

COMPILING THE LATEST STABLE KERNEL AND DUAL BOOTING IT WITH THE CURRENT VERSION

Submitted By:

Kolli Jogi Naidu

BI90605CS

TABLE OF CONTENTS:

1. Introduction
2. Problem Statement
3. Methodology
4. Explanation with Screenshots
5. Summary
6. References

I. Introduction:

What is kernel?

Kernel is central component of operating system that manages the operations of computer and hardware. It establishes communication between user level application and hardware. It decides state of incoming processes. It plays key role in disk management, memory management, task management.

2. Problem Statement:

Download the latest stable Linux kernel from kernel.org, compile it, and dual boot it with your current Linux version. Your current version as well as the new version should be present in the grub-menu.

3. Methodology:

The Latest Linux kernel compilation and installation includes the following steps:

1. Downloading the kernel
2. Extracting the source
3. Configuring the kernel
4. Installing the requirements
5. Compiling and installing the kernel
6. Grub menu

4. Explanation with Screenshots:

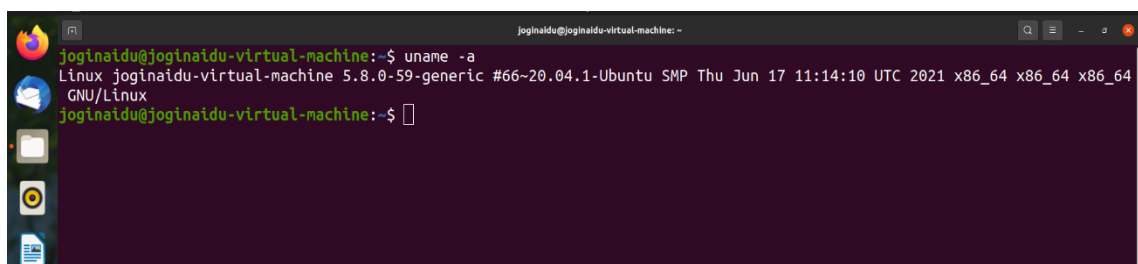
Step 1: Downloading the kernel

- a. Check the current system Information by running the command:

`uname -a`

For checking the current version enter the following command:

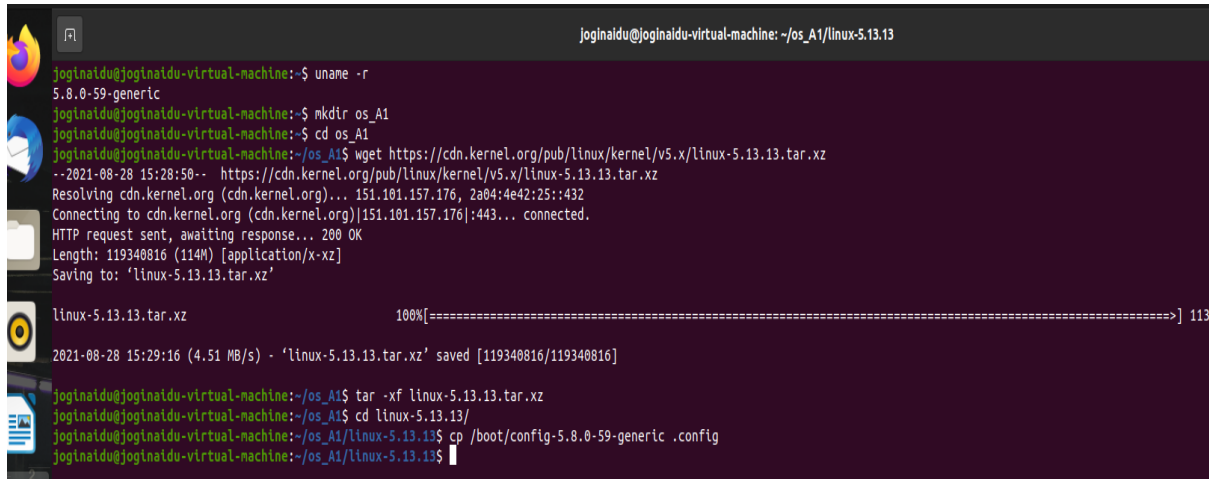
`uname -r`



```
joginaidu@joginaidu-virtual-machine: ~  
joginaidu@joginaidu-virtual-machine:~$ uname -a  
Linux joginaidu-virtual-machine 5.8.0-59-generic #66~20.04.1-Ubuntu SMP Thu Jun 17 11:14:10 UTC 2021 x86_64 x86_64 x86_64  
GNU/Linux  
joginaidu@joginaidu-virtual-machine:~$
```

- b. Now download the latest kernel source file from kernel.org with the following command in the terminal (**The latest kernel now available is 5.13.13**):

wget <https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.13.13.tar.xz>



```
joginaidu@joginaidu-virtual-machine: ~/os_A1/linux-5.13.13
joginaidu@joginaidu-virtual-machine:~$ uname -r
5.8.0-59-generic
joginaidu@joginaidu-virtual-machine:~$ mkdir os_A1
joginaidu@joginaidu-virtual-machine:~$ cd os_A1
joginaidu@joginaidu-virtual-machine:~/os_A1$ wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.13.13.tar.xz
--2021-08-28 15:28:50-- https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.13.13.tar.xz
Resolving cdn.kernel.org (cdn.kernel.org)... 151.101.157.176, 2a04:4e42:25::432
Connecting to cdn.kernel.org (cdn.kernel.org)|151.101.157.176|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 119340816 (114M) [application/x-xz]
Saving to: 'linux-5.13.13.tar.xz'

linux-5.13.13.tar.xz      100%[=====] 113
2021-08-28 15:29:16 (4.51 MB/s) - 'linux-5.13.13.tar.xz' saved [119340816/119340816]

joginaidu@joginaidu-virtual-machine:~/os_A1$ tar -xvf linux-5.13.13.tar.xz
joginaidu@joginaidu-virtual-machine:~/os_A1$ cd linux-5.13.13/
joginaidu@joginaidu-virtual-machine:~/os_A1/linux-5.13.13$ cp /boot/config-5.8.0-59-generic .config
joginaidu@joginaidu-virtual-machine:~/os_A1/linux-5.13.13$
```

