# OAI Build platform guide ( for EPC )

**tags:** `OAI`

## Hardware environment

| **Hardware** | |
|---|---|
| CPU | Intel i7 8700 |
| RAM | 16GB |
| Disk | SSD 256GB |
| Network interface | ***2*** |

## Software environment

| **Software** | |
|---|---|
| OS | Ubuntu 16.04.5 |
| Kernel | 4.7.7 oaiepc |

---

## Installation Flow

1. Install linux kernel
2. Environment Setting
3. Build EPC

---

### 1. Install Linux Kernel

- Install linux kernel version 4.7.7-oaiepc for CN

```
$ git clone https://gitlab.eurecom.fr/oai/linux-4.7.x.git
$ cd linux-4.7.x
$ sudo dpkg -i *.deb
```

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

- During Reboot, press ***ESC*** and choose advance option to choose the kernel version

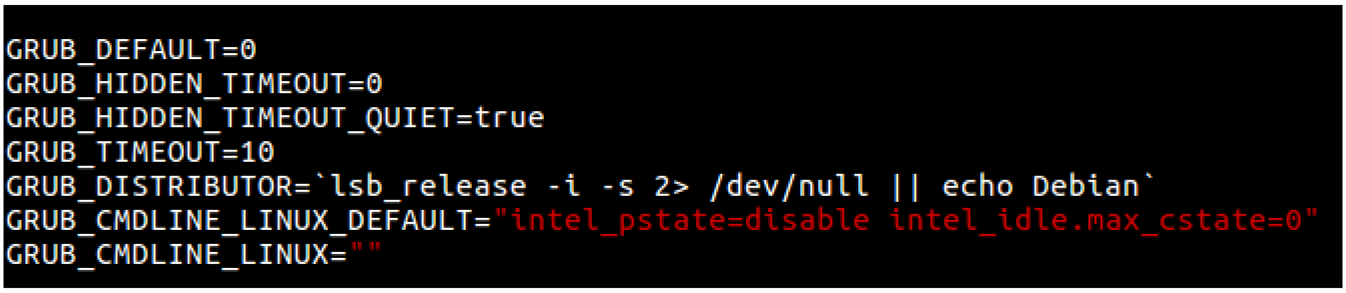
- Choose ***4.7.7-oaiepc kernel version***

## 2. Environment Setting

### ( a ) Disable C-states and P-states in Linux

```
$ sudo vim /etc/default/grub
```

- Add “intel_pstate=disable intel_idle.max_cstate=0 processor.max_cstate=0 idle=poll” in line GRUB_CMDLINE_LINUX_DEFAULT



```
GRUB_DEFAULT=0
GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="intel_pstate=disable intel_idle.max_cstate=0"
GRUB_CMDLINE_LINUX=""
```

