OS Assignment - 1

Assignment – 1

Name: Ponnuru Aadarsh Roll No: B210507CS

Batch: CS02 Date: 7th September 2023

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

Methodology

- Dual boot process can be done directly with the host OS. There will be several kernel versions anyway whenever we update the kernel.

  NOTE: any small mistake can cause malfunction of current OS. So we just have to accurately do the process. And maintain a backup of current kernel.

- Obtain the kernel's source code from kernel.org.
- Install the required dependencies.
- Compile the kernel.
- Install the compiled kernel.
- Add grub entry.
- Reboot the system.

Introduction

The objective of this report is to document the process of downloading, compiling and dual-booting the latest table kernel version from kernel.org alongside our current Linux kernel version.

This report is particularly for the fedora distribution users.

Procedure

Disclaimer

This Process might take a lot of time based on the System performance. Patience is the key.

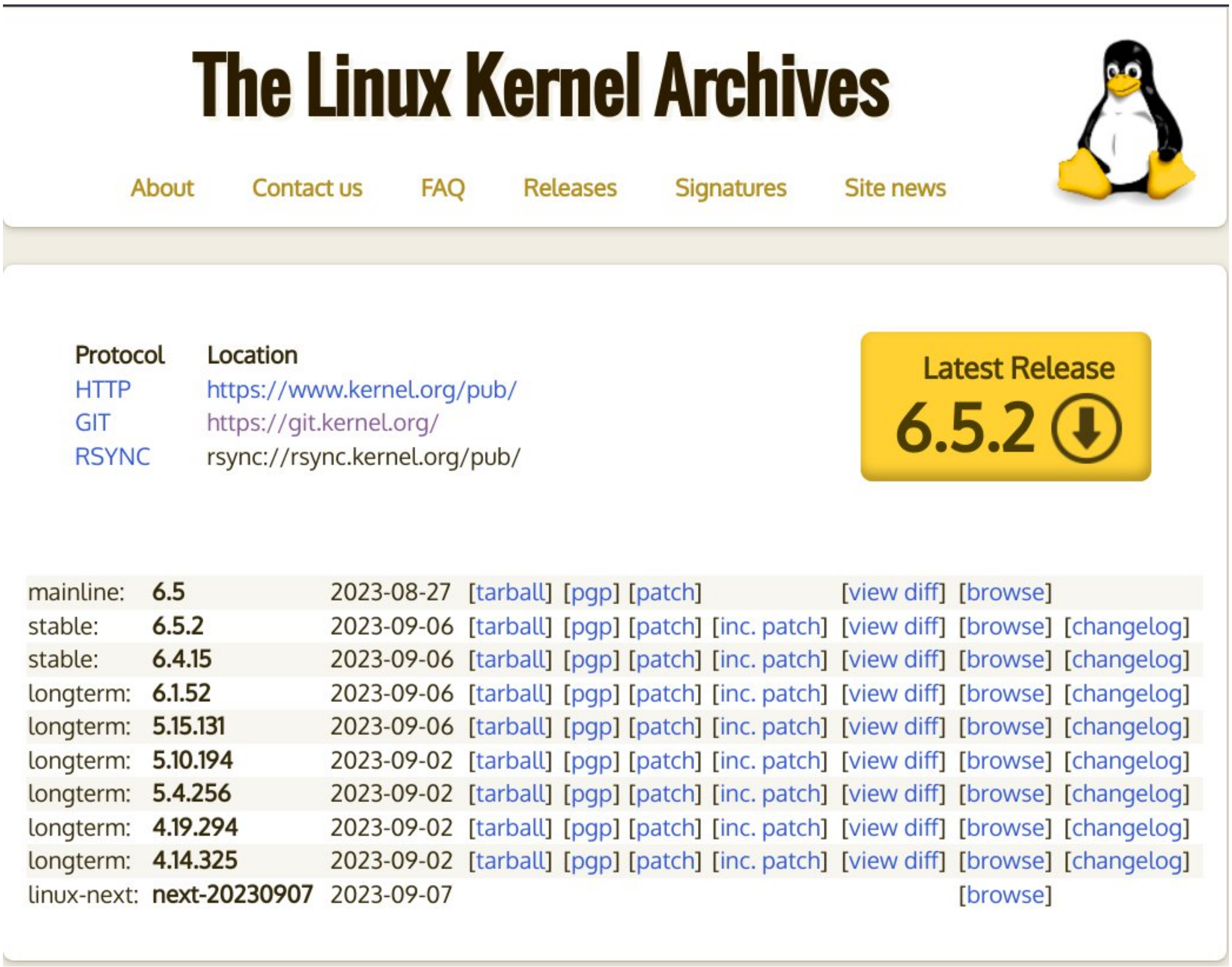
——-x——

These are the steps that we are going to do

- Download the latest kernel source code
- Extract the compressed tar file
- Configure the Linux kernel features and modules
