Kernel Compile

Practical Class 4

Systems and Storage Laboratory
Department of Computer Science and Engineering
Chung-Ang University

Index

1. **Download Linux Kernel Source Code**
2. **Install Prerequisites**
3. **Kernel Configuration**
4. **Compile the Kernel**
5. **Kernel Installation**
6. **Grub Configuration**
7. **Boot into the Kernel**
8. **Appendix**
   - **Troubleshooting 1**: out of memory
   - **Troubleshooting 2**: `TSC_DEADLINE`
   - **Troubleshooting 3**: Initramfs unpacking failed

Related Resources

- There are some resources that you might find useful:
  - Ubuntu wiki – Build Kernel
  - Linux Kernel documentation
  - GNU GRUB Manual

- You can learn more about Building Linux from the above links

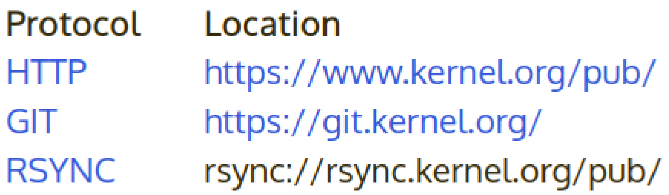
[1] https://git-scm.com/about/staging-area

1. Download Linux Kernel Source Code

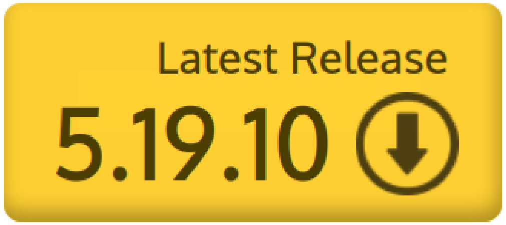
- **There is three ways to get the Linux kernel source code:**
  - From **website** https://www.kernel.org/ to download the source code archive file
  - Use **Git** to clone the kernel code repository
  - Download the source package from **package manager**
    - In Ubuntu, run `sudo apt-cache search linux-source` to check the available versions
    - These kernel sources are distro patched

1. Download Linux Kernel Source Code

- Here we try to download source code from website:
  - https://www.kernel.org/

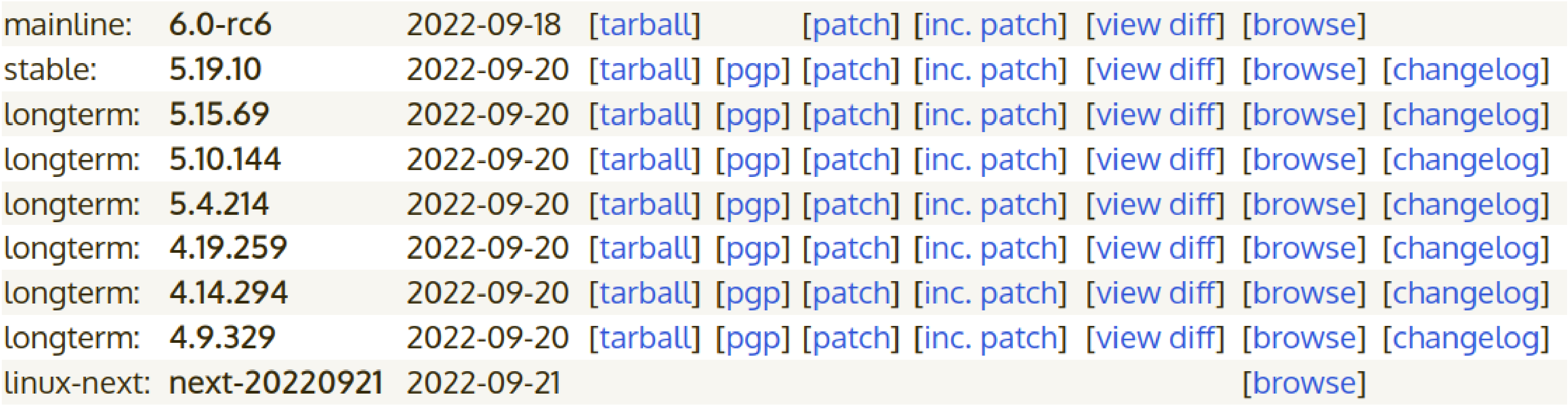


Protocol	Location
HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/



Latest Release

5.19.10

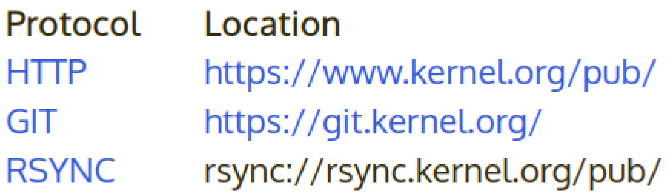


mainline:	6.0-rc6	2022-09-18	[tarball]		[patch]	[inc. patch]	[view diff]	[browse]	
stable:	5.19.10	2022-09-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.15.69	2022-09-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.10.144	2022-09-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.4.214	2022-09-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.19.259	2022-09-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.14.294	2022-09-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.9.329	2022-09-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
linux-next:	next-20220921	2022-09-21						[browse]	

1. Download Linux Kernel Source Code

Download Linux Kernel Source Codes

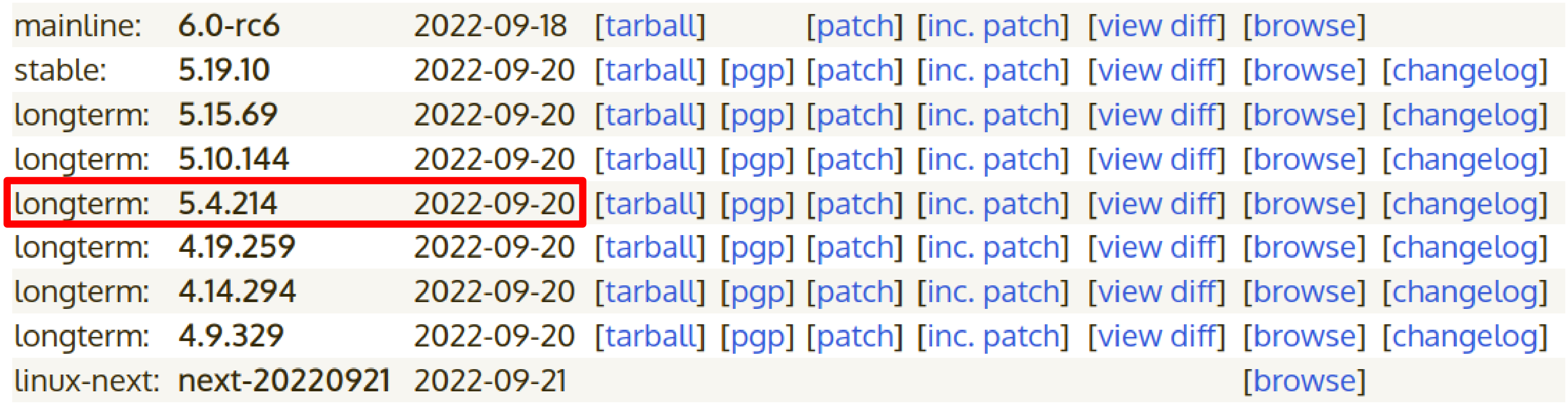
- You can find all the version of kernel from the website
- Here we will go with the latest 5.4 LTS version - **5.4.214**



Protocol	Location
HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

Latest Release

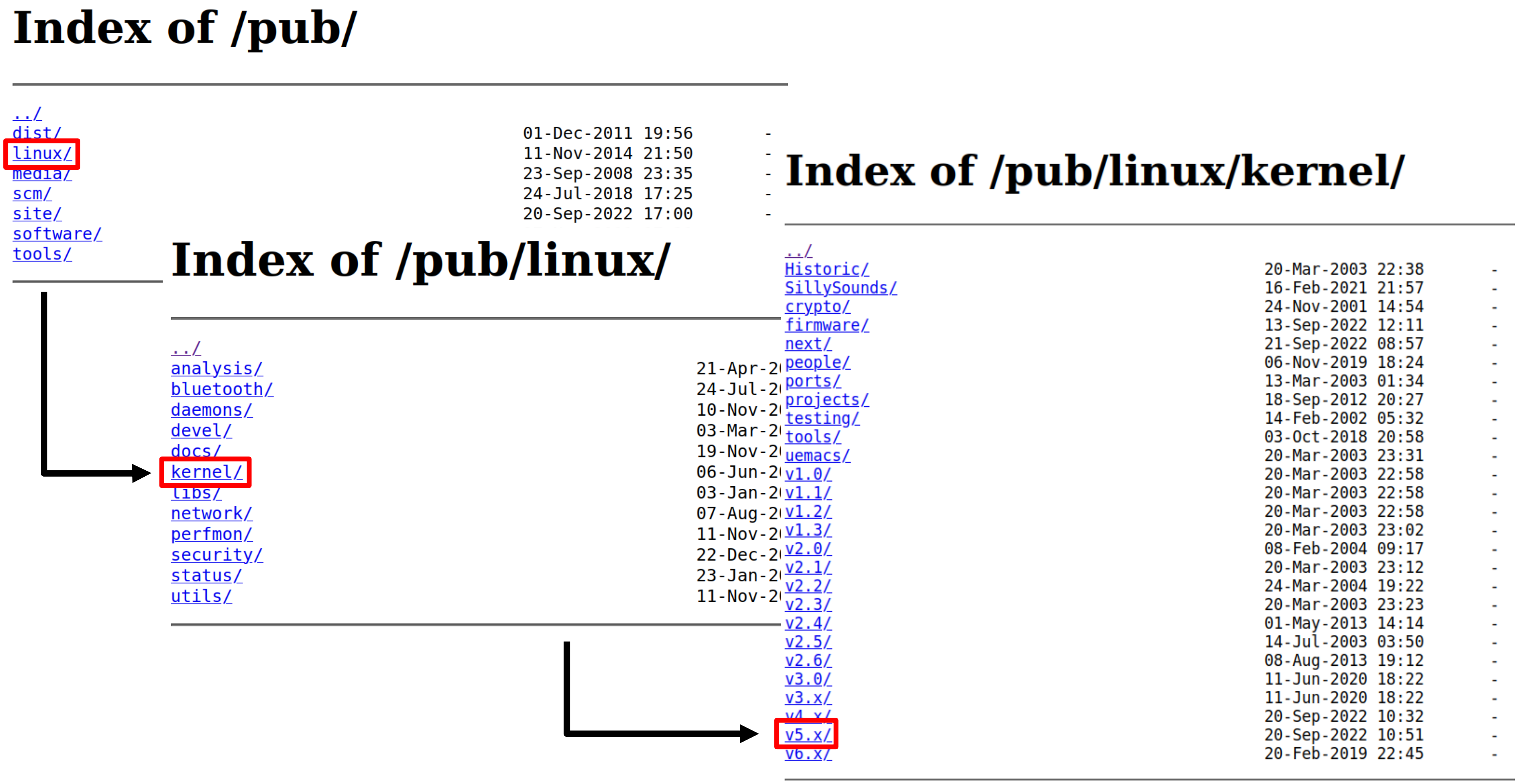
5.19.10



mainline:	6.0-rc6	2022-09-18	[tarball]		[patch]	[inc. patch]	[view diff]	[browse]	
stable:	5.19.10	2022-09-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.15.69	2022-09-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.10.144	2022-09-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.4.214	2022-09-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.19.259	2022-09-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.14.294	2022-09-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.9.329	2022-09-20	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
linux-next:	next-20220921	2022-09-21						[browse]	

1. Download Linux Kernel Source Code

- You can find other versions through:
  - https://mirrors.edge.kernel.org/pub/



Index of /pub/

```
../
dist/                                              01-Dec-2011 19:56       -
linux/                                             11-Nov-2014 21:50       -
media/                                             23-Sep-2008 23:35       -
scm/                                               24-Jul-2018 17:25       -
site/                                              20-Sep-2022 17:00       -
software/
tools/
```

