://mirrors.edge.kernel.org/pub/linux/kernel/v4.x/linux-4.16.18.tar.xz
```

After downloading, we need to unpack the archives and patch the Linux kernel.

```
xz -cd linux-4.16.18.tar.xz | tar xvf -
cd linux-4.16.18
xzcat ../patch-4.16.18-rt12.patch.xz | patch -p1
```

1.2 Configure the kernel

There are some packages required to help us configure the kernel.

```
sudo apt-get install qt5-default libssl-dev libelf-dev
sudo apt-get install bison flex
```

Take the actual working config, which is the configuration for generic kernel:

```
yes "" | make oldconfig
```

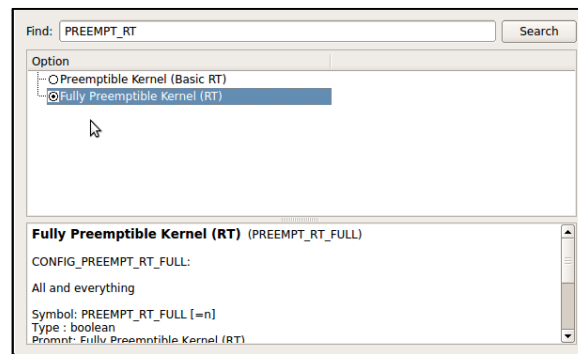
Change your configuration.

```
make xconfig
```

*The only necessary configuration for real-time Linux kernel is the choice of the “**Fully Preemptible Kernel**” preemption model (`CONFIG_PREEMPT_RT_FULL`). All other kernel configuration parameters depend on system requirements. For detailed information about how to configure a kernel have a look at [Linux kernel documentation](#).*

[must-do]

* Enable `CONFIG_PREEMPT_RT_FULL` option !



Recommendation Configuration (if you cannot change some configuration, just forget about it)

- * General setup
 - > Timers subsystem
 - > High Resolution Timer Support (Enable)
- * Power management and ACPI options
 - > ACPI (Advanced Configuration and Power Interface) Support
 - > Processor (Disable)
 - > CPU Frequency scaling
 - > CPU Frequency scaling (Disable)
 - > CPU Idle
 - > CPU idle PM support (Disable)
- * Processor type and features
 - > Enable maximum number of SMP processors and NUMA nodes (Disable)
 - > Processor family
 - > Core 2/newer Xeon if "cat /proc/cpuinfo | grep family" returns 6,
 - > set as Generic otherwise
 - > Transparent Hugepage Support (Disable)
 - > Allow for memory compaction (Disable)
 - > Contiguous Memory Allocation (Disable)
 - > Allow for memory compaction
 - > Page Migration (Disable)

Save configuration **CTRL+S**.

1.3 Build the kernel

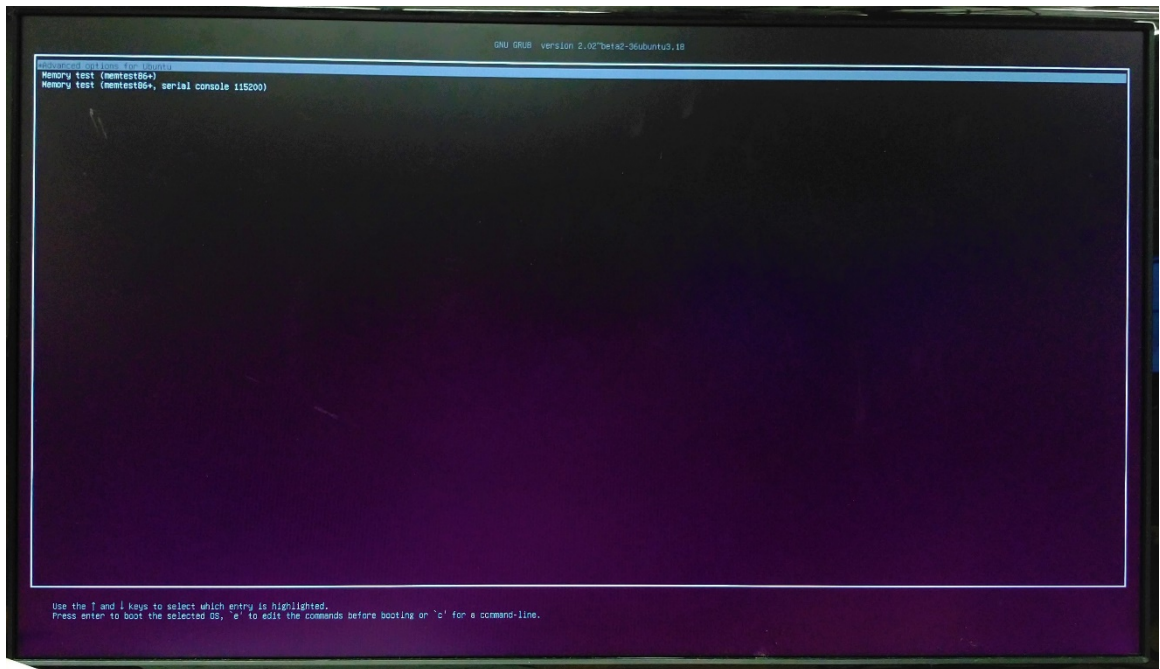
“-j4” is for my quad-core CPU. This can take a long time.