- Install Required Compilers and other tools
- Compile the Linux kernel

Step-1 : Download the Latest Kernel source code

Visit the website from the link here and download the latest stable kernel version. It will displayed as shown in picture below



The Linux Kernel Archives

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

mainline:	6.5	2023-08-27	[tarball]	[pgp]	[patch]		[view diff]	[browse]	
stable:	6.5.2	2023-09-06	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
stable:	6.4.15	2023-09-06	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	6.1.52	2023-09-06	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.15.131	2023-09-06	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.10.194	2023-09-02	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.4.256	2023-09-02	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.19.294	2023-09-02	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.14.325	2023-09-02	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
linux-next:	next-20230907	2023-09-07						[browse]	

Step2:Extract the kernel’s tar file

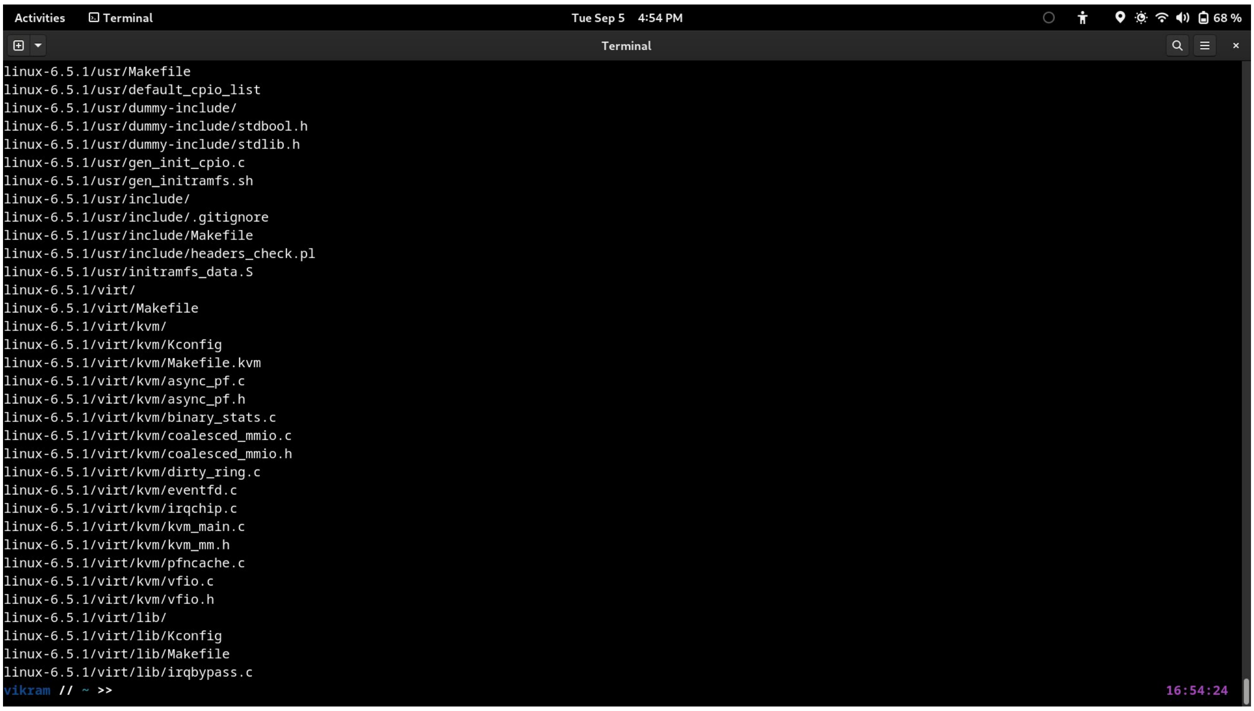
Download the latest release and extract the kernel tar file. The file looks like linux-x.y.z.tar.xz. Use the following command to extract the tar file