Index of /pub/linux/

```
../
analysis/                                          21-Apr-2
bluetooth/                                         24-Jul-2
daemons/                                           10-Nov-2
devel/                                             03-Mar-2
docs/                                              19-Nov-2
kernel/                                            06-Jun-2
libs/                                              03-Jan-2
network/                                           07-Aug-2
perfmon/                                           11-Nov-2
security/                                          22-Dec-2
status/                                            23-Jan-2
utils/                                             11-Nov-2
```

Index of /pub/linux/kernel/

```
../
Historic/                                          20-Mar-2003 22:38       -
SillySounds/                                       16-Feb-2021 21:57       -
crypto/                                            24-Nov-2001 14:54       -
firmware/                                          13-Sep-2022 12:11       -
next/                                              21-Sep-2022 08:57       -
people/                                            06-Nov-2019 18:24       -
ports/                                             13-Mar-2003 01:34       -
projects/                                          18-Sep-2012 20:27       -
testing/                                           14-Feb-2002 05:32       -
tools/                                             03-Oct-2018 20:58       -
uemacs/                                            20-Mar-2003 23:31       -
v1.0/                                              20-Mar-2003 22:58       -
v1.1/                                              20-Mar-2003 22:58       -
v1.2/                                              20-Mar-2003 22:58       -
v1.3/                                              20-Mar-2003 23:02       -
v2.0/                                              08-Feb-2004 09:17       -
v2.1/                                              20-Mar-2003 23:12       -
v2.2/                                              24-Mar-2004 19:22       -
v2.3/                                              20-Mar-2003 23:23       -
v2.4/                                              01-May-2013 14:14       -
v2.5/                                              14-Jul-2003 03:50       -
v2.6/                                              08-Aug-2013 19:12       -
v3.0/                                              11-Jun-2020 18:22       -
v3.x/                                              11-Jun-2020 18:22       -
v4.x/                                              20-Sep-2022 10:32       -
v5.x/                                              20-Sep-2022 10:51       -
v6.x/                                              20-Feb-2019 22:45       -
```

2. Install Prerequisites

- **Before compiling our kernel, we should install some tools first.**
  - You can find the list of minimal requirements here.
- **Use the following commands to install these tools:**
  - These commands might be enough to build the Linux kernel

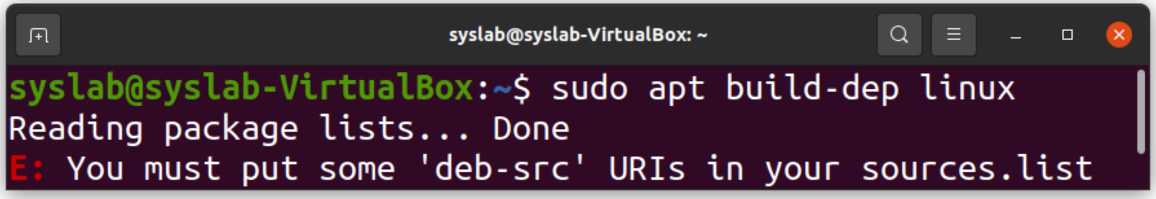
```
$ sudo apt build-dep linux
$ sudo apt install libncurses-dev
```

  - However if you have any issue with missing packages, try installing some more prerequisites

```
$ sudo apt build-dep linux linux-image-$(uname -r)
$ sudo apt install libncurses-dev gawk flex bison
openssl libssl-dev dkms libelf-dev libudev-dev
libpci-dev libiberty-dev autoconf llvm
```

2. Install Prerequisites

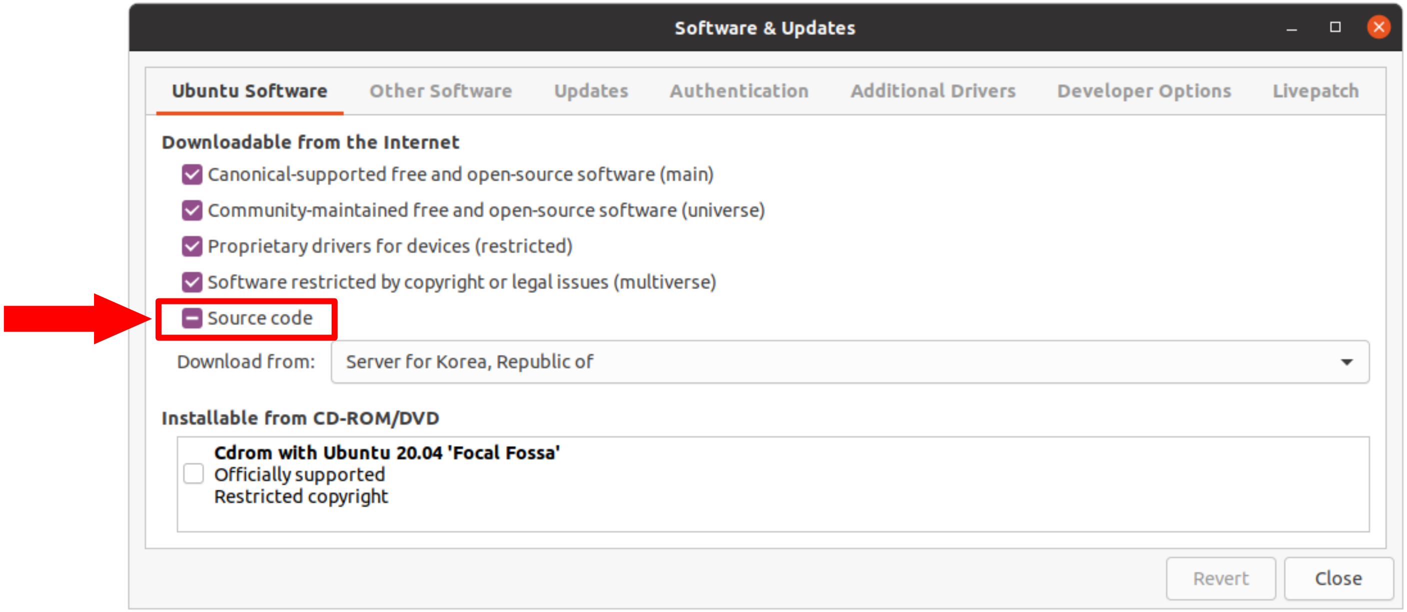
- **There may be an error like this when you use the first command:**



```
syslab@syslab-VirtualBox:~$ sudo apt build-dep linux
Reading package lists... Done
E: You must put some 'deb-src' URIs in your sources.list
```

- **To solve this problem:**

1. Go to system's **Software & Updates** setting and check the "**Source code**" option. It will generate the source URIs.
2. Then run previous commands again.



Downloadable from the Internet

- ☑ Canonical-supported free and open-source software (main)
- ☑ Community-maintained free and open-source software (universe)
- ☑ Proprietary drivers for devices (restricted)
- ☑ Software restricted by copyright or legal issues (multiverse)
- ⊟ Source code

Download from: Server for Korea, Republic of

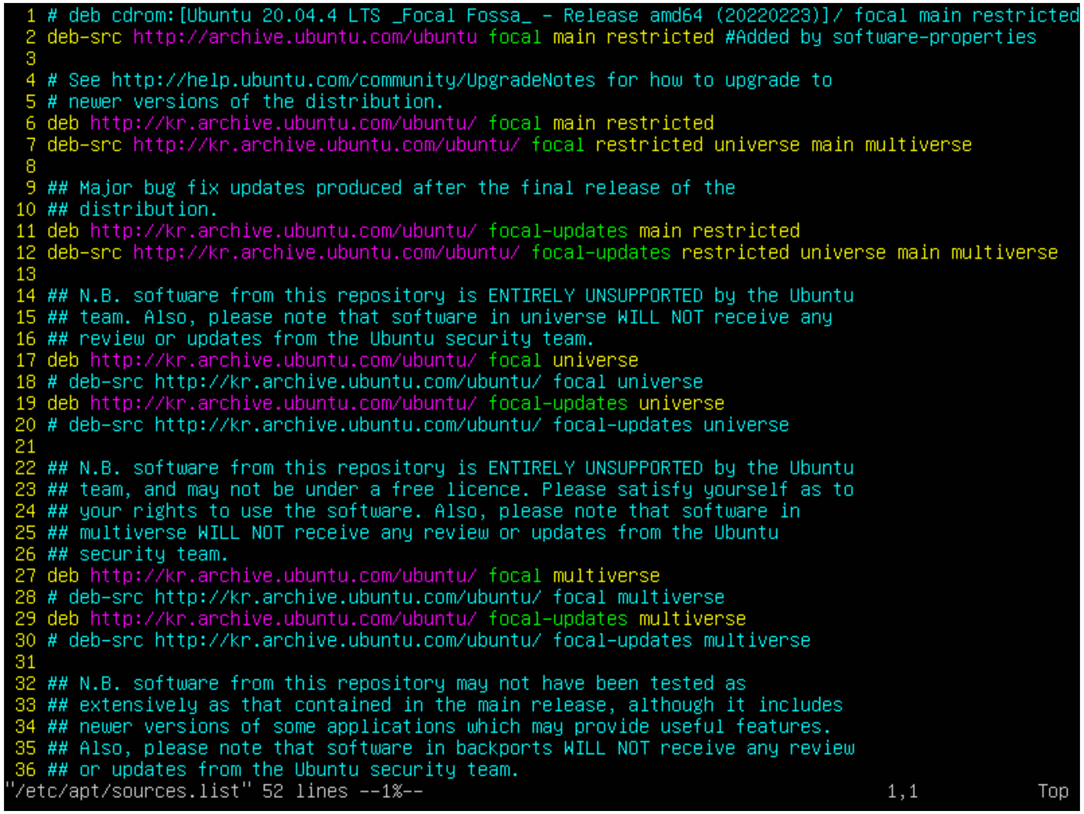
Installable from CD-ROM/DVD

☐ **Cdrom with Ubuntu 20.04 'Focal Fossa'**
Officially supported
Restricted copyright

2. Install Prerequisites

cf) For Ubuntu Server

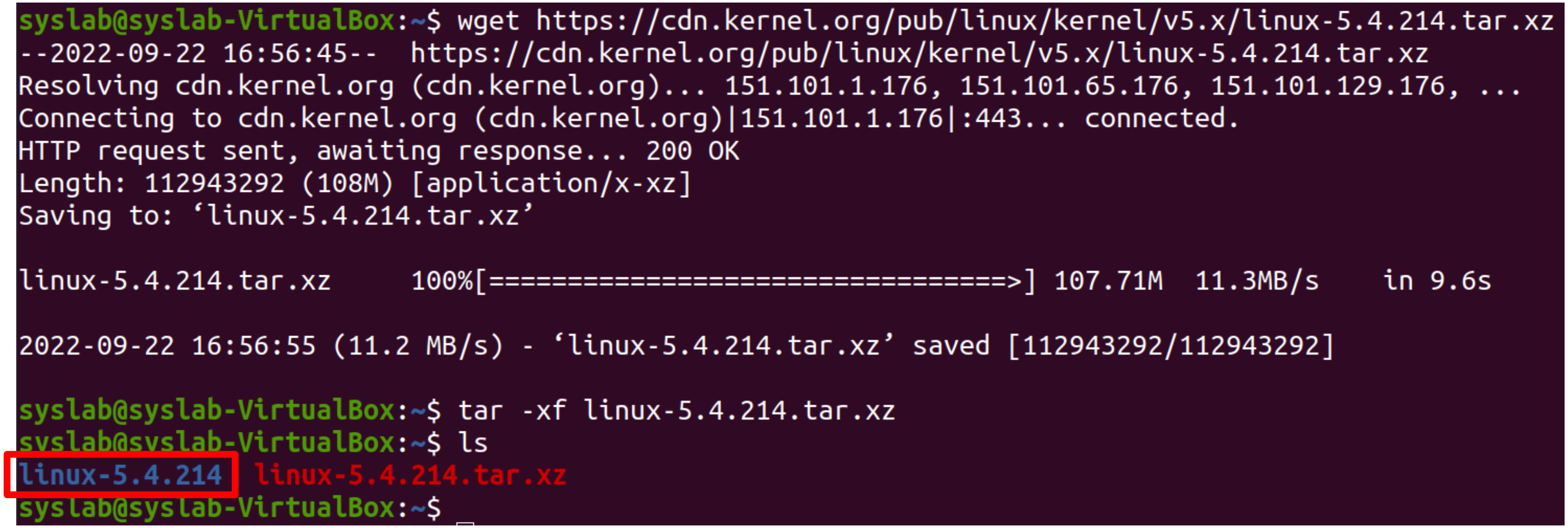
- You can also edit **`/etc/apt/sources.list`** by yourself
- Simply uncomment "**`deb-src`**" lines
- Don't forget to run **`apt-get update`** after adding sources



```
# deb cdrom:[Ubuntu 20.04.4 LTS _Focal Fossa_ - Release amd64 (20220223)]/ focal main restricted
deb-src http://archive.ubuntu.com/ubuntu focal main restricted #Added by software-properties

# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://kr.archive.ubuntu.com/ubuntu/ focal main restricted
deb-src http://kr.archive.ubuntu.com/ubuntu/ focal restricted universe main multiverse

## Major bug fix updates produced after the final release of the
## distribution.
deb http://kr.archive.ubuntu.com/ubuntu/ focal-updates main restricted
deb-src http://kr.archive.ubuntu.com/ubuntu/ focal-updates restricted universe main multiverse

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://kr.archive.ubuntu.com/ubuntu/ focal universe
# deb-src http://kr.archive.ubuntu.com/ubuntu/ focal universe
deb http://kr.archive.ubuntu.com/ubuntu/ focal-updates universe
# deb-src http://kr.archive.ubuntu.com/ubuntu/ focal-updates universe

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://kr.archive.ubuntu.com/ubuntu/ focal multiverse
# deb-src http://kr.archive.ubuntu.com/ubuntu/ focal multiverse
deb http://kr.archive.ubuntu.com/ubuntu/ focal-updates multiverse
# deb-src http://kr.archive.ubuntu.com/ubuntu/ focal-updates multiverse

## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
"/etc/apt/sources.list" 52 lines --1%--                            1,1           Top
```

