

# CS3003D: Operating Systems

## Assignment 1

October 12, 2020

### 1. 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.

### 2. Methodology:

1. Dual booting can be directly done with the host OS but if some thing goes wrong then OS could be corrupted. Hence it's recommended to use a Virtual Machine. Install VMWare and load pop-OS into it.(Note: pop-OS is linux based on Ubuntu)
2. Download the kernel source code from kernel.org and extract the corresponding tar file.
3. Get all the required dependencies using apt
4. Compile the kernel
5. Install the compiled kernel and add it to grub entry.
6. Reboot the system

### 3. Process and Explanation:

Install VMware and get pop-OS stable version from the pop-OS official website. I'm using Pop-OS 20.04 LTS for this purpose. Give at least 80 to 100 GB memory space since the kernel compilation itself will be 21 GB. After installing ubuntu. Open terminal to get all the required dependencies.

```
$wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.8.13.tar.xz
```

This will download the kernel code published on <https://cdn.kernel.org> as a tar file. This file is of 107 mb around.

```
$tar -xf linux-5.8.13.tar.xz
```

This will extract the downloaded tar file into the folder linux-5.8.13 folder which contain all the source code. The kernel compilation options (like which modules should be included, which all drivers to be added) are managed through '.config' file located in the same folder. We are provided with 'make menuconfig' option to configure them with a TUI, I needed to install the following dependencies. (To get those work)

```
$sudo apt install build-essential libncurses-dev flex bison libssl-dev libelf-dev
```

The above command will install all the required dependencies for the kernel compilation. Now get the kernel code using wget command.

```
$cd linux-5.8.13/  
$cp /boot/config-4.15.0-0-generic .config  
sudo menuconfig
```

Just to make sure you have all the required files in this directory and you are good to compile the code check the menuconfig. I'm not making any changes so the configuration will be default configuration. Now save it as it is and run the following command to compile the kernel code.

```
$make -j5
```

This command will start compiling the kernel code. Here 5 describes no. of threads or parallel GCC instances taken to compile . If your system is good you can give 6 too. This compilation took two and half hours for me. Again it depends on your system. So maintain patience while executing and make sure you turn off screenoff and sleep mode. That can sometimes stop the process and os hangs. (Make sure for both host OS and vm)

After the compilation is over check the directory size which should be around 21 GB. Now run make install command which will install kernel modules to the /, and kernel image to /boot and start grub update to add new entry to boot.

```
$sudo make modules_install  
$sudo make install
```

sudo make modules\_install is not required if you are using Ubuntu but if you are using other linux you must include this command for proper execution.

The newer Pop-OS installations by default hide the GRUB boot menu by setting the 'GRUB TIMEOUT=0', we need to update this, so that the menu remains visible. The GRUB config file is located in '/etc/default/grub', we can update the 'GRUB TIMEOUT' there, and run grub update so that new configuration is loaded. Update 'GRUB TIMEOUT' to 5 or 10. Then comment the command GRUB MENU VISIBILITY so that Grub Menu Remains Visible.

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

If you get an error while executing vim command it means that you haven't installed it in your OS. For this use the command "sudo apt install vim" to install the latest vim application into your OS. Now we are good to go and can reboot the system.

```
$sudo reboot
```

You can also directly restart or type the above command in the terminal. Now once the system starts it will open GRUB menu. Open advanced settings and select our newly booted kernel i.e. version 5.8.14

```
$uname -a
```

This above command will give the current version of the kernel which should be 5.8.14