```
make -j4  
make -j4 modules  
make -j4 modules_install  
make -j4 install
```

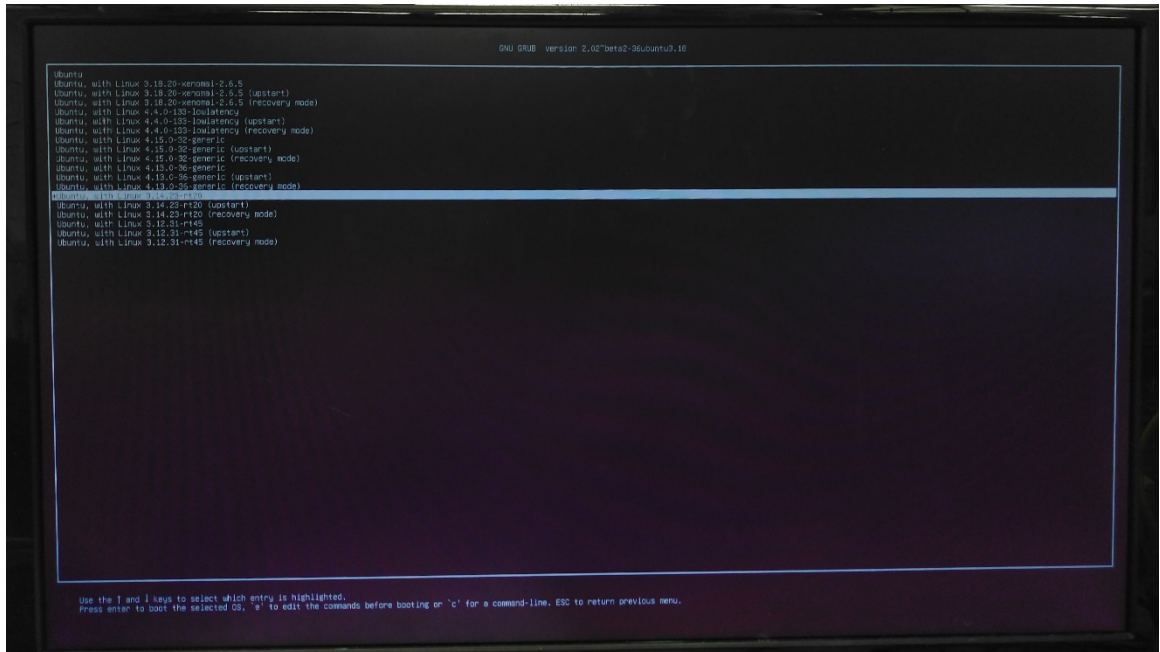
1.4 Reboot Ubuntu system

```
sudo reboot
```

After reboot, select “Advanced options for Ubuntu”. See the picture below.

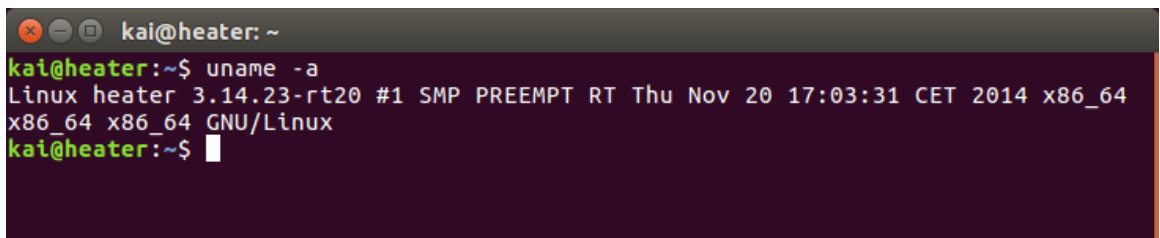


Under this directory, you should be able to see one option end with “-rtXX”. Use to select it and press “ENTER” key. Now, wait for it to finish the rebooting.



After the reboot, let's see your current kernel with the following command.