3. Kernel Configuration

- **Before compiling the kernel, we should do some configurations**
  - Find the kernel source code archive file that we downloaded previously, and unzip it:
    - Double click is OK
    - Or use command: `tar -xvf <tar file>`



```
syslab@syslab-VirtualBox:~$ wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.4.214.tar.xz
--2022-09-22 16:56:45--  https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.4.214.tar.xz
Resolving cdn.kernel.org (cdn.kernel.org)... 151.101.1.176, 151.101.65.176, 151.101.129.176, ...
Connecting to cdn.kernel.org (cdn.kernel.org)|151.101.1.176|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 112943292 (108M) [application/x-xz]
Saving to: ‘linux-5.4.214.tar.xz’

linux-5.4.214.tar.xz     100%[==================================>] 107.71M  11.3MB/s    in 9.6s

2022-09-22 16:56:55 (11.2 MB/s) - ‘linux-5.4.214.tar.xz’ saved [112943292/112943292]

syslab@syslab-VirtualBox:~$ tar -xf linux-5.4.214.tar.xz
syslab@syslab-VirtualBox:~$ ls
linux-5.4.214  linux-5.4.214.tar.xz
syslab@syslab-VirtualBox:~$ 
```

3. Kernel Configuration

- **Configuration is stored in the `.config` file under the source code root directory**

- **There are three ways to get a `.config` file**
  - Use the previously used `.config` files
    - Under `/boot` directory
  - Create a new one by yourself
    - By `make config` or `make menuconfig`
  - Use the ARCH default .config file
    - By `make defconfig`

3. Kernel Configuration

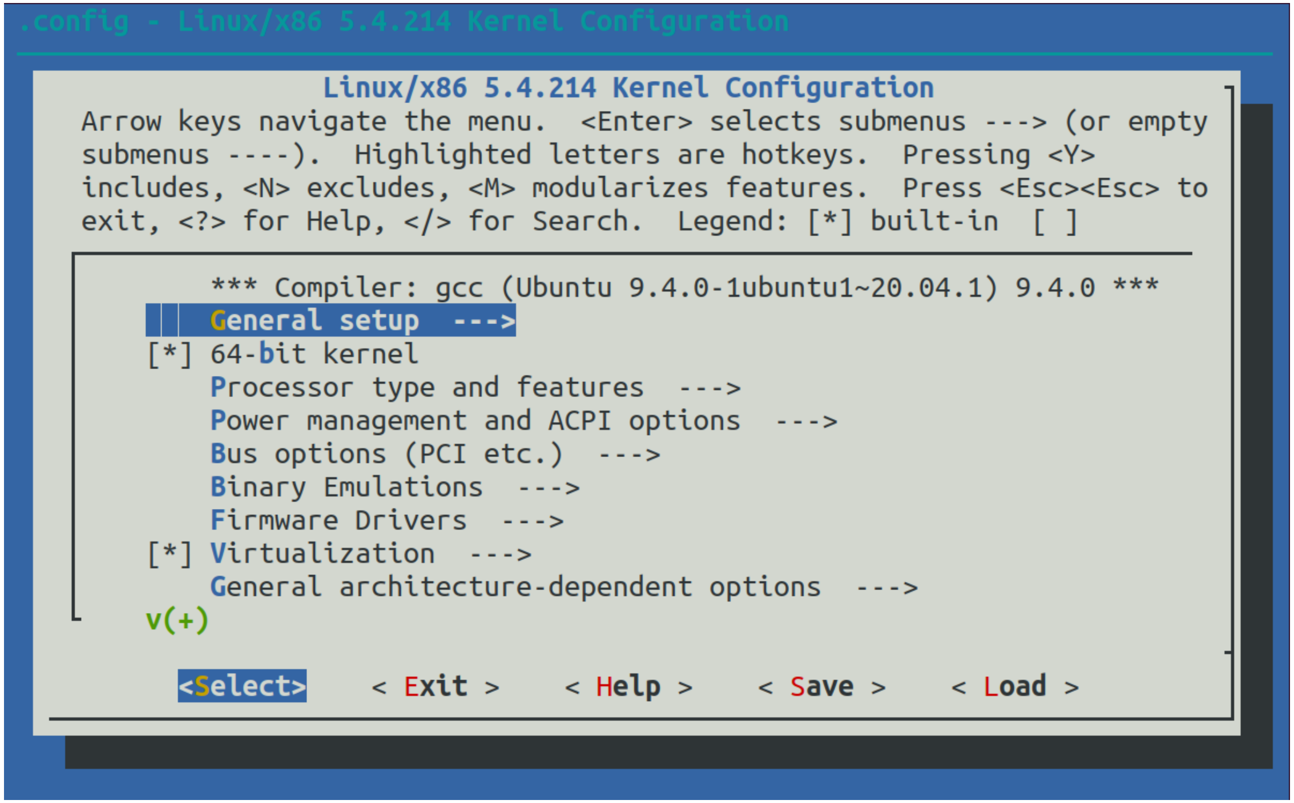
- **Here, as a practice, we will create a new `.config` file and only change the version name of the kernel.**
- **In the kernel code directory, use command:**

```
/[kernel source directory]$ make menuconfig
```

  - **P.S.** In real situations, using the distro-defined previous `.config` file is preferred.
    - This is because distros may support some different functionalities

3. Kernel Configuration

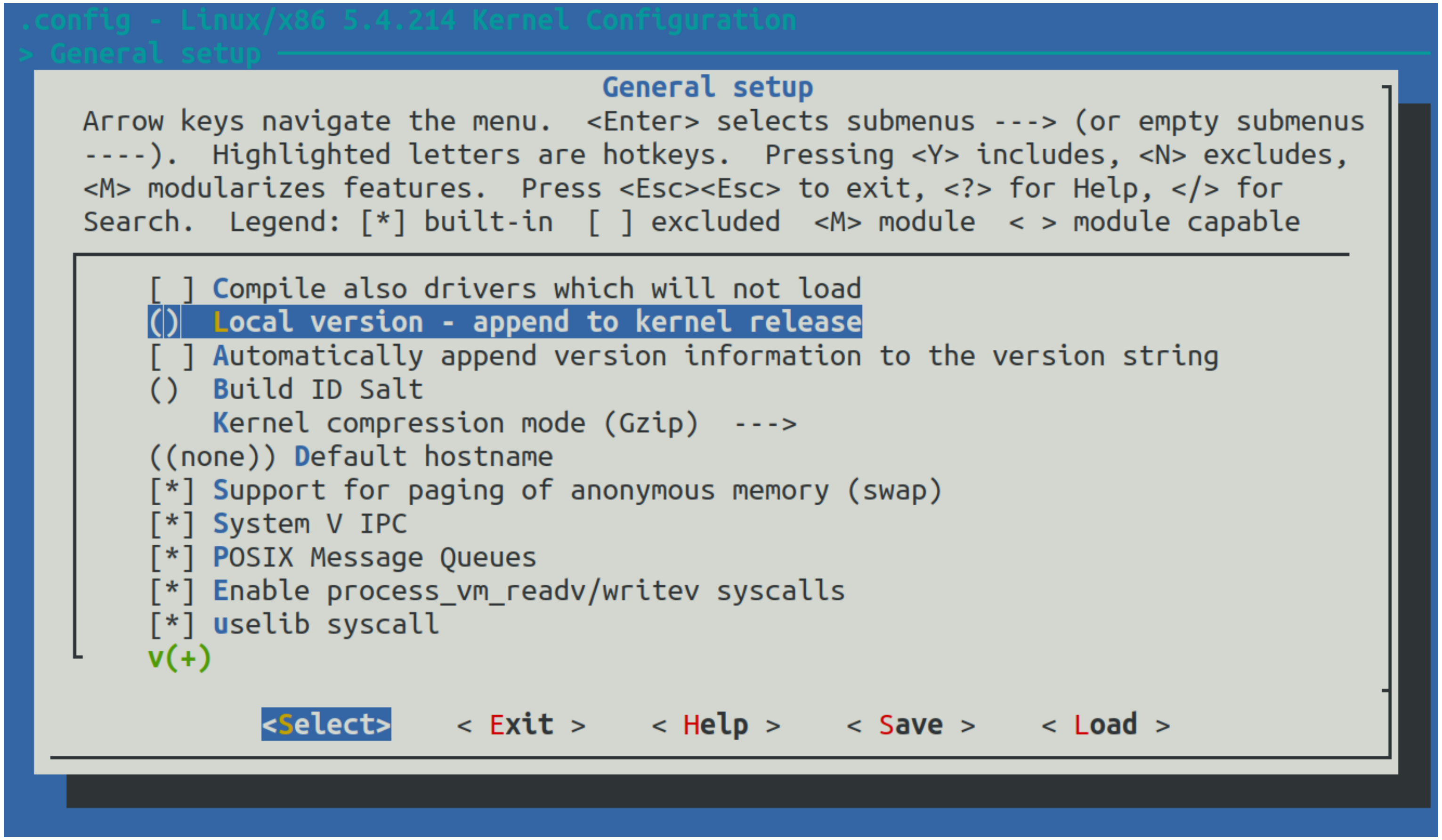
- This is the configuration window
  - Use **<Enter>** to get into the submenus
  - Use **<Esc><Esc>** to go back
  - Use **<Tab>** to choose the last line operations



```
.config - Linux/x86 5.4.214 Kernel Configuration

                 Linux/x86 5.4.214 Kernel Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
  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  [ ]

        *** Compiler: gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0 ***
        General setup  --->
    [*] 64-bit kernel
        Processor type and features  --->
        Power management and ACPI options  --->
        Bus options (PCI etc.)  --->
        Binary Emulations  --->
        Firmware Drivers  --->
    [*] Virtualization  --->
        General architecture-dependent options  --->
    v(+)

      <Select>    < Exit >    < Help >    < Save >    < Load >
```