## 4. Screenshots

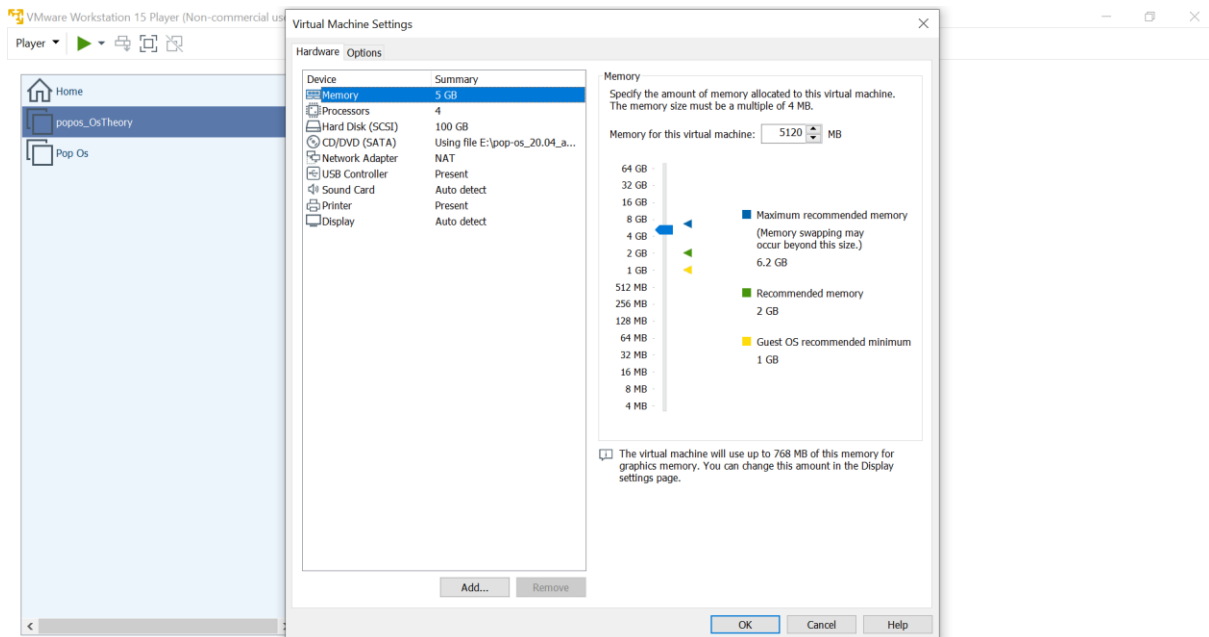


Figure 1.1 Specifications of Pop-OS

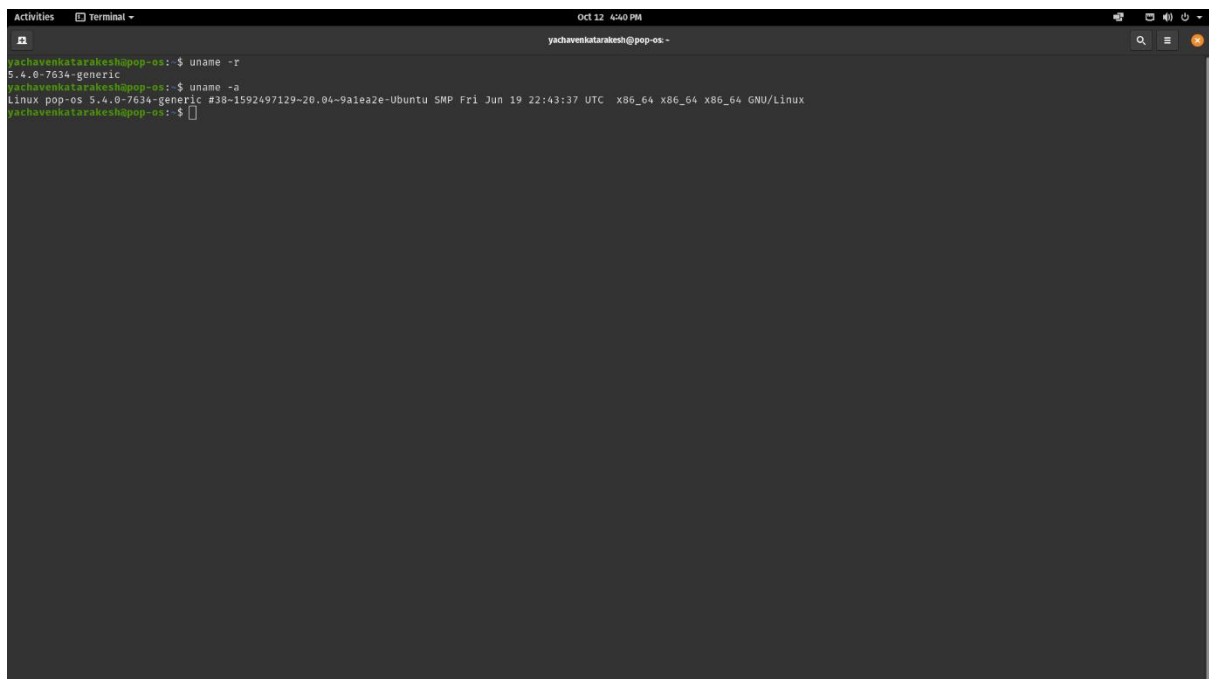


Figure 1.2 Initial kernel version

```
yachavenkatarakesh@pop-os: ~/linux-5.8.14
LD [M] drivers/i2c/muxes/i2c-mux-reg.ko
LD [M] drivers/i3c/i3c.ko
LD [M] drivers/i3c/master/dw-i3c-master.ko
LD [M] drivers/i3c/master/i3c-master-cdns.ko
LD [M] drivers/iio/accel/adis16201.ko
LD [M] drivers/iio/accel/adis16209.ko
LD [M] drivers/iio/accel/adxl372.ko
LD [M] drivers/iio/accel/adxl372_i2c.ko
LD [M] drivers/iio/accel/adxl372_spi.ko
LD [M] drivers/iio/accel/bma220_spi.ko
LD [M] drivers/iio/accel/bmc150-accel-core.ko
LD [M] drivers/iio/accel/bmc150-accel-i2c.ko
LD [M] drivers/iio/accel/bmc150-accel-spi.ko
LD [M] drivers/iio/accel/cros_ec-accel_legacy.ko
LD [M] drivers/iio/accel/da280.ko
LD [M] drivers/iio/accel/da311.ko
LD [M] drivers/iio/accel/dmard09.ko
LD [M] drivers/iio/accel/dmard10.ko
LD [M] drivers/iio/accel/hid-sensor-accel-3d.ko
LD [M] drivers/iio/accel/kxcjk-1013.ko
LD [M] drivers/iio/accel/kxsd9-i2c.ko