```
uname -a
```



If it matches the one you chose, then your installation is succeed! Congratulations !

2. Test your latency

To test your latency, first make sure that you have `cyclicttest_run.sh` and `cyclicttest_plot.sh` and made them executable using `chmod +x` command.

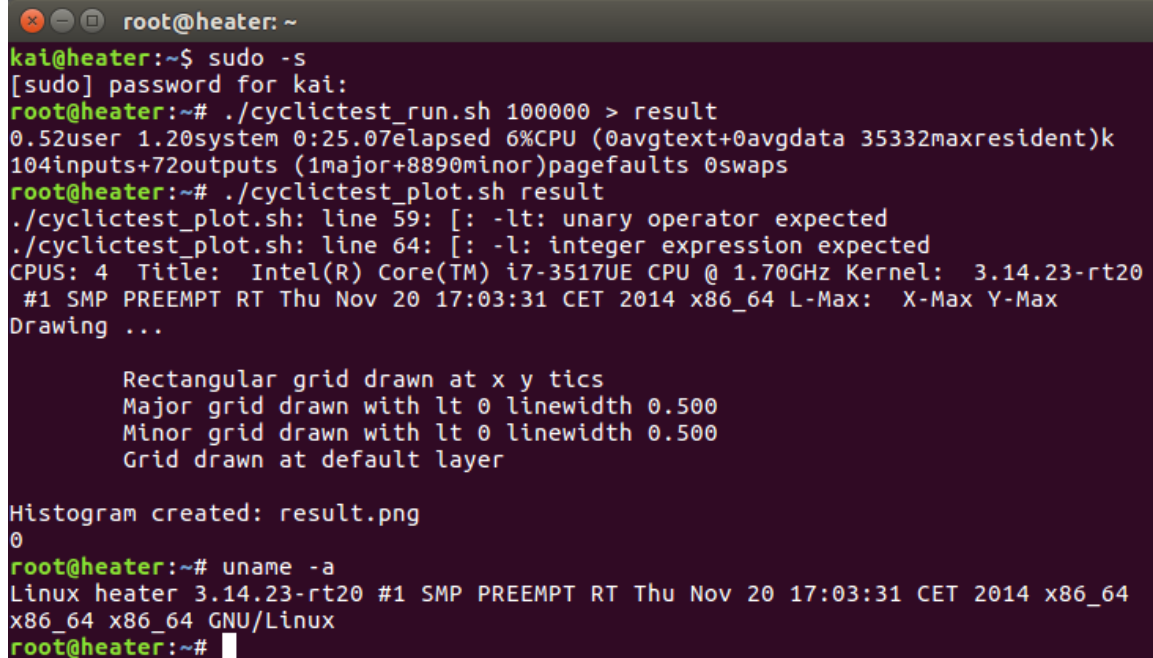
Now let's run it.

```
sudo -s #we need root permission
./cyclicttest_run.sh 100000 > result
```

Wait for a few minutes. This command creates a file named "result" containing the testing result. Then,

```
./cyclicttest_plot.sh result
```

This command visualizes your data and put it in `result.png` in your current directory.

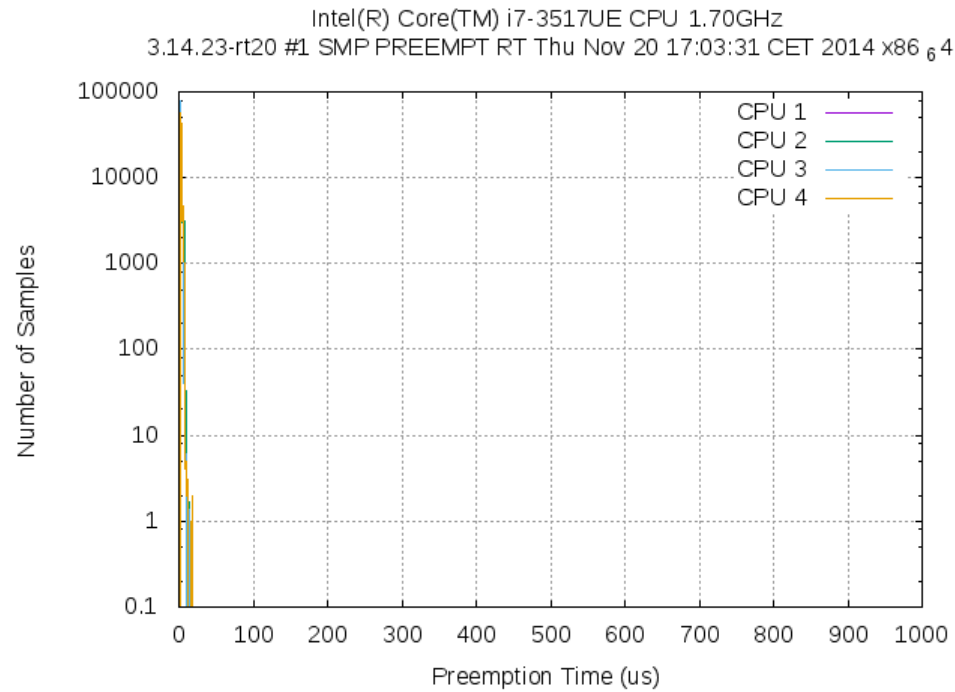