3. Kernel Configuration

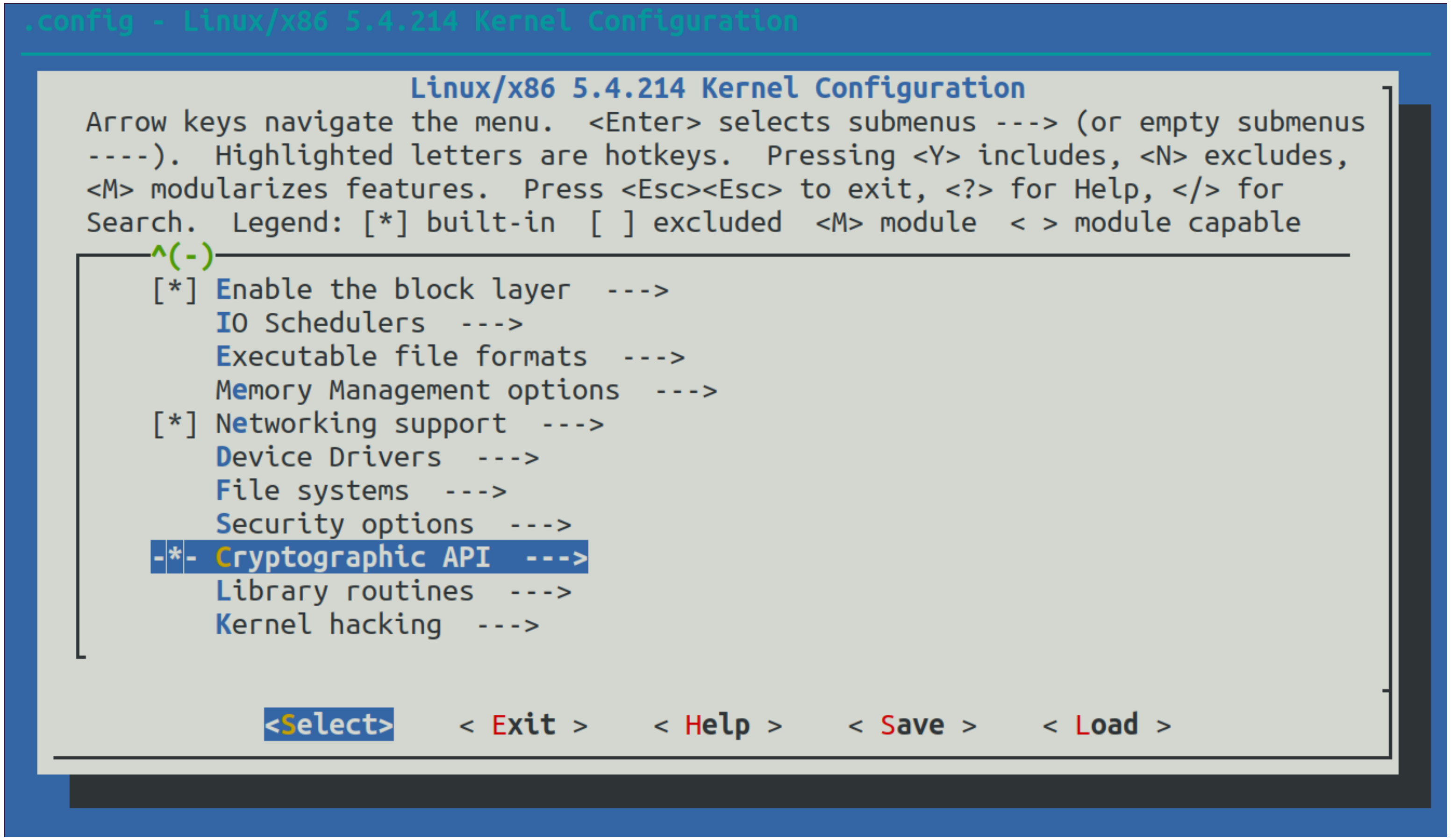
- **In the `General setup` submenu**
  - Select `Local version` option
  - We can set the name of the local kernel version here



```
.config - Linux/x86 5.4.214 Kernel Configuration
> General setup ─────────────────────────────────────────────────────
                          General setup
   Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty 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 capable

       [ ] Compile also drivers which will not load
       ()  Local version - append to kernel release
       [ ] Automatically append version information to the version string
       ()  Build ID Salt
           Kernel compression mode (Gzip)  --->
       ((none)) Default hostname
       [*] Support for paging of anonymous memory (swap)
       [*] System V IPC
       [*] POSIX Message Queues
       [*] Enable process_vm_readv/writev syscalls
       [*] uselib syscall
       v(+)

             <Select>    < Exit >    < Help >    < Save >    < Load >
```

3. Kernel Configuration

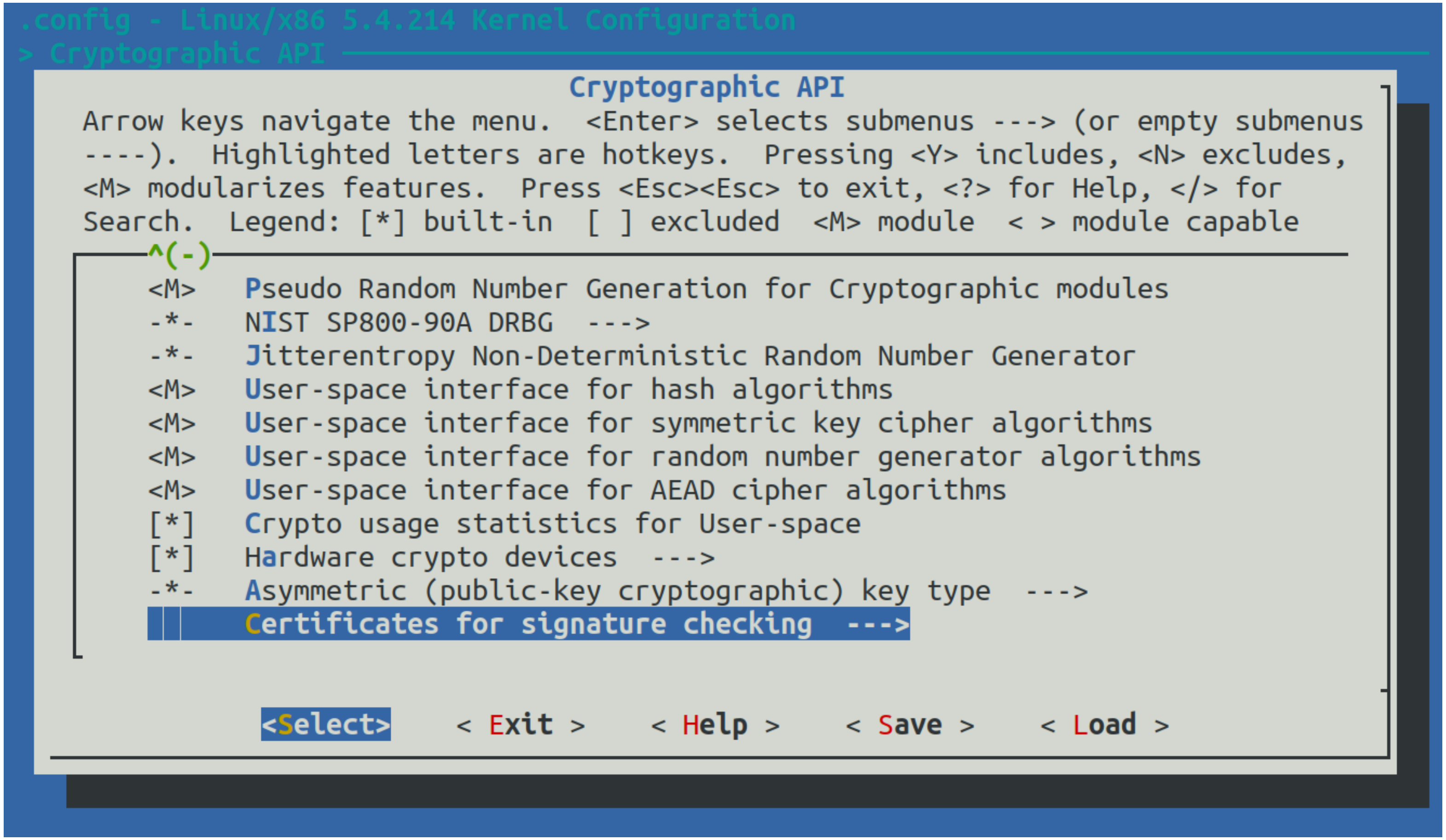
- We need to disable `Additional X.509 keys for system keyring` option
  - Select `Cryptographic API` from the top menu



```
.config - Linux/x86 5.4.214 Kernel Configuration

                    Linux/x86 5.4.214 Kernel Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty 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 capable
 ┌────^(-)──────────────────────────────────────────────────────────────────────
 │    [*] Enable the block layer  --->
 │        IO Schedulers  --->
 │        Executable file formats  --->
 │        Memory Management options  --->
 │    [*] Networking support  --->
 │        Device Drivers  --->
 │        File systems  --->
 │        Security options  --->
 │    -*- Cryptographic API  --->
 │        Library routines  --->
 │        Kernel hacking  --->
 └

            <Select>    < Exit >    < Help >    < Save >    < Load >
```

3. Kernel Configuration

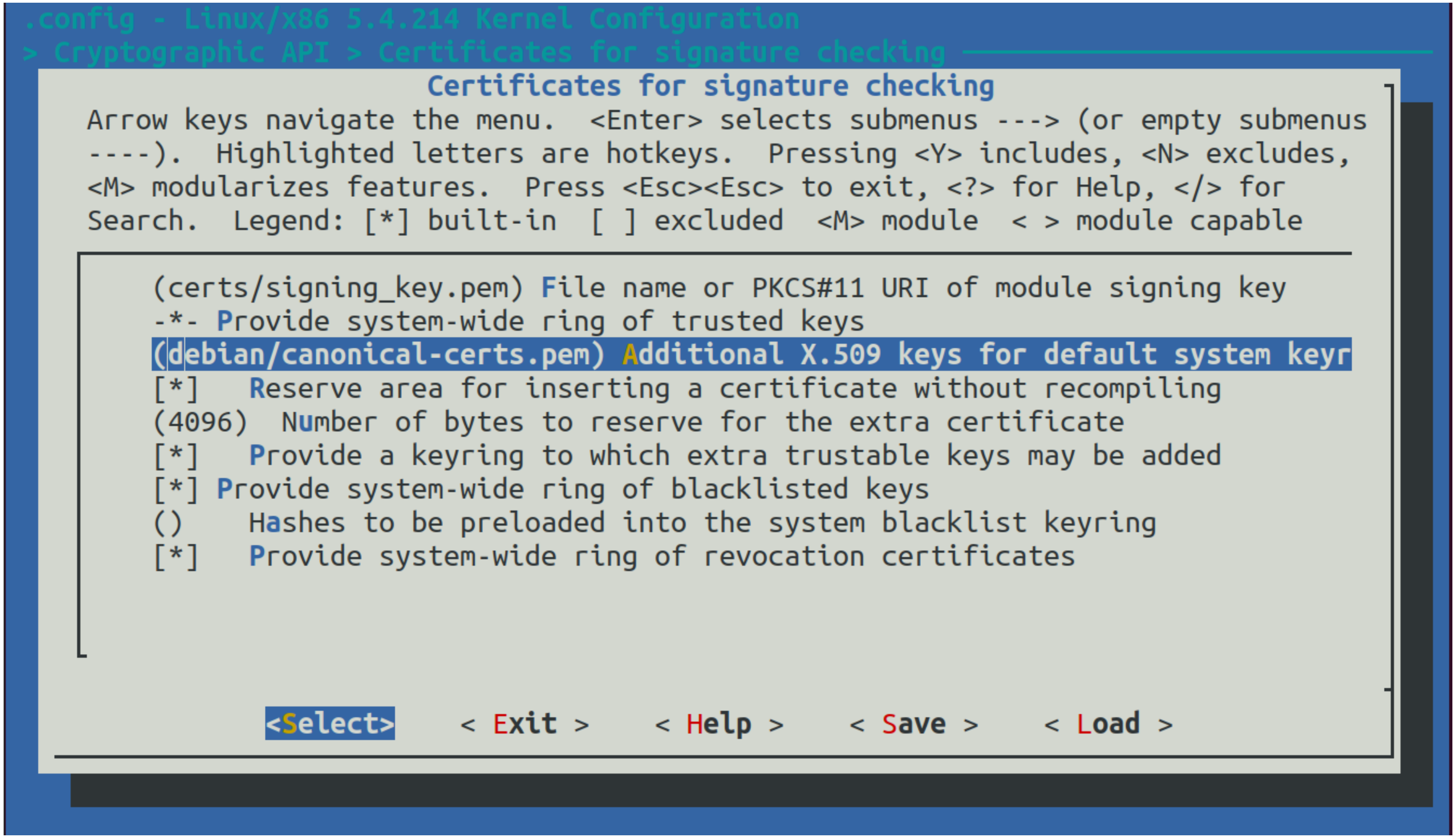
- In the `Cryptographic API` submenu
  - Select `Certificates for signature checking` (last entry)



```
.config - Linux/x86 5.4.214 Kernel Configuration
> Cryptographic API ──────────────────────────────────────────────────
                          Cryptographic API
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty 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 capable
 ┌────^(-)──────────────────────────────────────────────────────────────────┐
 │    <M>   Pseudo Random Number Generation for Cryptographic modules
 │    -*-   NIST SP800-90A DRBG  --->
 │    -*-   Jitterentropy Non-Deterministic Random Number Generator
 │    <M>   User-space interface for hash algorithms
 │    <M>   User-space interface for symmetric key cipher algorithms
 │    <M>   User-space interface for random number generator algorithms
 │    <M>   User-space interface for AEAD cipher algorithms
 │    [*]   Crypto usage statistics for User-space
 │    [*]   Hardware crypto devices  --->
 │    -*-   Asymmetric (public-key cryptographic) key type  --->
 │    ││    Certificates for signature checking  --->
 └

            <Select>    < Exit >    < Help >    < Save >    < Load >
```