Step 2: Extracting the source

- a. Extract the kernel source with the command within our newly downloaded kernel directory:

tar -xvf linux-5.13.13.tar.xz

- b. Change into the newly created directory with the following command:

cd linux-5.13.13/

step3: Configuring the kernel

Before compiling the kernel, we need to install required modules to include in it. With a single command we can copy the current kernel's config file and then use the tried-and-true menuconfig command to make necessary changes.

We can do this by copying the existing kernel config file using the following command:

cp/boot/config-\$(uname -r) .config

(Since the version might be slightly different, give `uname -r`)

To configure the kernel:

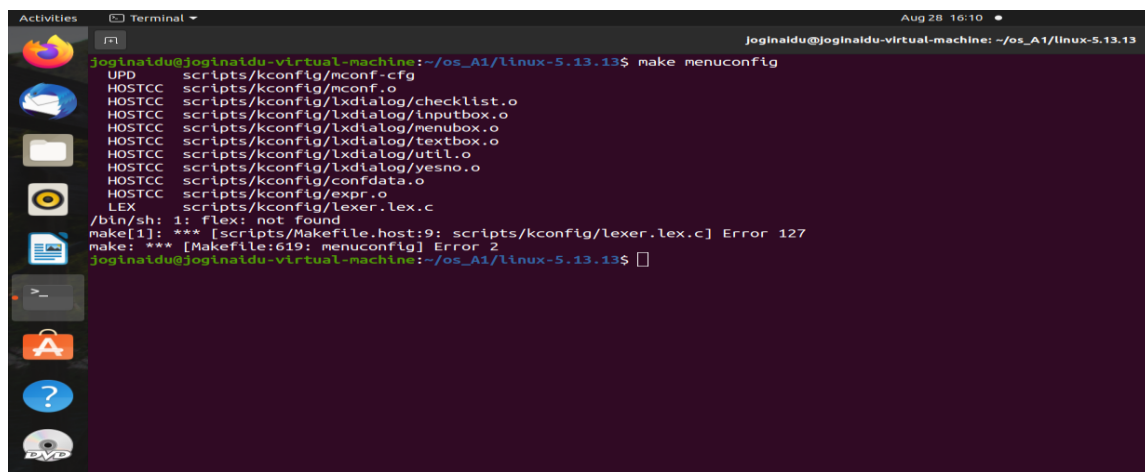
We need to configure before compiling it because we need to specify which modules we want to be installed and which to not. We have the option of just

sticking with the configuration offered by the kernel or to alter the existing configuration or add your own, we will use the default configuration.

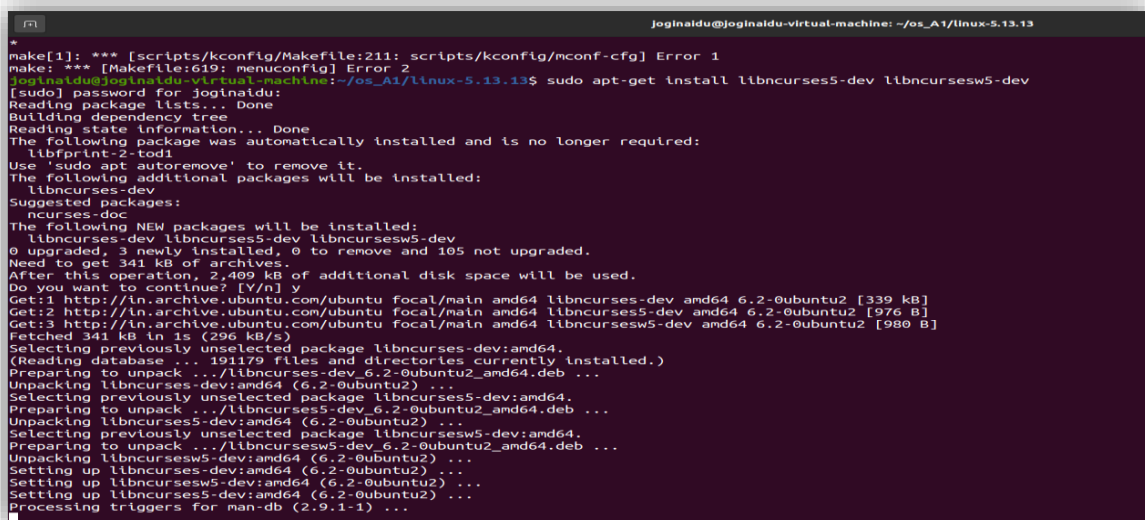
The following command will configure the kernel:

make menuconfig

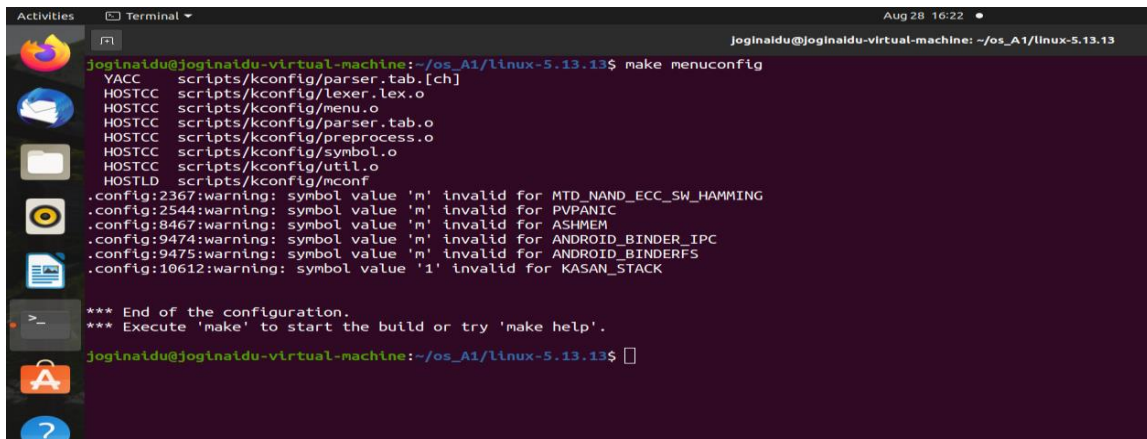
This command will open up a tool that allows us to go through every module available and enable or disable what we need or not. If any errors are prompted try figuring out by installing necessary packages to configure the kernel. Save config file in .config and exit the text-based menu.



```
Aug 28 16:10
joginaidu@joginaidu-virtual-machine: ~/os_A1/linux-5.13.13
joginaidu@joginaidu-virtual-machine:~/os_A1/linux-5.13.13$ make menuconfig
UPD      scripts/kconfig/mconf.cfg
HOSTCC   scripts/kconfig/mconf.o
HOSTCC   scripts/kconfig/Lxdialog/checklist.o
HOSTCC   scripts/kconfig/Lxdialog/inputbox.o
HOSTCC   scripts/kconfig/Lxdialog/checkbox.o
HOSTCC   scripts/kconfig/Lxdialog/textbox.o
HOSTCC   scripts/kconfig/Lxdialog/util.o
HOSTCC   scripts/kconfig/Lxdialog/yesno.o
HOSTCC   scripts/kconfig/confdata.o
HOSTCC   scripts/kconfig/expr.o
LEX       scripts/kconfig/lexer.lex.c
/bin/sh: 1: flex: not found
make[1]: *** [scripts/Makefile.host:9: scripts/kconfig/lexer.lex.c] Error 127
make: *** [Makefile:619: menuconfig] Error 2
joginaidu@joginaidu-virtual-machine:~/os_A1/linux-5.13.13$
```