NOTE: At the time of install there was only 6.5.1 so the following steps will contain linux-6.5.1

Make sure you are in the downloaded file’s Directory

You can use the following command to do so

$ tar –xvf linux-6.5.1.tar.xz



```
linux-6.5.1/usr/Makefile
linux-6.5.1/usr/default_cpio_list
linux-6.5.1/usr/dummy-include/
linux-6.5.1/usr/dummy-include/stdbool.h
linux-6.5.1/usr/dummy-include/stdlib.h
linux-6.5.1/usr/gen_init_cpio.c
linux-6.5.1/usr/gen_initramfs.sh
linux-6.5.1/usr/include/
linux-6.5.1/usr/include/.gitignore
linux-6.5.1/usr/include/Makefile
linux-6.5.1/usr/include/headers_check.pl
linux-6.5.1/usr/initramfs_data.S
linux-6.5.1/virt/
linux-6.5.1/virt/Makefile
linux-6.5.1/virt/kvm/
linux-6.5.1/virt/kvm/Kconfig
linux-6.5.1/virt/kvm/Makefile.kvm
linux-6.5.1/virt/kvm/async_pf.c
linux-6.5.1/virt/kvm/async_pf.h
linux-6.5.1/virt/kvm/binary_stats.c
linux-6.5.1/virt/kvm/coalesced_mmio.c
linux-6.5.1/virt/kvm/coalesced_mmio.h
linux-6.5.1/virt/kvm/dirty_ring.c
linux-6.5.1/virt/kvm/eventfd.c
linux-6.5.1/virt/kvm/irqchip.c
linux-6.5.1/virt/kvm/kvm_main.c
linux-6.5.1/virt/kvm/kvm_mm.h
linux-6.5.1/virt/kvm/pfncache.c
linux-6.5.1/virt/kvm/vfio.c
linux-6.5.1/virt/kvm/vfio.h
linux-6.5.1/virt/lib/
linux-6.5.1/virt/lib/Kconfig
linux-6.5.1/virt/lib/Makefile
linux-6.5.1/virt/lib/irqbypass.c
vikram // ~ >>                                                          16:54:24
```

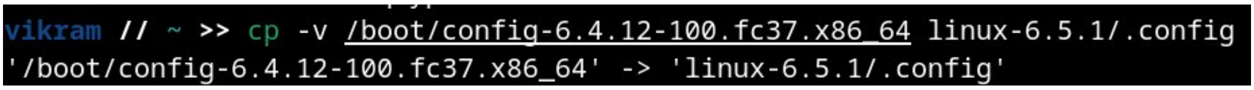
Step3:Configure the Linux kernel Features and Modules

You can check the curren version of your kernel using $uname -r

Now create the config file required by copying the configuration file of the current kernel to the .config file. You can use the following commands for this

$ cd linux-6.5.1

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



```
vikram // ~ >> cp -v /boot/config-6.4.12-100.fc37.x86_64 linux-6.5.1/.config
'/boot/config-6.4.12-100.fc37.x86_64' -> 'linux-6.5.1/.config'
```

Step4: Install Required Compilers and Other tools

$ sudo dnf install git gcc fakeroot make ncurses-devel bison flex openssl-devel dwarves make elfutils-libelf-devel

This installs the required tools for the installation

Step5:Compile the Linux Kernel

Start compiling to create a compressed kernel image, enter:

$ make

To speed up compile time, pass the -j as follows:

```
## use 4 core/thread ##

$ make -j 4

## get thread or cpu core count using nproc command ##

$ make -j $(nproc)
```