3. Kernel Configuration

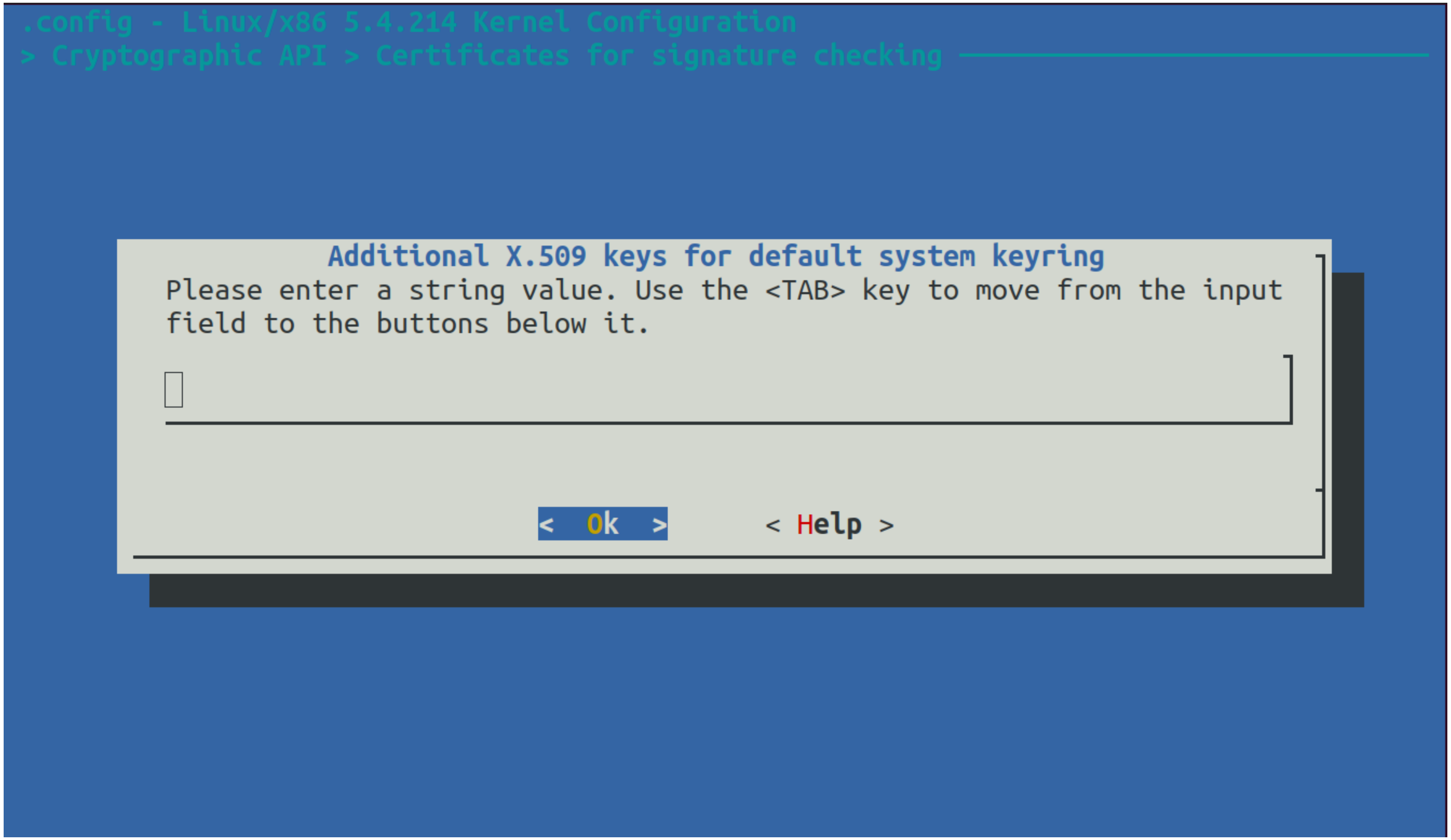
- In the `Certificates for signature checking` submenu
  - Select `Additional X.509 keys for default system keyring`



```
.config - Linux/x86 5.4.214 Kernel Configuration
> Cryptographic API > Certificates for signature checking ─────────────────
                    Certificates for signature checking
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty 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 capable

      (certs/signing_key.pem) File name or PKCS#11 URI of module signing key
      -*- Provide system-wide ring of trusted keys
      (debian/canonical-certs.pem) Additional X.509 keys for default system keyr
      [*]   Reserve area for inserting a certificate without recompiling
      (4096)  Number of bytes to reserve for the extra certificate
      [*]   Provide a keyring to which extra trustable keys may be added
      [*] Provide system-wide ring of blacklisted keys
      ()    Hashes to be preloaded into the system blacklist keyring
      [*]   Provide system-wide ring of revocation certificates


             <Select>    < Exit >    < Help >    < Save >    < Load >
```

3. Kernel Configuration

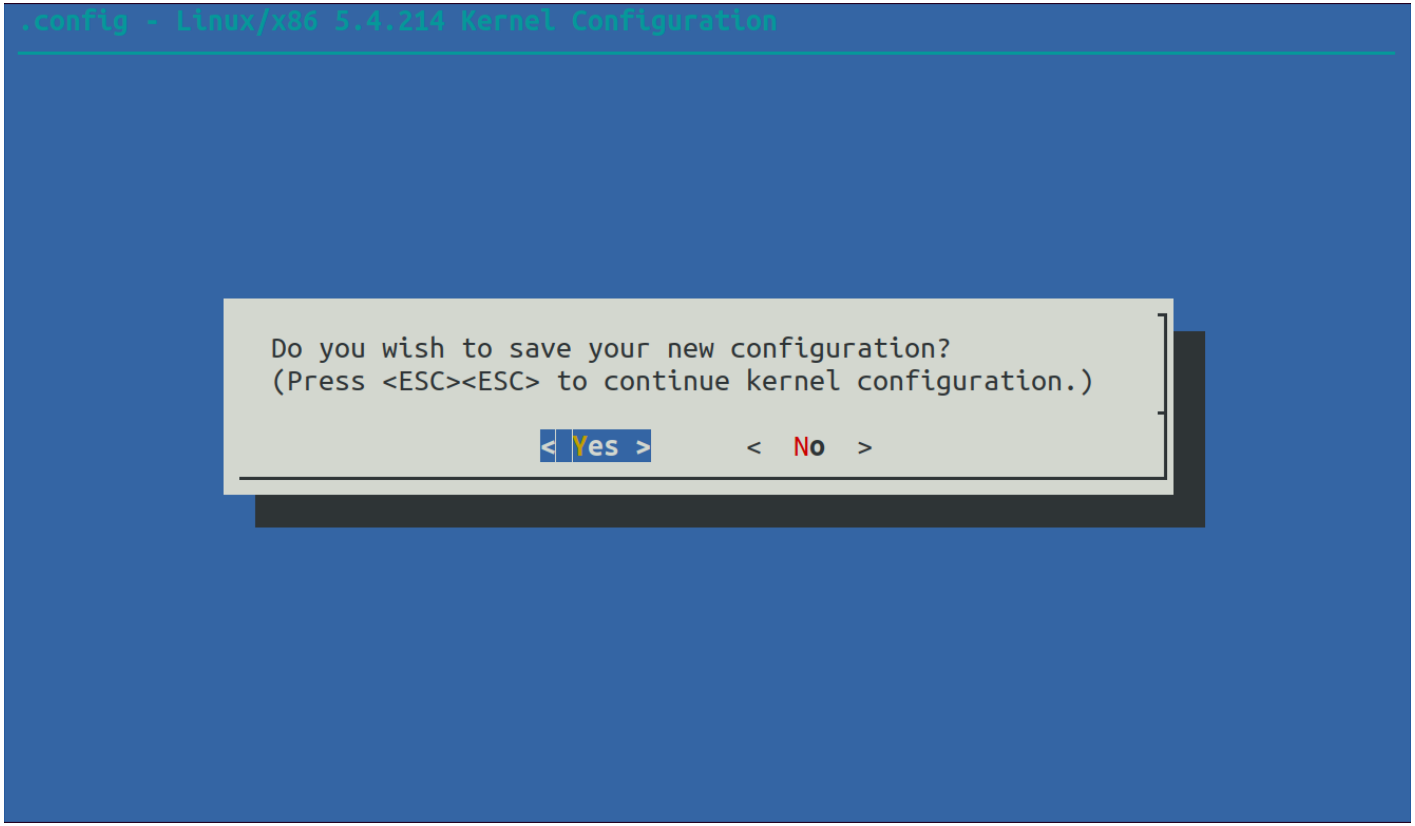
- Remove string "debian/canonical-certs.pem"



```
.config - Linux/x86 5.4.214 Kernel Configuration
> Cryptographic API > Certificates for signature checking ──────────────

          Additional X.509 keys for default system keyring
   Please enter a string value. Use the <TAB> key to move from the input
   field to the buttons below it.



                 <  Ok  >          < Help >
```

3. Kernel Configuration

- **After done, <Save> and <Exit> the configuration program**
  - Once we get the `.config` file, we can now compile the kernel



```
.config - Linux/x86 5.4.214 Kernel Configuration

 Do you wish to save your new configuration?
 (Press <ESC><ESC> to continue kernel configuration.)

                  < Yes >      <  No  >
```

4. Compile the Kernel

- **Kernel compiling is very simple, there is already a `Makefile` in the kernel code directory**

1. **The only thing we need to do is run the `make` command**

```
/[kernel source dir]$ make -j<number of processors>
```

- You can use `-j` option to specify the number of jobs to run simultaneously
  - Number of jobs depends on your processor
- Compiling the kernel may take a while

5. Kernel installation

2. **Once the kernel compiling is finished, we use the following command to build kernel modules**

```
$ make -j<number of processors> modules
```

3. **Install kernel modules that we built**

```
$ sudo make -j<number of processors> modules_install
```

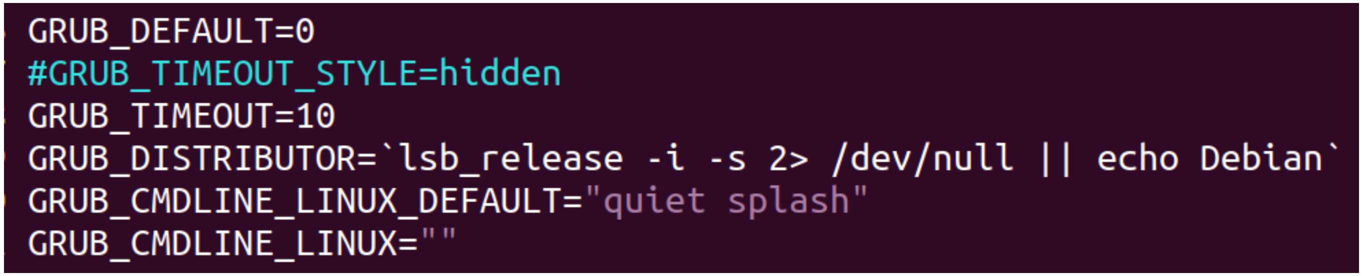
4. **And Install the kernel**

```
$ sudo make -j<number of processors> install
```

5. **When they are finished successfully, Kernel installation is done**
   - Now, if you reboot your machine, it will automatically use the latest (highest) version of the kernel.
     - But we will do some modifications to our **GRUB** boot loader before reboot.