LD [M] drivers/iio/accel/kxsd9-spi.ko
LD [M] drivers/iio/accel/kxsd9.ko
LD [M] drivers/iio/accel/mc3230.ko
LD [M] drivers/iio/accel/mma7455_core.ko
LD [M] drivers/iio/accel/mma7455_i2c.ko
LD [M] drivers/iio/accel/mma7455_spi.ko
LD [M] drivers/iio/accel/mma7660.ko
LD [M] drivers/iio/accel/mma8452.ko
LD [M] drivers/iio/accel/mma951.ko
LD [M] drivers/iio/accel/mma951_core.ko
LD [M] drivers/iio/accel/mma953.ko
LD [M] drivers/iio/accel/mxc4005.ko
LD [M] drivers/iio/accel/mxc6255.ko
LD [M] drivers/iio/accel/sca3000.ko
LD [M] drivers/iio/accel/ssp_accel_sensor.ko
LD [M] drivers/iio/accel/st_accel.ko
LD [M] drivers/iio/accel/st_accel_i2c.ko
LD [M] drivers/iio/accel/st_accel_spi.ko
LD [M] drivers/iio/accel/stk8312.ko
LD [M] drivers/iio/accel/stk8ba50.ko
LD [M] drivers/iio/adc/ad7124.ko
LD [M] drivers/iio/adc/ad7192.ko
LD [M] drivers/iio/adc/ad7266.ko
LD [M] drivers/iio/adc/ad7291.ko
LD [M] drivers/iio/adc/ad7298.ko
LD [M] drivers/iio/adc/ad7476.ko
LD [M] drivers/iio/adc/ad7606.ko
LD [M] drivers/iio/adc/ad7606_nar.ko
```