- Second , perform update-grub

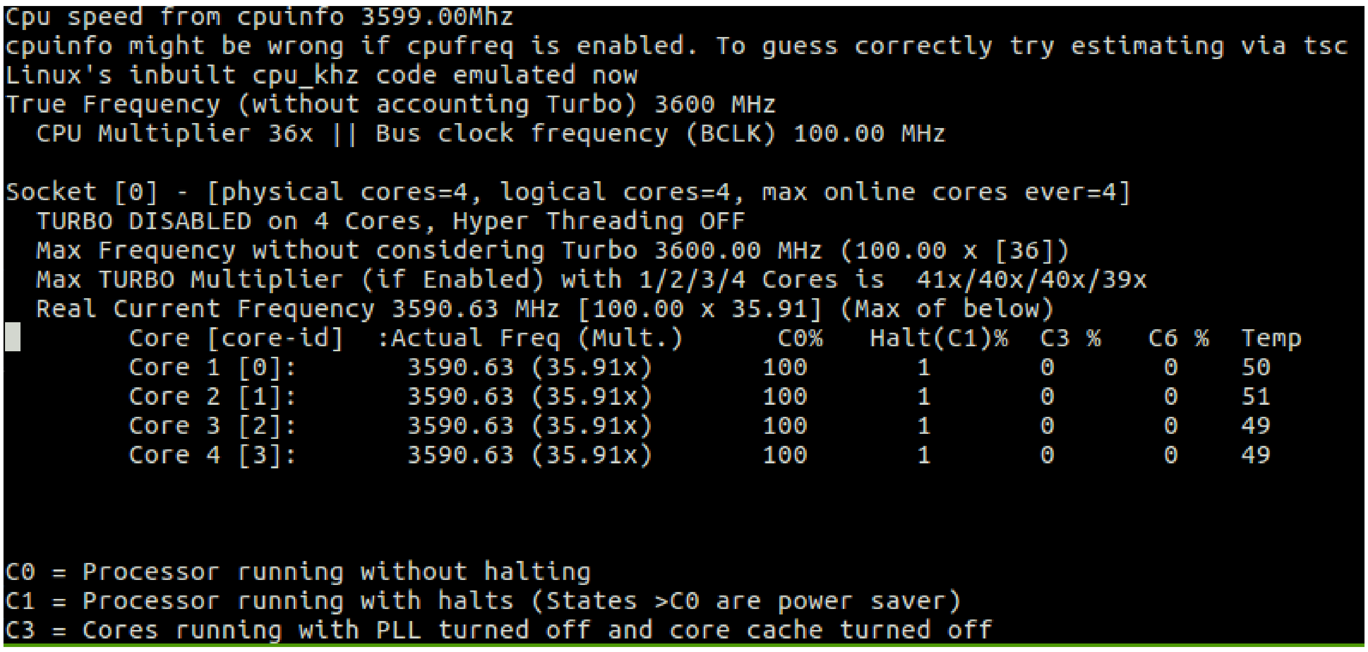
```
$ sudo update-grub
$ reboot
```

### ( b ) Remove all power management and CPU frequency scaling

1. Append “blacklist intel_powerclamp” to the end of /etc/modprobe.d/blacklist.conf
2. Disable “hyperthreading”, “CPU frequency control”, “C-states”, “P-states” features in BIOS.
3. Install i7z to check the cpu

```
$ sudo apt-get isntall i7z
$ sudo i7z
```

4. The CPU should not change its frequency by more than 1-2 hertz and should not be any C-state other than C0



```
Cpu speed from cpuinfo 3599.00Mhz
cpuinfo might be wrong if cpufreq is enabled. To guess correctly try estimating via tsc
Linux's inbuilt cpu_khz code emulated now
True Frequency (without accounting Turbo) 3600 MHz
  CPU Multiplier 36x || Bus clock frequency (BCLK) 100.00 MHz

Socket [0] - [physical cores=4, logical cores=4, max online cores ever=4]
  TURBO DISABLED on 4 Cores, Hyper Threading OFF
  Max Frequency without considering Turbo 3600.00 MHz (100.00 x [36])
  Max TURBO Multiplier (if Enabled) with 1/2/3/4 Cores is  41x/40x/40x/39x
  Real Current Frequency 3590.63 MHz [100.00 x 35.91] (Max of below)
        Core [core-id]  :Actual Freq (Mult.)      C0%   Halt(C1)%  C3 %   C6 %  Temp
        Core 1 [0]:       3590.63 (35.91x)       100       1        0       0    50
        Core 2 [1]:       3590.63 (35.91x)       100       1        0       0    51
        Core 3 [2]:       3590.63 (35.91x)       100       1        0       0    49
        Core 4 [3]:       3590.63 (35.91x)       100       1        0       0    49


C0 = Processor running without halting
C1 = Processor running with halts (States >C0 are power saver)
C3 = Cores running with PLL turned off and core cache turned off
```

5. Disable CPU frequency scaling

   (1) Get cpufrequtils

   ```
   $ sudo apt-get install cpufrequtils
   ```

   (2) Edit the following file ( If it doesn't exist, create it )

   ```
   $ sudo vim /etc/default/cpufrequtils
   ```

   (3) Then, add the following line into it and save it.

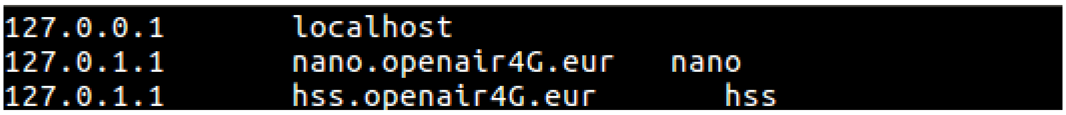
   ```
   GOVERNOR="performance"
   ```

   (4) Disable ondemand daemon

   ```
   $ sudo update-rc.d ondemand disable
   ```

**( c ) Change your hostname**

```
$ sudo vim /etc/hosts
```



```
127.0.0.1        localhost
127.0.1.1        nano.openair4G.eur    nano
127.0.1.1        hss.openair4G.eur        hss
```