6. Grub Configuration

- **`sudo vim /etc/default/grub`:**
  - Comment out **`GRUB_TIMEOUT_STYLE`**
    - ✓ **`GRUB_TIMEOUT_STYLE=hidden`** → **`#GRUB_TIMEOUT_STYLE=hidden`**
  - Set **`GRUB_TIMEOUT`**, the unit is second
  - Save the changes with **`:wq`** (you need root permission here)
  - Run command **`sudo update-grub`** every time you change this file

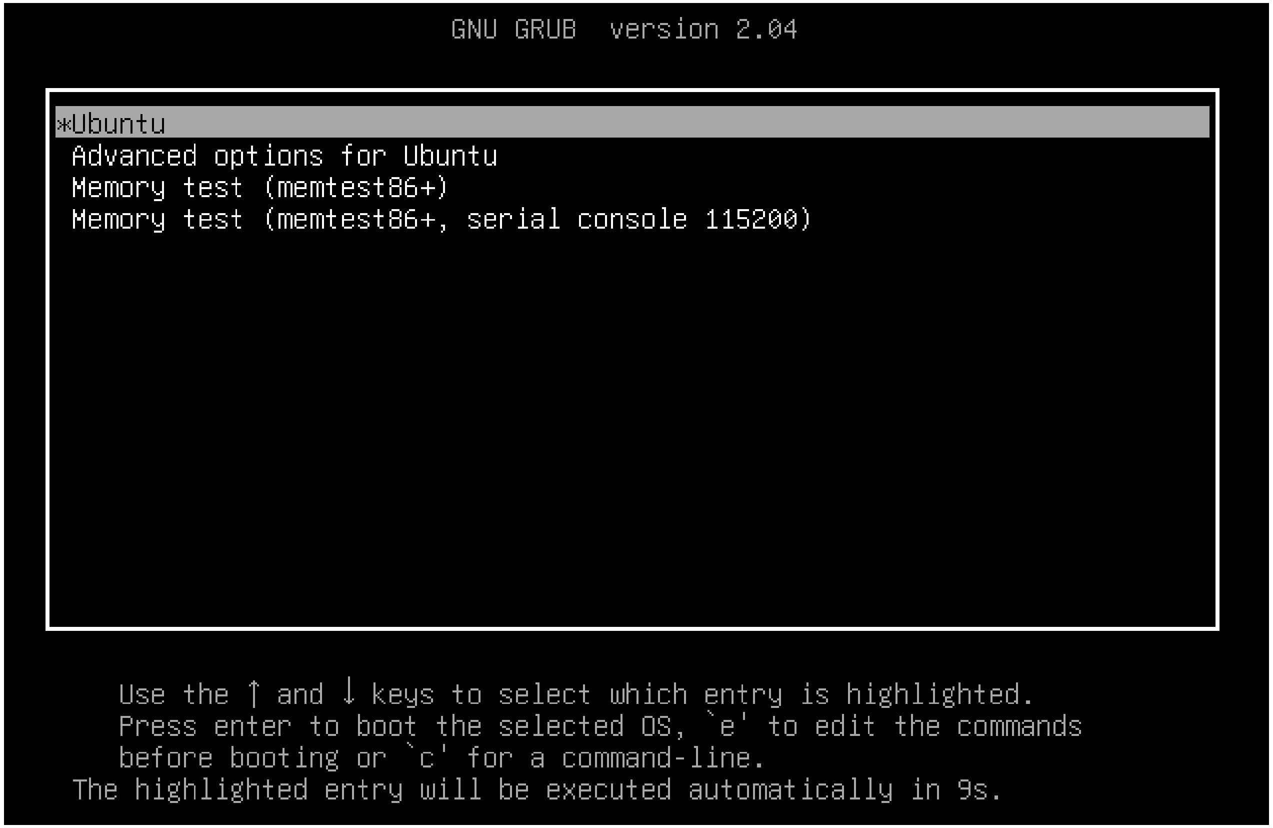


```
GRUB_DEFAULT=0
#GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""
```

- This setting can let us choose which kernel to use during the booting process
- If we did something wrong with the kernel, we can still use the old kernel to boot up the system

6. Grub Configuration

- Now, if we reboot the machine, we will see this kind of menu during the booting process.
- In the Advanced options for Ubuntu submenu, we can choose different kernels



```
                    GNU GRUB  version 2.04

 *Ubuntu
  Advanced options for Ubuntu
  Memory test (memtest86+)
  Memory test (memtest86+, serial console 115200)

     Use the ↑ and ↓ keys to select which entry is highlighted.
     Press enter to boot the selected OS, `e' to edit the commands
     before booting or `c' for a command-line.
  The highlighted entry will be executed automatically in 9s.
```

6. Grub Configuration

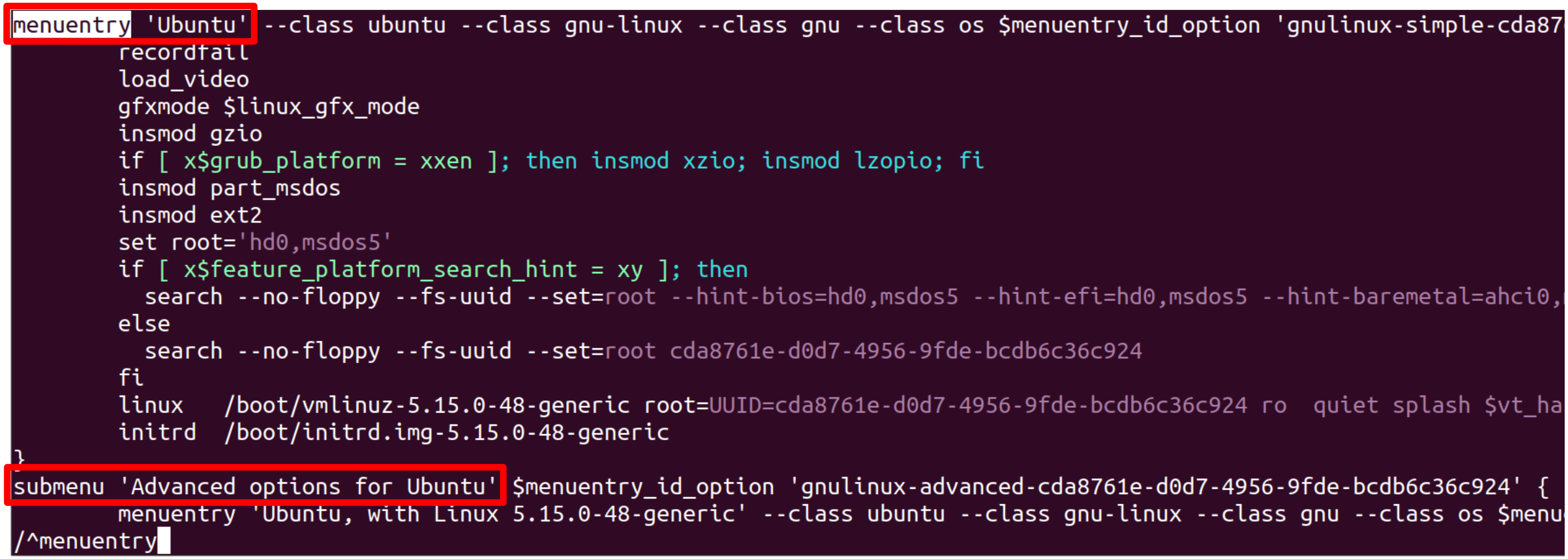
Default booting kernel

- When multiple kernels exist, it may be cumbersome to choose the kernel every time we boot our machine.
- We can use `/etc/default/grub` file’s `GRUB_DEFAULT` option to set the default kernel
- (You DO NOT need to do this if you do not need to change the default booting kernel)

6. Grub Configuration

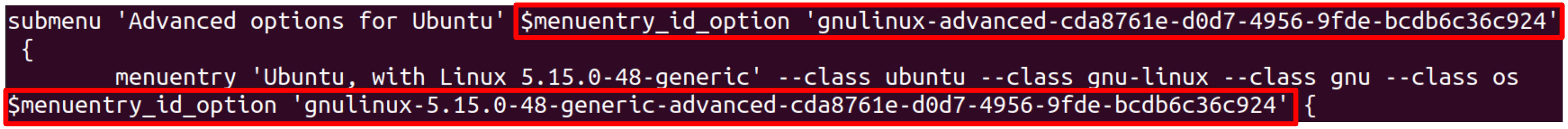
- **In file `/boot/grub/grub.cfg` file, we can find the GRUB menu entries**
  - If you use vim to open the file, you can use the search command to find the entry very easily:

```
/^menuentry
```



```
menuentry 'Ubuntu' --class ubuntu --class gnu-linux --class gnu --class os $menuentry_id_option 'gnulinux-simple-cda87
        recordfail
        load_video
        gfxmode $linux_gfx_mode
        insmod gzio
        if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
        insmod part_msdos
        insmod ext2
        set root='hd0,msdos5'
        if [ x$feature_platform_search_hint = xy ]; then
          search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos5 --hint-efi=hd0,msdos5 --hint-baremetal=ahci0,
        else
          search --no-floppy --fs-uuid --set=root cda8761e-d0d7-4956-9fde-bcdb6c36c924
        fi
        linux   /boot/vmlinuz-5.15.0-48-generic root=UUID=cda8761e-d0d7-4956-9fde-bcdb6c36c924 ro  quiet splash $vt_ha
        initrd  /boot/initrd.img-5.15.0-48-generic
}
submenu 'Advanced options for Ubuntu' $menuentry_id_option 'gnulinux-advanced-cda8761e-d0d7-4956-9fde-bcdb6c36c924' {
        menuentry 'Ubuntu, with Linux 5.15.0-48-generic' --class ubuntu --class gnu-linux --class gnu --class os $menu
/^menuentry
```