fig 1.3 compiling the kernel code

```
yachavenkatarakesh@pop-os: ~/linux-5.8.14$ sudo make modules_install
[sudo] password for yachavenkatarakesh:
INSTALL arch/x86/crypto/aegis128-aesni.ko
INSTALL arch/x86/crypto/aesni-intel.ko
INSTALL arch/x86/crypto/blowfish-x86_64.ko
INSTALL arch/x86/crypto/camellia-aesni-avx-x86_64.ko
INSTALL arch/x86/crypto/camellia-aesni-avx2.ko
INSTALL arch/x86/crypto/camellia-x86_64.ko
INSTALL arch/x86/crypto/cast5-avx-x86_64.ko
INSTALL arch/x86/crypto/caste-avx-x86_64.ko
INSTALL arch/x86/crypto/chacha-x86_64.ko
INSTALL arch/x86/crypto/crc32-pclmul.ko
INSTALL arch/x86/crypto/crc10dif-pclmul.ko
INSTALL arch/x86/crypto/des3_ede-x86_64.ko
INSTALL arch/x86/crypto/ghash-clmulni-intel.ko
INSTALL arch/x86/crypto/glue_helper.ko
INSTALL arch/x86/crypto/nhpoly1305-avx2.ko
INSTALL arch/x86/crypto/nhpoly1305-sse2.ko
INSTALL arch/x86/crypto/poly1305-x86_64.ko
INSTALL arch/x86/crypto/serpent-avx-x86_64.ko
INSTALL arch/x86/crypto/serpent-avx2.ko
INSTALL arch/x86/crypto/serpent-sse2-x86_64.ko
INSTALL arch/x86/crypto/sha1-ssse3.ko
INSTALL arch/x86/crypto/sha256-ssse3.ko
INSTALL arch/x86/crypto/sha512-ssse3.ko
INSTALL arch/x86/crypto/twofish-avx-x86_64.ko
INSTALL arch/x86/crypto/twofish-x86_64-3way.ko
INSTALL arch/x86/crypto/twofish-x86_64.ko
INSTALL arch/x86/events/intel/intel-cstate.ko
INSTALL arch/x86/events/rapl.ko
INSTALL arch/x86/kernel/cpu/mce/mce-inject.ko
INSTALL arch/x86/kernel/cpuid.ko
INSTALL arch/x86/kernel/msr.ko
INSTALL arch/x86/kvm/kvm-amd.ko
INSTALL arch/x86/kvm/kvm-intel.ko
INSTALL arch/x86/kvm/kvm.ko
INSTALL arch/x86/oprofile/oprofile.ko
INSTALL arch/x86/platform/atom/punit_atom_debug.ko
INSTALL block/bfq.ko
INSTALL block/kyber-iosched.ko
INSTALL crypto/842.ko
INSTALL crypto/adiantum.ko
INSTALL crypto/aegis128.ko
INSTALL crypto/aes_ti.ko
INSTALL crypto/af_alg.ko
INSTALL crypto/algif_aead.ko
INSTALL crypto/algif_hash.ko
INSTALL crypto/algif_rng.ko
INSTALL crypto/algif_skcipher.ko
```