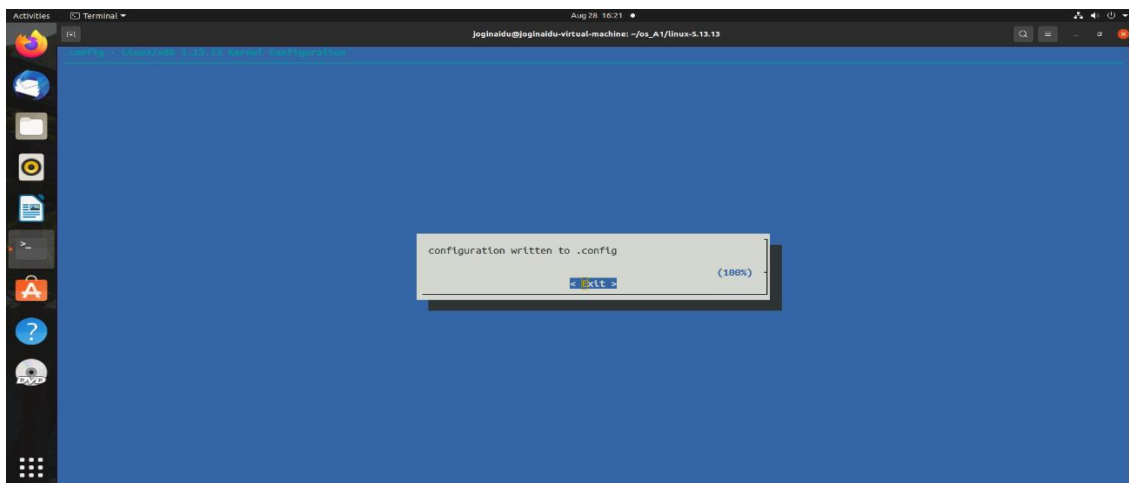
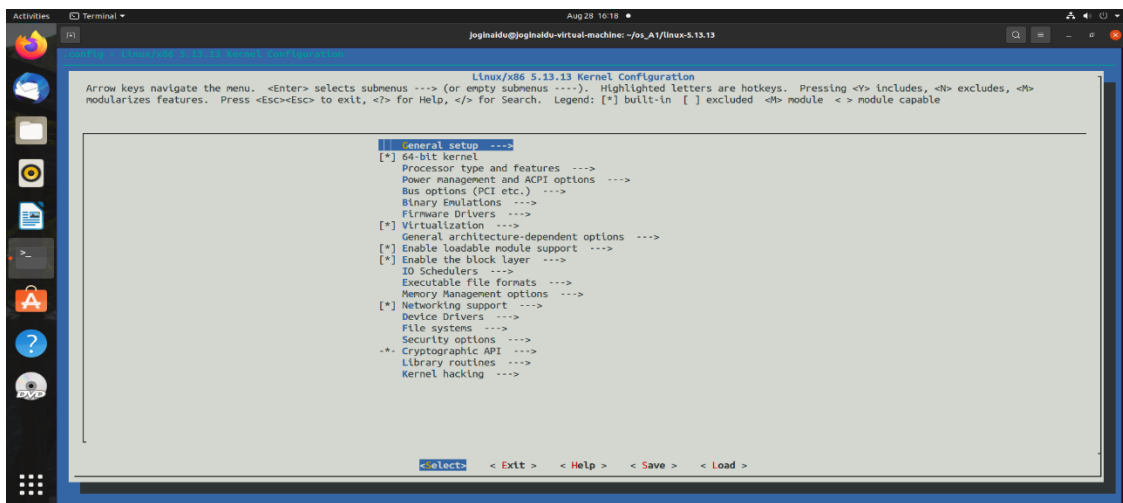
```
joginaidu@joginaidu-virtual-machine: ~/os_A1/linux-5.13.13
*
make[1]: *** [scripts/kconfig/Makefile:211: scripts/kconfig/mconf.cfg] Error 1
make: *** [Makefile:619: menuconfig] Error 2
joginaidu@joginaidu-virtual-machine:~/os_A1/linux-5.13.13$ sudo apt-get install libncurses5-dev libncursesw5-dev
[sudo] password for joginaidu:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libncurses5-dev
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libncursesw5-dev
Suggested packages:
  ncurses-doc
The following NEW packages will be installed:
  libncurses5-dev libncursesw5-dev
0 upgraded, 3 newly installed, 0 to remove and 105 not upgraded.
Need to get 341 kB of archives.
After this operation, 2,409 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libncurses-dev amd64 6.2-0ubuntu2 [339 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libncurses5-dev amd64 6.2-0ubuntu2 [976 B]
Get:3 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libncursesw5-dev amd64 6.2-0ubuntu2 [980 B]
Fetched 341 kB in 1s (296 kB/s)
Selecting previously unselected package libncurses-dev:amd64.
(Reading database ... 101179 files and directories currently installed.)
Preparing to unpack .../libncurses-dev_6.2-0ubuntu2_amd64.deb ...
Unpacking libncurses-dev:amd64 (6.2-0ubuntu2) ...
Selecting previously unselected package libncurses5-dev:amd64.
Preparing to unpack .../libncurses5-dev_6.2-0ubuntu2_amd64.deb ...
Unpacking libncurses5-dev:amd64 (6.2-0ubuntu2) ...
Selecting previously unselected package libncursesw5-dev:amd64.
Preparing to unpack .../libncursesw5-dev_6.2-0ubuntu2_amd64.deb ...
Unpacking libncursesw5-dev:amd64 (6.2-0ubuntu2) ...
Setting up libncurses5-dev:amd64 (6.2-0ubuntu2) ...
Setting up libncursesw5-dev:amd64 (6.2-0ubuntu2) ...
Processing triggers for man-db (2.9.1-1) ...
```



```
Aug 28 16:22
joginaidu@joginaidu-virtual-machine: ~/os_A1/linux-5.13.13
joginaidu@joginaidu-virtual-machine:~/os_A1/linux-5.13.13$ make menuconfig
YACC    scripts/kconfig/parser.tab.[ch]
HOSTCC  scripts/kconfig/lexer.lex.o
HOSTCC  scripts/kconfig/menu.o
HOSTCC  scripts/kconfig/parser.tab.o
HOSTCC  scripts/kconfig/preprocess.o
HOSTCC  scripts/kconfig/symbol.o
HOSTCC  scripts/kconfig/util.o
HOSTLD  scripts/kconfig/mconf
.config:2367:warning: symbol value 'm' invalid for MTD_NAND_ECC_SW_HAMMING
.config:2544:warning: symbol value 'm' invalid for PVPANIC
.config:8467:warning: symbol value 'm' invalid for ASHMEM
.config:9474:warning: symbol value 'm' invalid for ANDROID_BINDER_IPC
.config:9475:warning: symbol value 'm' invalid for ANDROID_BINDERFS
.config:10612:warning: symbol value '1' invalid for KASAN_STACK

*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.

joginaidu@joginaidu-virtual-machine:~/os_A1/linux-5.13.13$
```