6. Grub Configuration

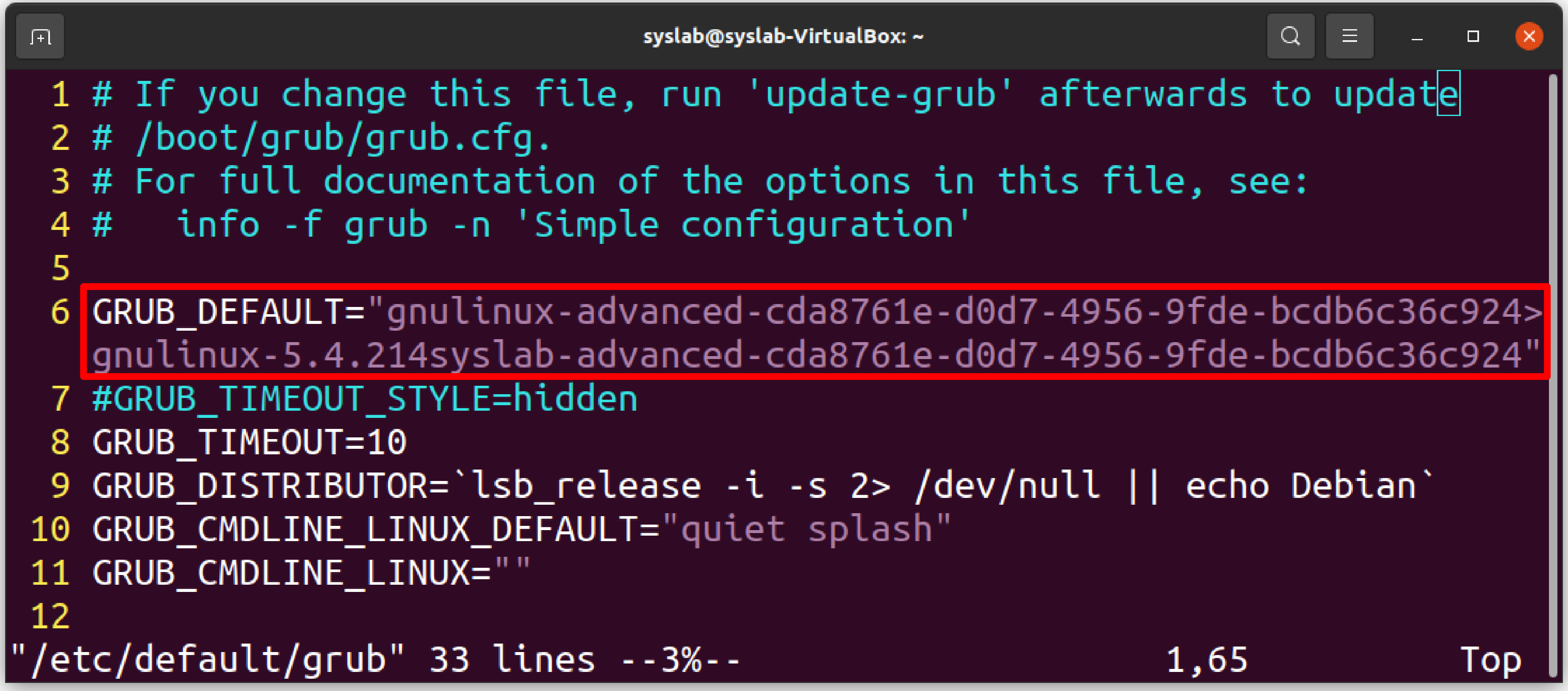


```
submenu 'Advanced options for Ubuntu' $menuentry_id_option 'gnulinux-advanced-cda8761e-d0d7-4956-9fde-bcdb6c36c924'
 {
        menuentry 'Ubuntu, with Linux 5.15.0-48-generic' --class ubuntu --class gnu-linux --class gnu --class os
$menuentry_id_option 'gnulinux-5.15.0-48-generic-advanced-cda8761e-d0d7-4956-9fde-bcdb6c36c924' {
```

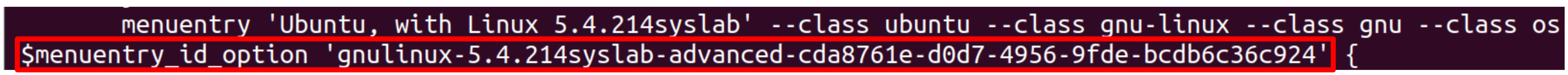
- **Everything start with `menuentry` is a single entry**
- **Here, we can know the first entry of the submenu `Advanced options for Ubuntu` is `‘Ubuntu, with Linux 5.15.0-48-generic’`**
- **The string after `$menuentry_id_option` is the id of the entry**
- **To use this kernel as a default booting kernel, change `GRUB_DEFAULT` option to `“submenu_id>entry_id”` (use > to connect)**
  - For example, here to use the first kernel as the default kernel, we change the setting (**`sudo vim /etc/default/grub`**) to:
    - `GRUB_DEFAULT=“gnulinux-advanced-cda8761e-d0d7-4956-9fde-bcdb6c36c924>gnulinux-5.15.0-48-generic-advanced-cda8761e-d0d7-4956-9fde-bcdb6c36c924”`
- **Do not forget to `sudo update-grub` after the modification.**

6. Grub Configuration

- **After you modify `/etc/default/grub`, the file would look like this:**



```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT="gnulinux-advanced-cda8761e-d0d7-4956-9fde-bcdb6c36c924>gnulinux-5.4.214syslab-advanced-cda8761e-d0d7-4956-9fde-bcdb6c36c924"
#GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

"/etc/default/grub" 33 lines --3%--                    1,65          Top
```

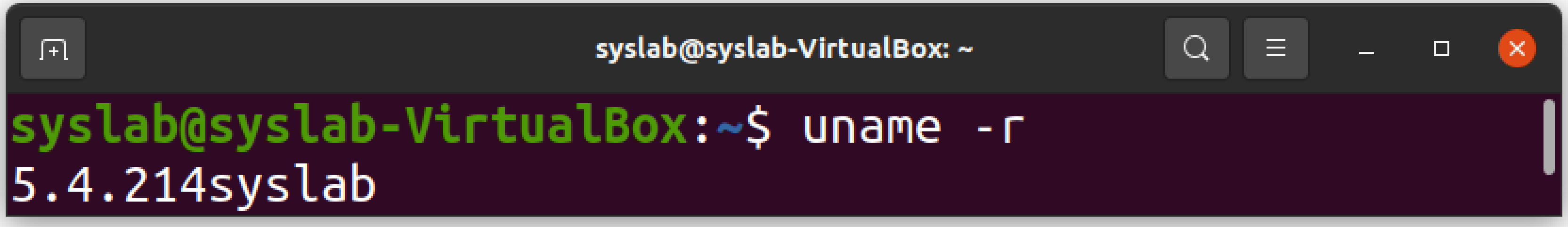


```
        menuentry 'Ubuntu, with Linux 5.4.214syslab' --class ubuntu --class gnu-linux --class gnu --class os
$menuentry_id_option 'gnulinux-5.4.214syslab-advanced-cda8761e-d0d7-4956-9fde-bcdb6c36c924' {
```

7. Boot into the Kernel

- After booting into your custom kernel, you can check the current kernel with

```
$ uname -r
```

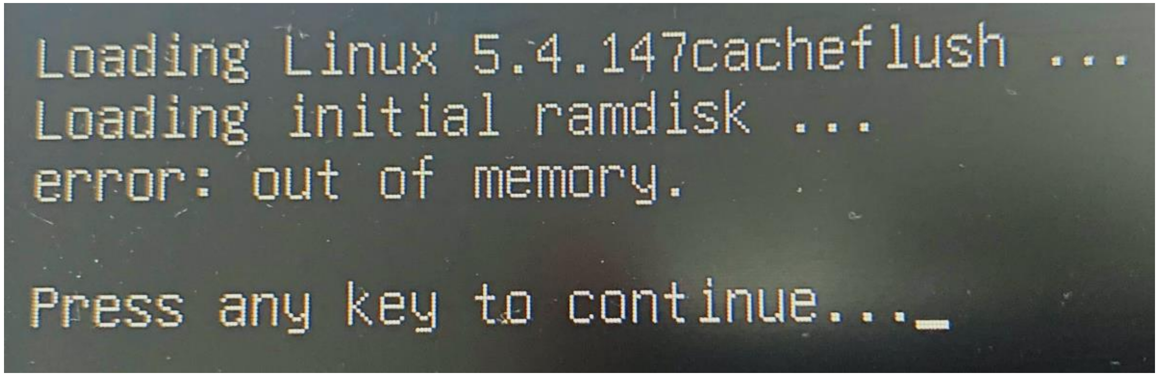


```
syslab@syslab-VirtualBox:~$ uname -r
5.4.214syslab
```

Troubleshooting 1

Issue

- When tring booting into a custom kernel
- Failed while loading initial ramdisk



```
Loading Linux 5.4.147cacheflush ...
Loading initial ramdisk ...
error: out of memory.

Press any key to continue..._
```

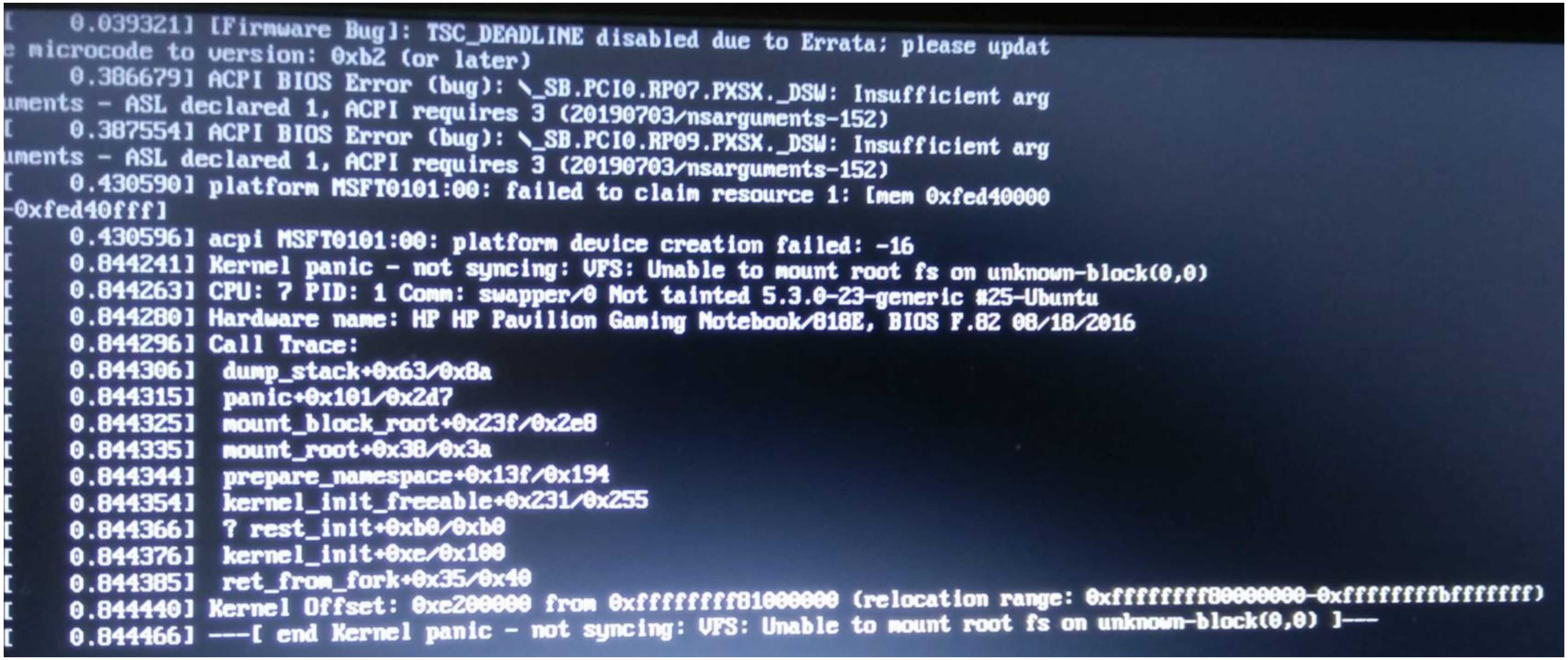
Solution

- Add option **INSTALL_MOD_STRIP** when installing built kernel modules

```
$ sudo make INSTALL_MOD_STRIP=1 -j<number of processors> modules_install
```

Troubleshooting 2

```
[0.000000] [Firmware Bug]: TSC_DEADLINE disabled due to Errata; please
update microdoce to version: 0x00 (or later)
```