fig 1.4 Executing make modules\_install command

```
yachavenkatarakesh@pop-os: ~/linux-5.8.14
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-max98090_ti.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-nau8824.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-rt5645.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-rt5672.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-glk-rt5682_max98357a.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-haswell.ko
INSTALL sound/soc/intel/common/snd-soc-acpi-intel-match.ko
INSTALL sound/soc/intel/common/snd-soc-sst-acpi.ko
INSTALL sound/soc/intel/common/snd-soc-sst-dsp.ko
INSTALL sound/soc/intel/common/snd-soc-sst-firmware.ko
INSTALL sound/soc/intel/common/snd-soc-sst-ipc.ko
INSTALL sound/soc/intel/haswell/snd-soc-sst-haswell-pcm.ko
INSTALL sound/soc/snd-soc-acpi.ko
INSTALL sound/soc/snd-soc-core.ko
INSTALL sound/soc/sof/intel/snd-sof-intel-byt.ko
INSTALL sound/soc/sof/intel/snd-sof-intel-hda-common.ko
INSTALL sound/soc/sof/intel/snd-sof-intel-hda.ko
INSTALL sound/soc/sof/intel/snd-sof-intel-ipc.ko
INSTALL sound/soc/sof/snd-sof-acpi.ko
INSTALL sound/soc/sof/snd-sof-pci.ko
INSTALL sound/soc/sof/snd-sof.ko
INSTALL sound/soc/sof/xtensa/snd-sof-xtensa-dsp.ko
INSTALL sound/soc/xilinx/snd-soc-xlnx-formatter-pcm.ko
INSTALL sound/soc/xilinx/snd-soc-xlnx-i2s.ko
INSTALL sound/soc/xilinx/snd-soc-xlnx-spdif.ko
INSTALL sound/soc/xtensa/snd-soc-xtfpga-i2s.ko
INSTALL sound/soc/zte/zx-tdm.ko
INSTALL sound/soundcore.ko
INSTALL sound/synth/emux/snd-emux-synth.ko
INSTALL sound/synth/snd-util-mem.ko
INSTALL sound/usb/6fire/snd-usb-6fire.ko
INSTALL sound/usb/bcd2000/snd-bcd2000.ko
INSTALL sound/usb/caiaq/snd-usb-caiaq.ko
INSTALL sound/usb/hiface/snd-usb-hiface.ko
INSTALL sound/usb/line6/snd-usb-line6.ko
INSTALL sound/usb/line6/snd-usb-pod.ko
INSTALL sound/usb/line6/snd-usb-podhd.ko
INSTALL sound/usb/line6/snd-usb-toneport.ko
INSTALL sound/usb/line6/snd-usb-variax.ko
INSTALL sound/usb/misc/snd-ua101.ko
INSTALL sound/usb/snd-usb-audio.ko
INSTALL sound/usb/snd-usbmidi-lib.ko
INSTALL sound/usb/usx2y/snd-usb-us122l.ko
INSTALL sound/usb/usx2y/snd-usb-usx2y.ko
INSTALL sound/x86/snd-hdmi-lpe-audio.ko
INSTALL sound/xen/snd_xen_front.ko
DEPMOD 5.8.14
yachavenkatarakesh@pop-os: ~/linux-5.8.14$
```