Step 4: Installing the requirements

We need to install the requirements before the compilation of kernel. We use the following command:

```
sudo apt-get install git fakeroot build-essential libncurses-dev xz-utils libssl-dev bc flex libelf-dev bison
```



```
Aug 28 23:21
joginaidu@joginaidu-virtual-machine: ~/os_A1/linux-5.13.13

CC [M] sound/usb/6fire/snd-usb-6fire.mod.o
LD [M] sound/usb/6fire/snd-usb-6fire.ko
CC [M] sound/usb/bcd2000/snd-bcd2000.mod.o
LD [M] sound/usb/bcd2000/snd-bcd2000.ko
CC [M] sound/usb/caiaq/snd-usb-caiaq.mod.o
LD [M] sound/usb/caiaq/snd-usb-caiaq.ko
CC [M] sound/usb/hiface/snd-usb-hiface.mod.o
LD [M] sound/usb/hiface/snd-usb-hiface.ko
CC [M] sound/usb/line6/snd-usb-line6.mod.o
LD [M] sound/usb/line6/snd-usb-line6.ko
CC [M] sound/usb/line6/snd-usb-pod.mod.o
LD [M] sound/usb/line6/snd-usb-pod.ko
CC [M] sound/usb/line6/snd-usb-podhd.mod.o
LD [M] sound/usb/line6/snd-usb-podhd.ko
CC [M] sound/usb/line6/snd-usb-toneport.mod.o
LD [M] sound/usb/line6/snd-usb-toneport.ko
CC [M] sound/usb/line6/snd-usb-variax.mod.o
LD [M] sound/usb/line6/snd-usb-variax.ko
CC [M] sound/usb/misc/snd-ua101.mod.o
LD [M] sound/usb/misc/snd-ua101.ko
CC [M] sound/usb/snd-usb-audio.mod.o
LD [M] sound/usb/snd-usb-audio.ko
CC [M] sound/usb/snd-usbmidi-lib.mod.o
LD [M] sound/usb/snd-usbmidi-lib.ko
CC [M] sound/usb/usx2y/snd-usb-us122l.mod.o
LD [M] sound/usb/usx2y/snd-usb-us122l.ko
CC [M] sound/usb/usx2y/snd-usb-usx2y.mod.o
LD [M] sound/usb/usx2y/snd-usb-usx2y.ko
CC [M] sound/x86/snd-hdmi-lpe-audio.mod.o
LD [M] sound/x86/snd-hdmi-lpe-audio.ko
CC [M] sound/xen/snd_xen_front.mod.o
LD [M] sound/xen/snd_xen_front.ko

joginaidu@joginaidu-virtual-machine: ~/os_A1/linux-5.13.13$
```

b. After completing the compilation process, we can install the modules we have enabled with the following command

sudo make modules_install

```
Aug 28 23:28
joginaidu@joginaidu-virtual-machine: ~/os_A1/linux-5.13.13

AR arch/x86/kernel/built-in.a
AR arch/x86/built-in.a
CC [M] arch/x86/kvm/../../../virt/kvm/dirty_ring.o
LD [M] arch/x86/kvm/../../../virt/kvm/dirty_ring.ko
CC [M] arch/x86/kvm/../../../virt/kvm/async_pf.o
LD [M] arch/x86/kvm/../../../virt/kvm/async_pf.ko
CC [M] arch/x86/kvm/emulate.o
LD [M] arch/x86/kvm/emulate.ko
CC [M] arch/x86/kvm/i8259.o
LD [M] arch/x86/kvm/i8259.ko
CC [M] arch/x86/kvm/i8254.o
LD [M] arch/x86/kvm/i8254.ko
CC [M] arch/x86/kvm/iopic.o
LD [M] arch/x86/kvm/iopic.ko
CC [M] arch/x86/kvm/i8042.o
LD [M] arch/x86/kvm/i8042.ko
CC [M] arch/x86/kvm/cpuid.o
LD [M] arch/x86/kvm/cpuid.ko
CC [M] arch/x86/kvm/pmu.o
LD [M] arch/x86/kvm/pmu.ko
CC [M] arch/x86/kvm/mtrr.o
LD [M] arch/x86/kvm/mtrr.ko
CC [M] arch/x86/kvm/hyperv.o
LD [M] arch/x86/kvm/hyperv.ko
CC [M] arch/x86/kvm/debugfs.o
LD [M] arch/x86/kvm/debugfs.ko
CC [M] arch/x86/kvm/mmu/mmu.o
LD [M] arch/x86/kvm/mmu/mmu.ko
CC [M] arch/x86/kvm/mmu/paging_track.o
LD [M] arch/x86/kvm/mmu/paging_track.ko
CC [M] arch/x86/kvm/mmu/spte.o
LD [M] arch/x86/kvm/mmu/spte.ko
CC [M] arch/x86/kvm/mmu/tdp_iter.o
LD [M] arch/x86/kvm/mmu/tdp_iter.ko
CC [M] arch/x86/kvm/mmu/tdp_mmu.o
LD [M] arch/x86/kvm/mmu/tdp_mmu.ko
CC [M] arch/x86/kvm/vmx/vmx.o
LD [M] arch/x86/kvm/vmx/vmx.ko
AS [M] arch/x86/kvm/vmx/vmexec.o
LD [M] arch/x86/kvm/vmx/vmexec.ko
CC [M] arch/x86/kvm/vmx/vmx_intel.o
LD [M] arch/x86/kvm/vmx/vmx_intel.ko
CC [M] arch/x86/kvm/vmx/vmx_vms12.o
LD [M] arch/x86/kvm/vmx/vmx_vms12.ko
CC [M] arch/x86/kvm/vmx/vmx_vms.o
LD [M] arch/x86/kvm/vmx/vmx_vms.ko
CC [M] arch/x86/kvm/vmx/nested.o
LD [M] arch/x86/kvm/vmx/nested.ko
CC [M] arch/x86/kvm/vmx/posted_intr.o
LD [M] arch/x86/kvm/vmx/posted_intr.ko
CC [M] arch/x86/kvm/svm/svm.o
LD [M] arch/x86/kvm/svm/svm.ko
AS [M] arch/x86/kvm/svm/vmexec.o
LD [M] arch/x86/kvm/svm/vmexec.ko
CC [M] arch/x86/kvm/svm/nested.o
LD [M] arch/x86/kvm/svm/nested.ko
CC [M] arch/x86/kvm/svm/avic.o
LD [M] arch/x86/kvm/svm/avic.ko
CC [M] arch/x86/kvm/svm/sev.o
LD [M] arch/x86/kvm/svm/sev.ko
LD [M] arch/x86/kvm/kvm.o
LD [M] arch/x86/kvm/kvm-intel.o
LD [M] arch/x86/kvm/kvm-amd.o
CC [M] kernel/kheaders.o

joginaidu@joginaidu-virtual-machine: ~/os_A1/linux-5.13.13$ sudo make modules_install
```