The $ nproc on my system gave 8 using which I performed the make command.

```
$ make -j8
```

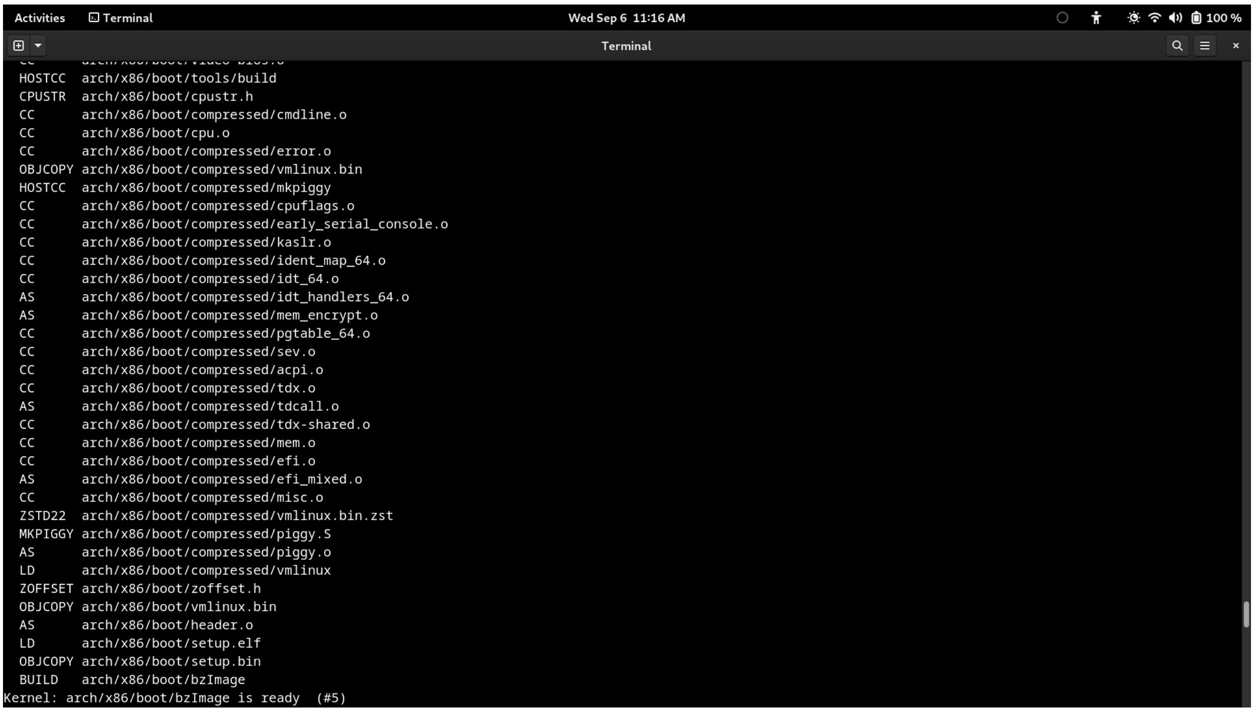
After this, it prompts several times for several modules to respond with y, n, m. but you can press enter for everything for default installation. You can do the same by running another version of above command

```
$ yes “” | make –j $(nproc)
```

Compiling and building the Linux kernel going take a significant amount of time. The build time depends upon your system’s resources such as available CPU core and the current system load. So we must have some patience.

The terminal lists all Linux kernel components: *memory management, hardware device drivers, filesystem drivers, network drivers, and process management.*