fig 1.5 Executed make module\_install successfully

```
yachavenkatarakesh@pop-os: ~/linux-5.8.14
INSTALL sound/x86/snd-hdmi-lpe-audio.ko
INSTALL sound/xen/snd_xen_front.ko
DEPMOD 5.8.14
yachavenkatarakesh@pop-os: ~/linux-5.8.14$ sudo make install
[sudo] password for yachavenkatarakesh:
sh ./arch/x86/boot/install.sh 5.8.14 arch/x86/boot/bzImage \
    System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 5.8.14 /boot/vmlinuz-5.8.14
run-parts: executing /etc/kernel/postinst.d/dkms 5.8.14 /boot/vmlinuz-5.8.14
* dkms: running auto installation service for kernel 5.8.14
Kernel preparation unnecessary for this kernel.  Skipping...

Building module:
cleaning build area...(bad exit status: 2)
unset ARCH; env NV_VERBOSE=1 'make' -j4 NV_EXCLUDE_BUILD_MODULES='' KERNEL_UNAME=5.8.14 IGNORE_XEN_PRESENCE=1 IGNORE_CC_MISMATCH=1 SYSSRC=/lib/modules/5.8.14/build LD
=/usr/bin/ld.bfd modules...(bad exit status: 2)
ERROR (dkms apport): kernel package linux-headers-5.8.14 is not supported
Error! Bad return status for module build on kernel: 5.8.14 (x86_64)
Consult /var/lib/dkms/nvidia/440.100/build/make.log for more information.

Kernel preparation unnecessary for this kernel.  Skipping...

Building module:
cleaning build area...
make -j4 KERNELRELEASE=5.8.14 -C /lib/modules/5.8.14/build M=/var/lib/dkms/system76-io/1.0.1-1559663713-20.04~ea5f61a/build....
cleaning build area...

DKMS: build completed.

system76-io.ko:
Running module version sanity check.
- Original module
- No original module exists within this kernel
- Installation
- Installing to /lib/modules/5.8.14/updates/dkms/

depmod.....

DKMS: install completed.

Kernel preparation unnecessary for this kernel.  Skipping...

Building module:
cleaning build area...
make -j4 KERNELRELEASE=5.8.14 -C /lib/modules/5.8.14/build M=/var/lib/dkms/system76/1.0.9-1592416603-20.04~706784b/build....
cleaning build area...

DKMS: build completed.
```

fig 1.6 Executing sudo make install command

```
yachavenkatarakesh@pop-os: ~/linux-5.8.14
cleaning build area...
DKMS: build completed.

system76.ko:
Running module version sanity check.
- Original module
- No original module exists within this kernel
- Installation
- Installing to /lib/modules/5.8.14/updates/dkms/

depmod...
DKMS: install completed.

Kernel preparation unnecessary for this kernel. Skipping...

Building module:
cleaning build area...
make -j4 KERNELRELEASE=5.8.14 -C /lib/modules/5.8.14/build M=/var/lib/dkms/system76_acpi/1.0.1-1571170639-20.04-d625910/build...
cleaning build area...
DKMS: build completed.

system76_acpi.ko:
Running module version sanity check.
- Original module
- No original module exists within this kernel
- Installation
- Installing to /lib/modules/5.8.14/updates/dkms/

depmod...
DKMS: install completed.

run-parts: executing /etc/kernel/postinst.d/initramfs-tools 5.8.14 /boot/vmlinuz-5.8.14
update-initramfs: Generating /boot/initrd.img-5.8.14
cryptsetup: WARNING: Resume target cryptswap uses a key file
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 5.8.14 /boot/vmlinuz-5.8.14
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.8.14
Found initrd image: /boot/initrd.img-5.8.14
Found linux image: /boot/vmlinuz-5.4.0-7634-generic
Found initrd image: /boot/initrd.img-5.4.0-7634-generic
done
yachavenkatarakesh@pop-os:~/linux-5.8.14$ sudo vim /etc/default/grub
```