## 3. Build EPC

**Overview of Installation**

- Prerequsite
- Build EPC
  - Build hss
  - Build mme
  - Build spgw

---

**Prequisite**

- Install git

```
$ sudo apt-get install git
```

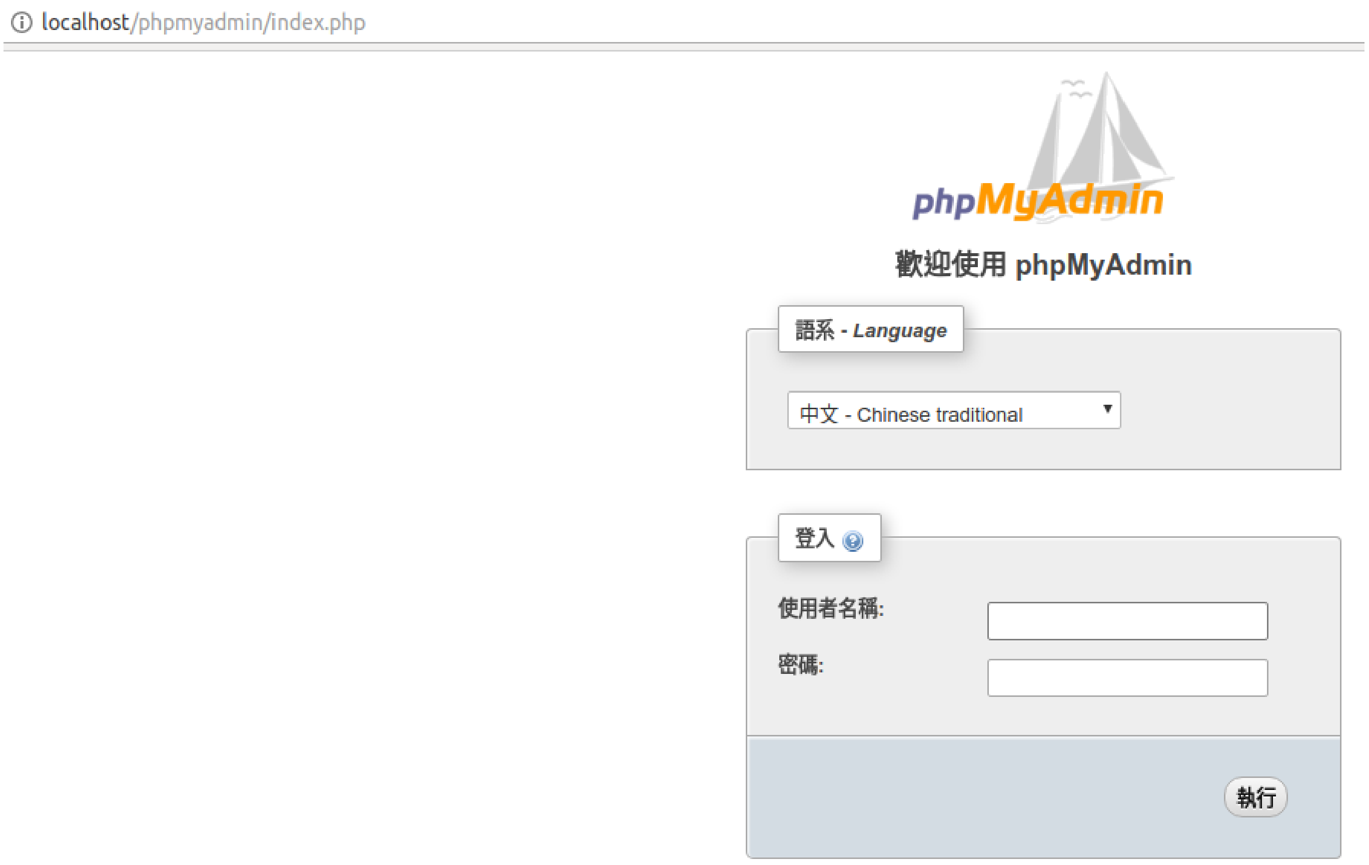
- Install mysql-server

```
$ sudo apt-get install mysql-server
```

- Install phpmyadmin
  - Username : root
  - Password : admin
  - Server : Apache2

```
$ sudo apt-get install phpmyadmin
```

After Installation , go to http://localhost/phpmyadmin to check if it is installed correctly.



If it failed , try the following method

( Method 1 )

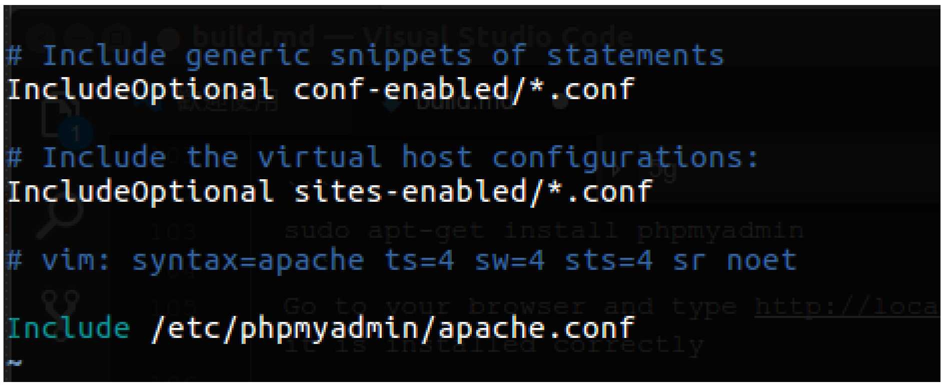
```
$ sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf-
available/phpmyadmin.conf
$ sudo a2enconf phpmyadmin
$ sudo service apache2 reload
```