```
root@heater: ~
kai@heater:~$ sudo -s
[sudo] password for kai:
root@heater:~# ./cyclicttest_run.sh 100000 > result
0.52user 1.20system 0:25.07elapsed 6%CPU (0avgtext+0avgdata 35332maxresident)k
104inputs+72outputs (1major+8890minor)pagefaults 0swaps
root@heater:~# ./cyclicttest_plot.sh result
./cyclicttest_plot.sh: line 59: [: -lt: unary operator expected
./cyclicttest_plot.sh: line 64: [: -l: integer expression expected
CPUS: 4 Title: Intel(R) Core(TM) i7-3517UE CPU @ 1.70GHz Kernel: 3.14.23-rt20
#1 SMP PREEMPT RT Thu Nov 20 17:03:31 CET 2014 x86_64 L-Max: X-Max Y-Max
Drawing ...

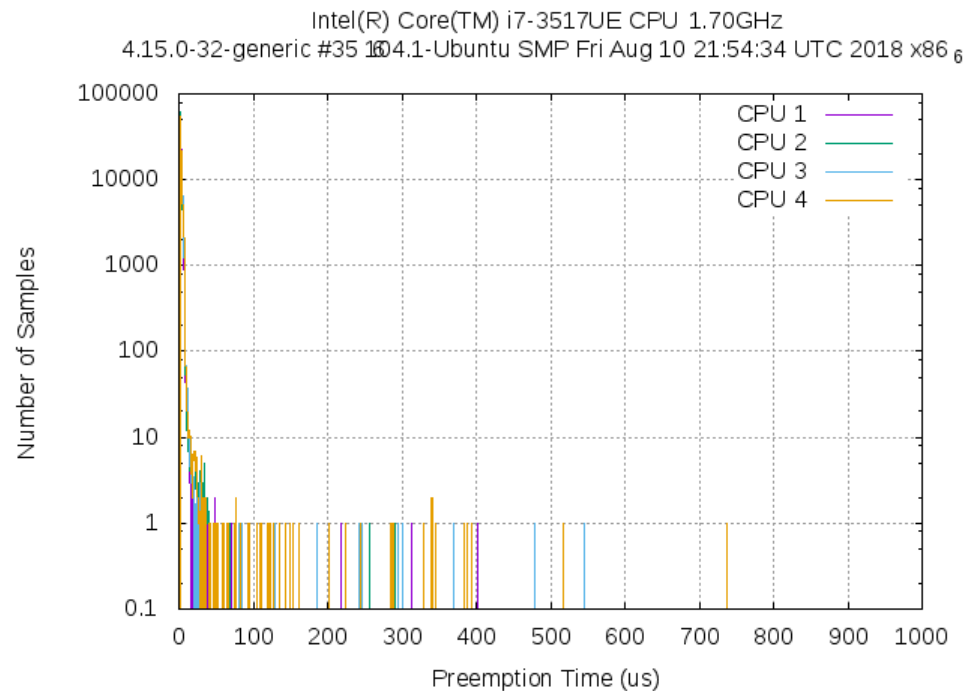
Rectangular grid drawn at x y tics
Major grid drawn with lt 0 linewidth 0.500
Minor grid drawn with lt 0 linewidth 0.500
Grid drawn at default layer

Histogram created: result.png
0
root@heater:~# uname -a
Linux heater 3.14.23-rt20 #1 SMP PREEMPT RT Thu Nov 20 17:03:31 CET 2014 x86_64
x86_64 x86_64 GNU/Linux
root@heater:~#
```

The following picture shows the latency test result for `-rt` kernel.



And you can compare it to latency result for `-generic` kernel



The preemption time here actually stands for latency. As you can see, the latency dropped dramatically!