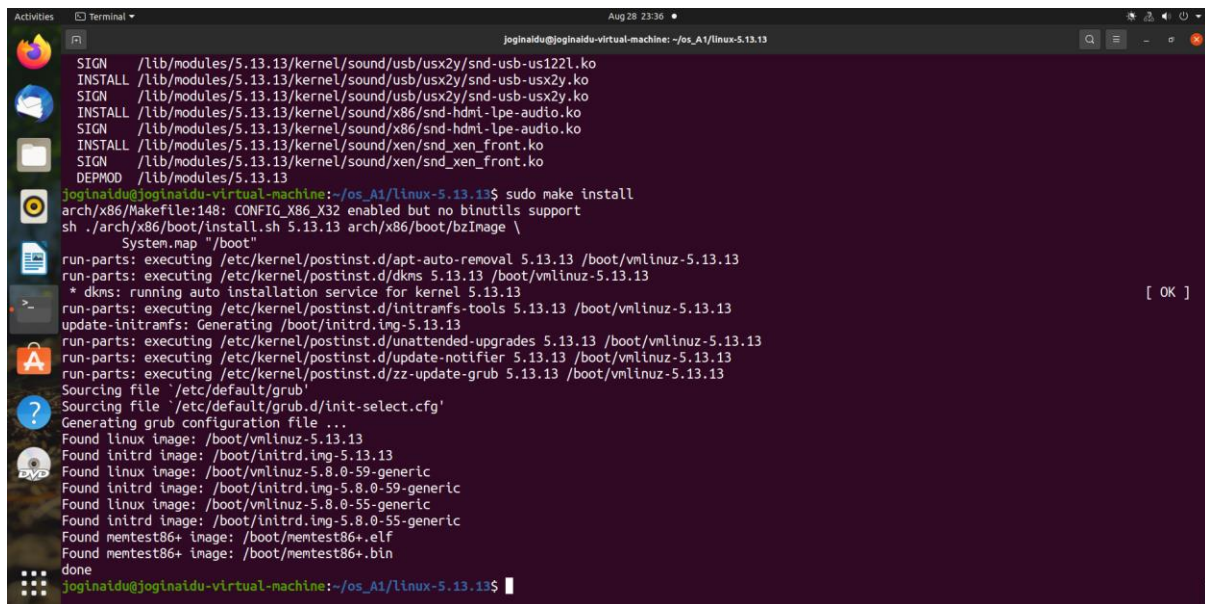
```
Aug 28 23:28
joginaidu@joginaidu-virtual-machine: ~/os_A1/linux-5.13.13

SIGN /lib/modules/5.13.13/kernel/sound/usb/6fire/snd-usb-6fire.ko
INSTALL /lib/modules/5.13.13/kernel/sound/usb/6fire/snd-usb-6fire.ko
SIGN /lib/modules/5.13.13/kernel/sound/usb/bcd2000/snd-bcd2000.ko
INSTALL /lib/modules/5.13.13/kernel/sound/usb/bcd2000/snd-bcd2000.ko
SIGN /lib/modules/5.13.13/kernel/sound/usb/caiaq/snd-usb-caiaq.ko
INSTALL /lib/modules/5.13.13/kernel/sound/usb/caiaq/snd-usb-caiaq.ko
SIGN /lib/modules/5.13.13/kernel/sound/usb/hiface/snd-usb-hiface.ko
INSTALL /lib/modules/5.13.13/kernel/sound/usb/hiface/snd-usb-hiface.ko
SIGN /lib/modules/5.13.13/kernel/sound/usb/line6/snd-usb-line6.ko
INSTALL /lib/modules/5.13.13/kernel/sound/usb/line6/snd-usb-line6.ko
SIGN /lib/modules/5.13.13/kernel/sound/usb/line6/snd-usb-pod.ko
INSTALL /lib/modules/5.13.13/kernel/sound/usb/line6/snd-usb-pod.ko
SIGN /lib/modules/5.13.13/kernel/sound/usb/line6/snd-usb-podhd.ko
INSTALL /lib/modules/5.13.13/kernel/sound/usb/line6/snd-usb-podhd.ko
SIGN /lib/modules/5.13.13/kernel/sound/usb/line6/snd-usb-toneport.ko
INSTALL /lib/modules/5.13.13/kernel/sound/usb/line6/snd-usb-toneport.ko
SIGN /lib/modules/5.13.13/kernel/sound/usb/line6/snd-usb-variax.ko
INSTALL /lib/modules/5.13.13/kernel/sound/usb/line6/snd-usb-variax.ko
SIGN /lib/modules/5.13.13/kernel/sound/usb/misc/snd-ua101.ko
INSTALL /lib/modules/5.13.13/kernel/sound/usb/misc/snd-ua101.ko
SIGN /lib/modules/5.13.13/kernel/sound/usb/snd-usb-audio.ko
INSTALL /lib/modules/5.13.13/kernel/sound/usb/snd-usb-audio.ko
SIGN /lib/modules/5.13.13/kernel/sound/usb/snd-usbmidi-lib.ko
INSTALL /lib/modules/5.13.13/kernel/sound/usb/snd-usbmidi-lib.ko
SIGN /lib/modules/5.13.13/kernel/sound/usb/usx2y/snd-usb-us122l.ko
INSTALL /lib/modules/5.13.13/kernel/sound/usb/usx2y/snd-usb-us122l.ko
SIGN /lib/modules/5.13.13/kernel/sound/usb/usx2y/snd-usb-usx2y.ko
INSTALL /lib/modules/5.13.13/kernel/sound/usb/usx2y/snd-usb-usx2y.ko
SIGN /lib/modules/5.13.13/kernel/sound/x86/snd-hdmi-lpe-audio.ko
INSTALL /lib/modules/5.13.13/kernel/sound/x86/snd-hdmi-lpe-audio.ko
SIGN /lib/modules/5.13.13/kernel/sound/xen/snd_xen_front.ko
INSTALL /lib/modules/5.13.13/kernel/sound/xen/snd_xen_front.ko
DEPMOD /lib/modules/5.13.13

joginaidu@joginaidu-virtual-machine: ~/os_A1/linux-5.13.13$
```