( Method 2 )

```
$ sudo vim /etc/apache2/apache2.conf
```

and add the following line to the end of the file, then restart apache2

```
Include /etc/phpmyadmin/apache.conf
```



```
# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

Include /etc/phpmyadmin/apache.conf
~
```

```
$ /etc/init.d/apache2 restart
```

**Build EPC**

**( a ) Modify conf files**

```
git clone http://gitlab.nems.cs.nctu.edu.tw/huangpoh1/conf.git
```

**Step 1. Copy and reconfigure *.conf files**

```
$ sudo mkdir -p /usr/local/etc/oai
$ sudo mkdir -p /usr/local/etc/oai/freeDiameter
$ cd cond
$ sudo cp hss.conf mme.conf spgw.conf /usr/local/etc/oai
$ cd freeDiameter
$ sudo cp acl.conf hss_fd.conf mme_fd.conf /usr/local/etc/oai/freeDiameter
```

# [ You have to modify the following conf. files based on your PC ]

1. hss.conf
2. mme.conf
3. spgw.conf
4. hss_fd.conf
5. mme_fd.conf

**( b ) Build**

```
$ git clone https://gitlab.eurecom.fr/oai/openair-cn.git
$ cd openair-cn
$ git checkout 724542d0b59797b010af8c5df15af7f669c1e838
$ cd script
```

**( 1 ) Build hss**

```
$ sudo ./build_hss -i
$ sudo ./build_hss
$ sudo ./hss_db_create 127.0.0.1 root admin root admin oai_db
$ ./check_hss_s6a_certificate /usr/local/etc/oai/freeDiameter
hss.openair4G.eur
```

**( 2 ) Build mme**

```
$ sudo ./build_mme -i -f
$ sudo ./build_mme
$ ./check_mme_s6a_certificate /usr/local/etc/oai/freeDiameter
nano.openair4G.eur
```

**( 3 ) Build spgw**

```
$ sudo ./build_spgw -i -f
$ sudo ./build_spgw
```

You have to see "hss/mme/spgw has been compiled" after every build.

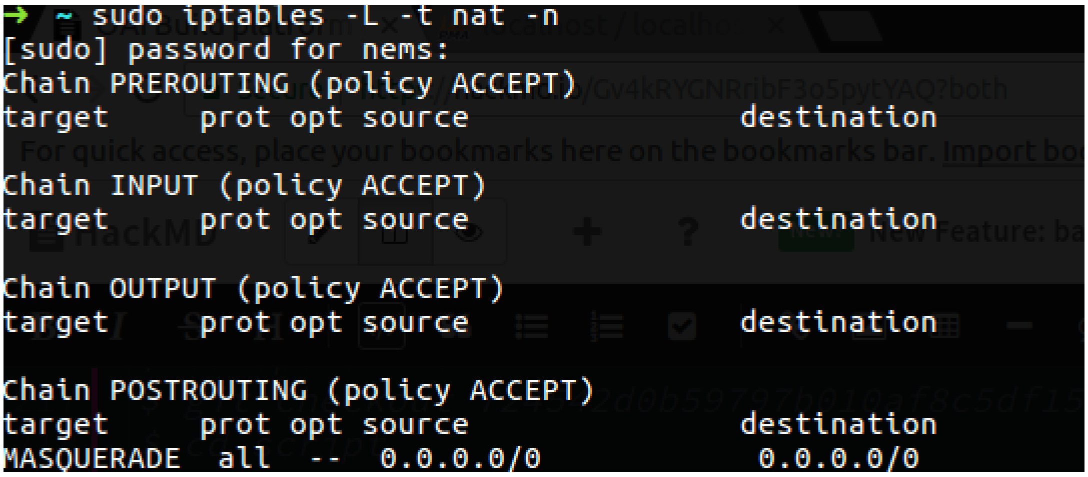
---

## OTEHRS

- check out your iptables

```
$ sudo iptables -t nat -L -n
```

it should be ...



```
➜  ~ sudo iptables -L -t nat -n
[sudo] password for nems:
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  0.0.0.0/0            0.0.0.0/0
```

Be aware that MASQUERADE should be existing. If there isn't any iptables policy list

```
$ sudo iptables -t nat -A POSTROUTING -o <spgw interface> -j MASQUERADE
```

So that your inner packet will be forwarded outside.

---

## Before you get start to run hss/mme/spgw, you have to get all interfaces prepared, e.g. interface for pgw, interface for mme.