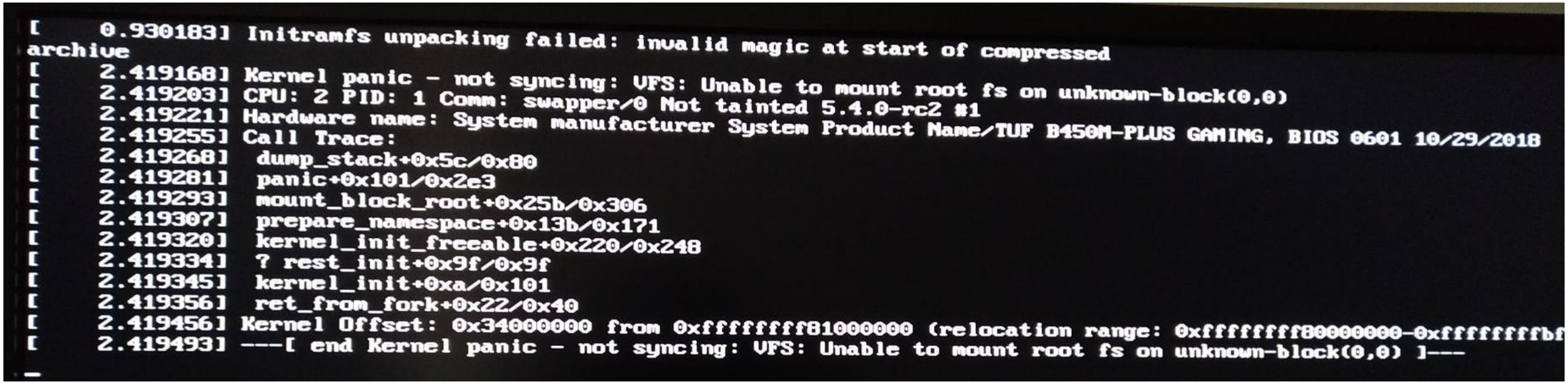
```
[    0.039321] [Firmware Bug]: TSC_DEADLINE disabled due to Errata; please updat
e microcode to version: 0xb2 (or later)
[    0.386679] ACPI BIOS Error (bug): \_SB.PCI0.RP07.PXSX._DSW: Insufficient arg
uments - ASL declared 1, ACPI requires 3 (20190703/nsarguments-152)
[    0.387554] ACPI BIOS Error (bug): \_SB.PCI0.RP09.PXSX._DSW: Insufficient arg
uments - ASL declared 1, ACPI requires 3 (20190703/nsarguments-152)
[    0.430590] platform MSFT0101:00: failed to claim resource 1: [mem 0xfed40000
-0xfed40fff]
[    0.430596] acpi MSFT0101:00: platform device creation failed: -16
[    0.844241] Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(0,0)
[    0.844263] CPU: 7 PID: 1 Comm: swapper/0 Not tainted 5.3.0-23-generic #25-Ubuntu
[    0.844280] Hardware name: HP HP Pavilion Gaming Notebook/818E, BIOS F.02 08/18/2016
[    0.844296] Call Trace:
[    0.844306]  dump_stack+0x63/0x8a
[    0.844315]  panic+0x101/0x2d7
[    0.844325]  mount_block_root+0x23f/0x2e8
[    0.844335]  mount_root+0x38/0x3a
[    0.844344]  prepare_namespace+0x13f/0x194
[    0.844354]  kernel_init_freeable+0x231/0x255
[    0.844366]  ? rest_init+0xb0/0xb0
[    0.844376]  kernel_init+0xe/0x100
[    0.844385]  ret_from_fork+0x35/0x40
[    0.844440] Kernel Offset: 0xe200000 from 0xffffffff81000000 (relocation range: 0xffffffff80000000-0xffffffffbfffffff)
[    0.844466] ---[ end Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(0,0) ]---
```

Solution

- Update motherboard **BIOS firmware** to current version

Troubleshooting 3

```
[0.000000] Initramfs unpacking failed: invalid magic at start of compressed
```

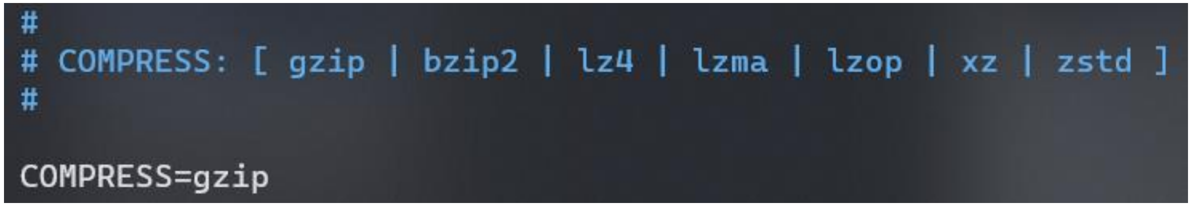


```
[    0.930183] Initramfs unpacking failed: invalid magic at start of compressed
archive
[    2.419168] Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(0,0)
[    2.419203] CPU: 2 PID: 1 Comm: swapper/0 Not tainted 5.4.0-rc2 #1
[    2.419221] Hardware name: System manufacturer System Product Name/TUF B450M-PLUS GAMING, BIOS 0601 10/29/2018
[    2.419255] Call Trace:
[    2.419268]  dump_stack+0x5c/0x80
[    2.419281]  panic+0x101/0x2e3
[    2.419293]  mount_block_root+0x25b/0x306
[    2.419307]  prepare_namespace+0x13b/0x171
[    2.419320]  kernel_init_freeable+0x220/0x248
[    2.419334]  ? rest_init+0x9f/0x9f
[    2.419345]  kernel_init+0xa/0x101
[    2.419356]  ret_from_fork+0x22/0x40
[    2.419456] Kernel Offset: 0x34000000 from 0xffffffff81000000 (relocation range: 0xffffffff80000000-0xffffffffbf
[    2.419493] ---[ end Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(0,0) ]---
_
```

❖ Solution

`/etc/initramfs-tools/initramfs.conf`

- Change **initramfs-tools** configuration
  - COMPRESS option into “**`gzip`**”
  - After modification, reimage initrd with `sudo update-initramfs`

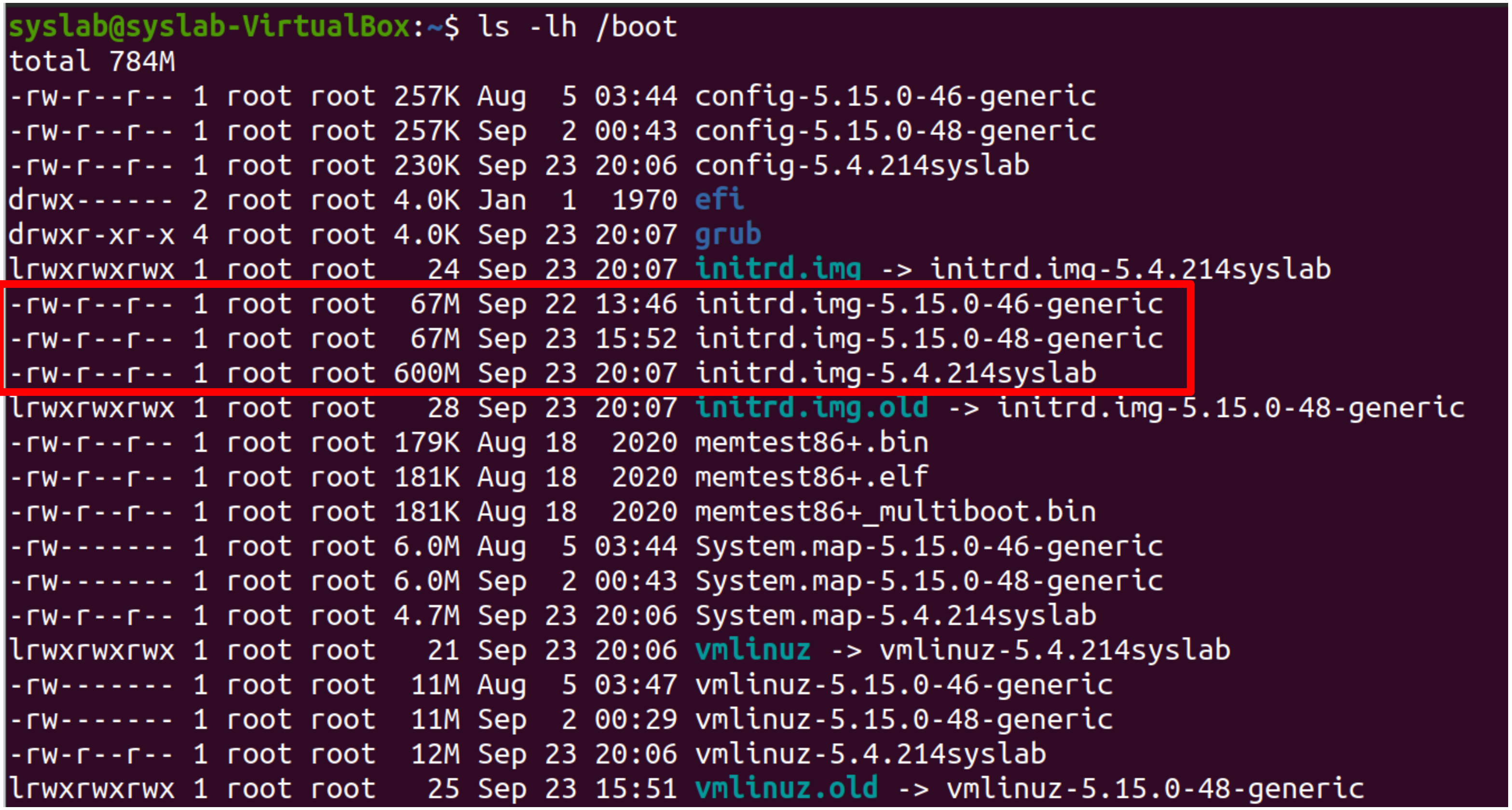


```
#
# COMPRESS: [ gzip | bzip2 | lz4 | lzma | lzop | xz | zstd ]
#

COMPRESS=gzip
```

Troubleshooting 3

❖ Initial ramdisk?

- A scheme for loading a temporary root file system into memory, to be used as part of the Linux startup process
- Resides in **`/boot`** directory



```
syslab@syslab-VirtualBox:~$ ls -lh /boot
total 784M
-rw-r--r-- 1 root root 257K Aug  5 03:44 config-5.15.0-46-generic
-rw-r--r-- 1 root root 257K Sep  2 00:43 config-5.15.0-48-generic
-rw-r--r-- 1 root root 230K Sep 23 20:06 config-5.4.214syslab
drwx------ 2 root root 4.0K Jan  1  1970 efi
drwxr-xr-x 4 root root 4.0K Sep 23 20:07 grub
lrwxrwxrwx 1 root root   24 Sep 23 20:07 initrd.img -> initrd.img-5.4.214syslab
-rw-r--r-- 1 root root  67M Sep 22 13:46 initrd.img-5.15.0-46-generic
-rw-r--r-- 1 root root  67M Sep 23 15:52 initrd.img-5.15.0-48-generic
-rw-r--r-- 1 root root 600M Sep 23 20:07 initrd.img-5.4.214syslab
lrwxrwxrwx 1 root root   28 Sep 23 20:07 initrd.img.old -> initrd.img-5.15.0-48-generic
-rw-r--r-- 1 root root 179K Aug 18  2020 memtest86+.bin
-rw-r--r-- 1 root root 181K Aug 18  2020 memtest86+.elf
-rw-r--r-- 1 root root 181K Aug 18  2020 memtest86+_multiboot.bin
-rw------- 1 root root 6.0M Aug  5 03:44 System.map-5.15.0-46-generic
-rw------- 1 root root 6.0M Sep  2 00:43 System.map-5.15.0-48-generic
-rw-r--r-- 1 root root 4.7M Sep 23 20:06 System.map-5.4.214syslab
lrwxrwxrwx 1 root root   21 Sep 23 20:06 vmlinuz -> vmlinuz-5.4.214syslab
-rw------- 1 root root  11M Aug  5 03:47 vmlinuz-5.15.0-46-generic
-rw------- 1 root root  11M Sep  2 00:29 vmlinuz-5.15.0-48-generic
-rw-r--r-- 1 root root  12M Sep 23 20:06 vmlinuz-5.4.214syslab
lrwxrwxrwx 1 root root   25 Sep 23 15:51 vmlinuz.old -> vmlinuz-5.15.0-48-generic
```