fig 1.7 Executed sudo make install command successfully

```
yachavenkatarakesh@pop-os: ~/linux-5.8.14
yachavenkatarakesh@pop-os:~/linux-5.8.14$ sudo apt install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  vim-runtime
Suggested packages:
  ctags vim-doc vim-scripts
The following NEW packages will be installed:
  vim vim-runtime
0 upgraded, 2 newly installed, 0 to remove and 418 not upgraded.
Need to get 7,111 kB of archives.
After this operation, 34.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu focal/main amd64 vim-runtime all 2:8.1.2269-1ubuntu5 [5,873 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal/main amd64 vim amd64 2:8.1.2269-1ubuntu5 [1,238 kB]
Fetched 7,111 kB in 45s (158 kB/s)
Selecting previously unselected package vim-runtime.
(Reading database ... 208624 files and directories currently installed.)
Preparing to unpack .../vim-runtime_2%3a8.1.2269-1ubuntu5_all.deb ...
Adding 'diversion of /usr/share/vim/vim81/doc/help.txt to /usr/share/vim/vim81/doc/help.txt.vim-tiny by vim-runtime'
Adding 'diversion of /usr/share/vim/vim81/doc/tags to /usr/share/vim/vim81/doc/tags.vim-tiny by vim-runtime'
Unpacking vim-runtime (2:8.1.2269-1ubuntu5) ...
Selecting previously unselected package vim.
Preparing to unpack .../vim_2%3a8.1.2269-1ubuntu5_amd64.deb ...
Unpacking vim (2:8.1.2269-1ubuntu5) ...
Setting up vim-runtime (2:8.1.2269-1ubuntu5) ...
Setting up vim (2:8.1.2269-1ubuntu5) ...
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/vim (vim) in auto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/vimdiff (vimdiff) in auto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/rvim (rvim) in auto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/rview (rview) in auto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/vi (vi) in auto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/view (view) in auto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/ex (ex) in auto mode
Processing triggers for man-db (2.9.1-1) ...
yachavenkatarakesh@pop-os:~/linux-5.8.14$ sudo vim /etc/default/grub
yachavenkatarakesh@pop-os:~/linux-5.8.14$ sudo vim /etc/default/grub
yachavenkatarakesh@pop-os:~/linux-5.8.14$ 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-5.8.14
Found initrd image: /boot/initrd.img-5.8.14
Found linux image: /boot/vmlinuz-5.4.0-7634-generic
Found initrd image: /boot/initrd.img-5.4.0-7634-generic
done
yachavenkatarakesh@pop-os:~/linux-5.8.14$
```