The end of this step looks like this

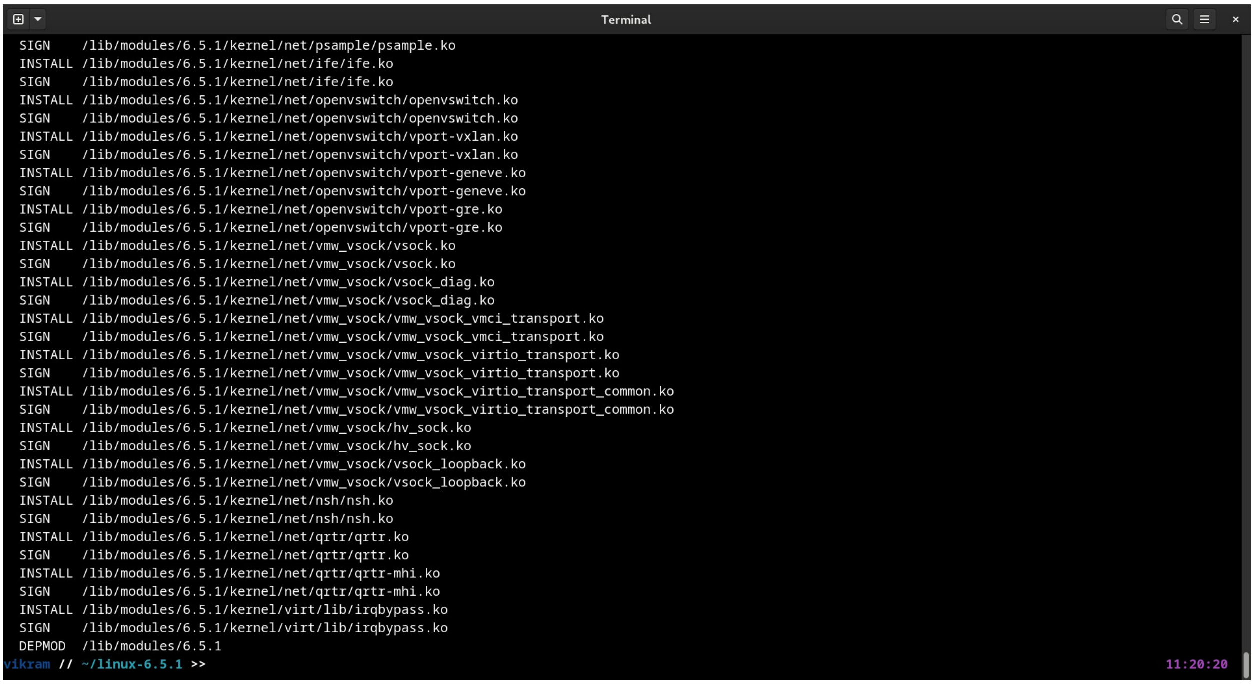


```
  HOSTCC  arch/x86/boot/tools/build
  CPUSTR  arch/x86/boot/cpustr.h
  CC      arch/x86/boot/compressed/cmdline.o
  CC      arch/x86/boot/cpu.o
  CC      arch/x86/boot/compressed/error.o
  OBJCOPY arch/x86/boot/compressed/vmlinux.bin
  HOSTCC  arch/x86/boot/compressed/mkpiggy
  CC      arch/x86/boot/compressed/cpuflags.o
  CC      arch/x86/boot/compressed/early_serial_console.o
  CC      arch/x86/boot/compressed/kaslr.o
  CC      arch/x86/boot/compressed/ident_map_64.o
  CC      arch/x86/boot/compressed/idt_64.o
  AS      arch/x86/boot/compressed/idt_handlers_64.o
  AS      arch/x86/boot/compressed/mem_encrypt.o
  CC      arch/x86/boot/compressed/pgtable_64.o
  CC      arch/x86/boot/compressed/sev.o
  CC      arch/x86/boot/compressed/acpi.o
  CC      arch/x86/boot/compressed/tdx.o
  AS      arch/x86/boot/compressed/tdcall.o
  CC      arch/x86/boot/compressed/tdx-shared.o
  CC      arch/x86/boot/compressed/mem.o
  CC      arch/x86/boot/compressed/efi.o
  AS      arch/x86/boot/compressed/efi_mixed.o
  CC      arch/x86/boot/compressed/misc.o
  ZSTD22  arch/x86/boot/compressed/vmlinux.bin.zst
  MKPIGGY arch/x86/boot/compressed/piggy.S
  AS      arch/x86/boot/compressed/piggy.o
  LD      arch/x86/boot/compressed/vmlinux
  ZOFFSET arch/x86/boot/zoffset.h
  OBJCOPY arch/x86/boot/vmlinux.bin
  AS      arch/x86/boot/header.o
  LD      arch/x86/boot/setup.elf
  OBJCOPY arch/x86/boot/setup.bin
  BUILD   arch/x86/boot/bzImage
Kernel: arch/x86/boot/bzImage is ready  (#5)
```