c. Now we can install the kernel with the following command

`sudo make install`



```
SIGN /lib/modules/5.13.13/kernel/sound/usb/usx2y/snd-usb-us122l.ko
INSTALL /lib/modules/5.13.13/kernel/sound/usb/usx2y/snd-usb-usx2y.ko
SIGN /lib/modules/5.13.13/kernel/sound/usb/usx2y/snd-usb-usx2y.ko
INSTALL /lib/modules/5.13.13/kernel/sound/x86/snd-hdmi-lpe-audio.ko
SIGN /lib/modules/5.13.13/kernel/sound/x86/snd-hdmi-lpe-audio.ko
INSTALL /lib/modules/5.13.13/kernel/sound/xen/snd_xen_front.ko
SIGN /lib/modules/5.13.13/kernel/sound/xen/snd_xen_front.ko
DEPMOD /lib/modules/5.13.13
joginaidu@joginaidu-virtual-machine:~/os_A1/linux-5.13.13$ sudo make install
arch/x86/Makefile:148: CONFIG_X86_X32 enabled but no binutils support
sh ./arch/x86/boot/install.sh 5.13.13 arch/x86/boot/bzImage \
System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 5.13.13 /boot/vmlinuz-5.13.13
run-parts: executing /etc/kernel/postinst.d/dkms 5.13.13 /boot/vmlinuz-5.13.13
* dkms: running auto installation service for kernel 5.13.13
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 5.13.13 /boot/vmlinuz-5.13.13
update-initramfs: Generating /boot/initrd.img-5.13.13
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 5.13.13 /boot/vmlinuz-5.13.13
run-parts: executing /etc/kernel/postinst.d/update-notifier 5.13.13 /boot/vmlinuz-5.13.13
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 5.13.13 /boot/vmlinuz-5.13.13
Sourcing file '/etc/default/grub'
Sourcing file '/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.13.13
Found initrd image: /boot/initrd.img-5.13.13
Found linux image: /boot/vmlinuz-5.8.0-59-generic
Found initrd image: /boot/initrd.img-5.8.0-59-generic
Found linux image: /boot/vmlinuz-5.8.0-55-generic
Found initrd image: /boot/initrd.img-5.8.0-55-generic
Found mentest86+ image: /boot/mentest86+.elf
Found mentest86+ image: /boot/mentest86+.bin
done
joginaidu@joginaidu-virtual-machine:~/os_A1/linux-5.13.13$
```

The above process in itself updates the initramfs, which is responsible to look for kernels in the /boot/folder and add them to the grub's configuration file.

We need to update the grub before rebooting and for that we need to run the following command

`sudo update-grub`

Now reboot the virtual machine and hold the right shift key button to access the **GRUB bootloader**.

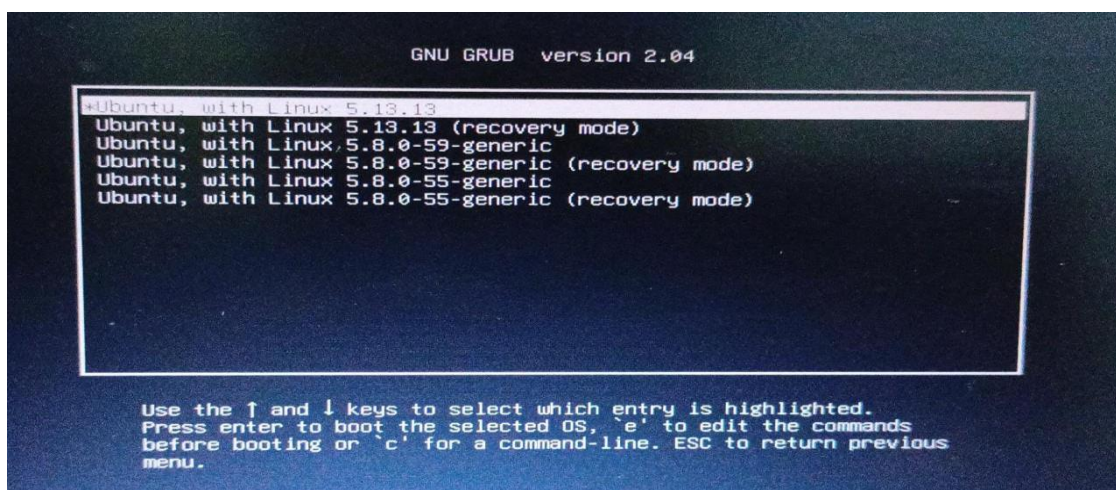
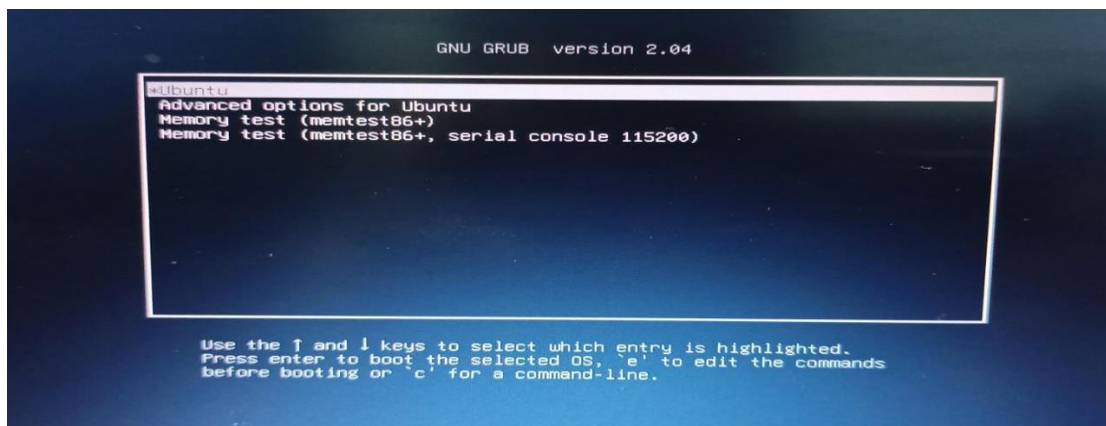
To reboot enter the following command:

`reboot`


```
Aug 28 23:39
joginaidu@joginaidu-virtual-machine: ~/os_A1/linux-5.13.13
run-parts: executing /etc/kernel/postinst.d/dkms 5.13.13 /boot/vmlinuz-5.13.13
* dkms: running auto installation service for kernel 5.13.13
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 5.13.13 /boot/vmlinuz-5.13.13
update-initramfs: Generating /boot/initrd.img-5.13.13
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 5.13.13 /boot/vmlinuz-5.13.13
run-parts: executing /etc/kernel/postinst.d/update-notifier 5.13.13 /boot/vmlinuz-5.13.13
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 5.13.13 /boot/vmlinuz-5.13.13
Sourcing file '/etc/default/grub'
Sourcing file '/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.13.13
Found initrd image: /boot/initrd.img-5.13.13
Found linux image: /boot/vmlinuz-5.8.0-59-generic
Found initrd image: /boot/initrd.img-5.8.0-59-generic
Found linux image: /boot/vmlinuz-5.8.0-55-generic
Found initrd image: /boot/initrd.img-5.8.0-55-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
joginaidu@joginaidu-virtual-machine:~/os_A1/linux-5.13.13$ sudo update-grub
Sourcing file '/etc/default/grub'
Sourcing file '/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.13.13
Found initrd image: /boot/initrd.img-5.13.13
Found linux image: /boot/vmlinuz-5.8.0-59-generic
Found initrd image: /boot/initrd.img-5.8.0-59-generic
Found linux image: /boot/vmlinuz-5.8.0-55-generic
Found initrd image: /boot/initrd.img-5.8.0-55-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
joginaidu@joginaidu-virtual-machine:~/os_A1/linux-5.13.13$ reboot
```

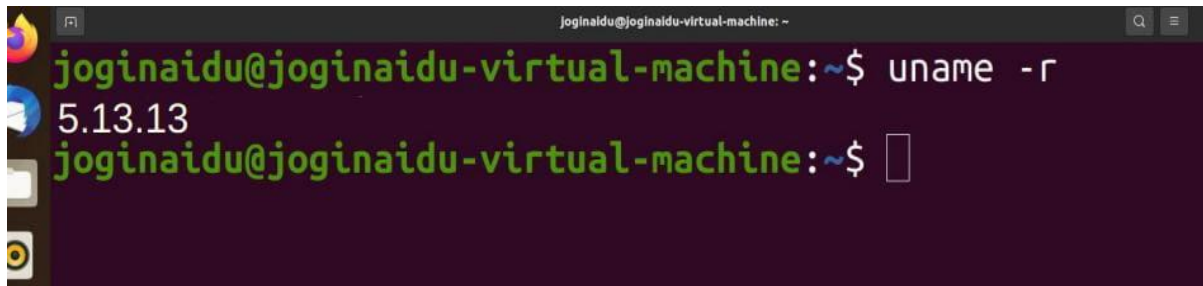
Step 6: GRUB MENU

While accessing the GRUB MENU click on Advanced options to see different versions installed and after that choose the latest kernel on top of it (i.e., *Ubuntu, with Linux-5.13.13 generic)



To show that latest kernel is installed, run the following command:

```
uname -r
```

A screenshot of a terminal window with a dark purple background. The window title is 'joginaidu@joginaidu-virtual-machine: ~'. The prompt is 'joginaidu@joginaidu-virtual-machine:~\$'. The command 'uname -r' has been entered and executed, resulting in the output '5.13.13'. The prompt is now 'joginaidu@joginaidu-virtual-machine:~\$' followed by a cursor.

```
joginaidu@joginaidu-virtual-machine:~$ uname -r
5.13.13
joginaidu@joginaidu-virtual-machine:~$
```

5.Summary:

We have successfully compiled the latest stable kernel (**Linux-5.13.13**) by following the above steps, resolving the errors and installing the requirements and we have dual booted it with the current version.

6.References:

<https://www.guru99.com/operating-system-tutorial.html>

<https://www.geeksforgeeks.org/operating-systems/?ref=ghm>

<https://www.cyberciti.biz/tips/compiling-linux-kernel-26.html>

<https://www.freecodecamp.org/news/building-and-installing-the-latest-linux-kernel-from-source-6d8df5345980/>