fig 1.8 Installed vim and updated Grub

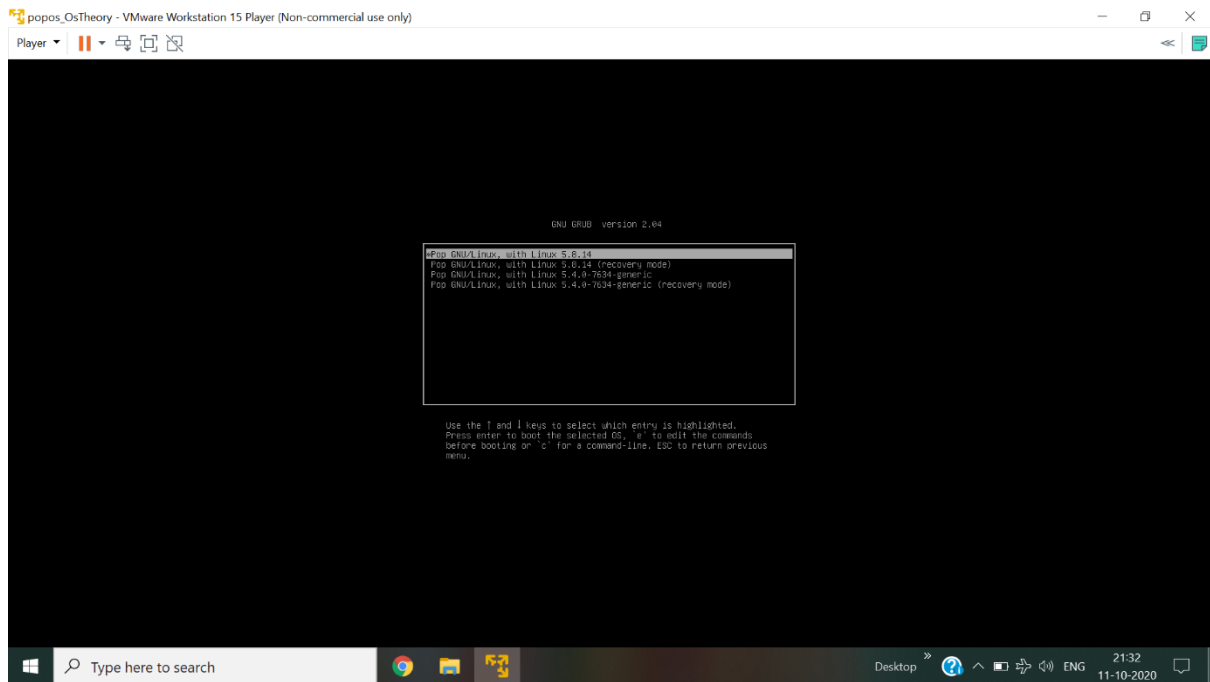


Fig 1.9 Grub Boot Menu with newly installed 5.8.14 kernel

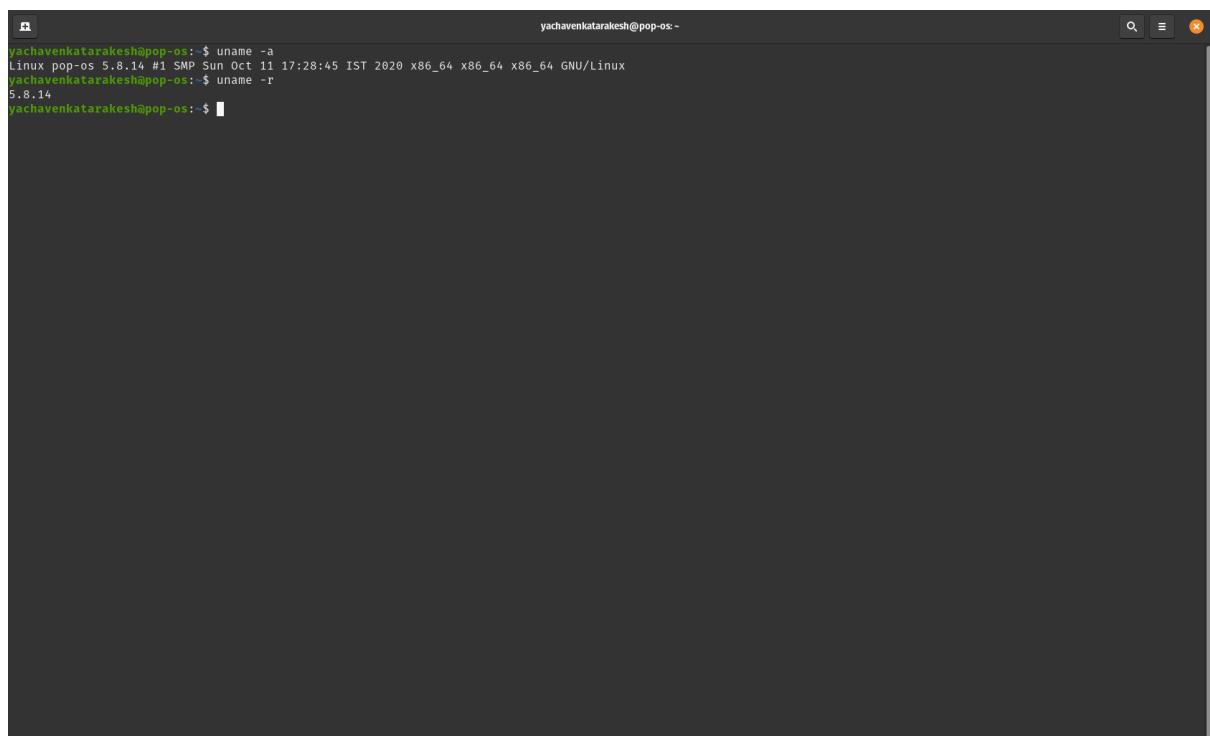


Fig 1.10 Final Kernel version

**References :**

[Medium Blogs](#)

[Kernel.org](#)

[Linux Documentation](#)

[Compiling Linux Kernel](#)