Install the Linux kernel modules

```
$ sudo make modules_install
```

The end looks like this.



```
  SIGN    /lib/modules/6.5.1/kernel/net/psample/psample.ko
  INSTALL /lib/modules/6.5.1/kernel/net/ife/ife.ko
  SIGN    /lib/modules/6.5.1/kernel/net/ife/ife.ko
  INSTALL /lib/modules/6.5.1/kernel/net/openvswitch/openvswitch.ko
  SIGN    /lib/modules/6.5.1/kernel/net/openvswitch/openvswitch.ko
  INSTALL /lib/modules/6.5.1/kernel/net/openvswitch/vport-vxlan.ko
  SIGN    /lib/modules/6.5.1/kernel/net/openvswitch/vport-vxlan.ko
  INSTALL /lib/modules/6.5.1/kernel/net/openvswitch/vport-geneve.ko
  SIGN    /lib/modules/6.5.1/kernel/net/openvswitch/vport-geneve.ko
  INSTALL /lib/modules/6.5.1/kernel/net/openvswitch/vport-gre.ko
  SIGN    /lib/modules/6.5.1/kernel/net/openvswitch/vport-gre.ko
  INSTALL /lib/modules/6.5.1/kernel/net/vmw_vsock/vsock.ko
  SIGN    /lib/modules/6.5.1/kernel/net/vmw_vsock/vsock.ko
  INSTALL /lib/modules/6.5.1/kernel/net/vmw_vsock/vsock_diag.ko
  SIGN    /lib/modules/6.5.1/kernel/net/vmw_vsock/vsock_diag.ko
  INSTALL /lib/modules/6.5.1/kernel/net/vmw_vsock/vmw_vsock_vmci_transport.ko
  SIGN    /lib/modules/6.5.1/kernel/net/vmw_vsock/vmw_vsock_vmci_transport.ko
  INSTALL /lib/modules/6.5.1/kernel/net/vmw_vsock/vmw_vsock_virtio_transport.ko
  SIGN    /lib/modules/6.5.1/kernel/net/vmw_vsock/vmw_vsock_virtio_transport.ko
  INSTALL /lib/modules/6.5.1/kernel/net/vmw_vsock/vmw_vsock_virtio_transport_common.ko
  SIGN    /lib/modules/6.5.1/kernel/net/vmw_vsock/vmw_vsock_virtio_transport_common.ko
  INSTALL /lib/modules/6.5.1/kernel/net/vmw_vsock/hv_sock.ko
  SIGN    /lib/modules/6.5.1/kernel/net/vmw_vsock/hv_sock.ko
  INSTALL /lib/modules/6.5.1/kernel/net/vmw_vsock/vsock_loopback.ko
  SIGN    /lib/modules/6.5.1/kernel/net/vmw_vsock/vsock_loopback.ko
  INSTALL /lib/modules/6.5.1/kernel/net/nsh/nsh.ko
  SIGN    /lib/modules/6.5.1/kernel/net/nsh/nsh.ko
  INSTALL /lib/modules/6.5.1/kernel/net/qrtr/qrtr.ko
  SIGN    /lib/modules/6.5.1/kernel/net/qrtr/qrtr.ko
  INSTALL /lib/modules/6.5.1/kernel/net/qrtr/qrtr-mhi.ko
  SIGN    /lib/modules/6.5.1/kernel/net/qrtr/qrtr-mhi.ko
  INSTALL /lib/modules/6.5.1/kernel/virt/lib/irqbypass.ko
  SIGN    /lib/modules/6.5.1/kernel/virt/lib/irqbypass.ko
  DEPMOD  /lib/modules/6.5.1
vikram // ~/linux-6.5.1 >>                                                                  11:20:20
```

Install the Linux kernel

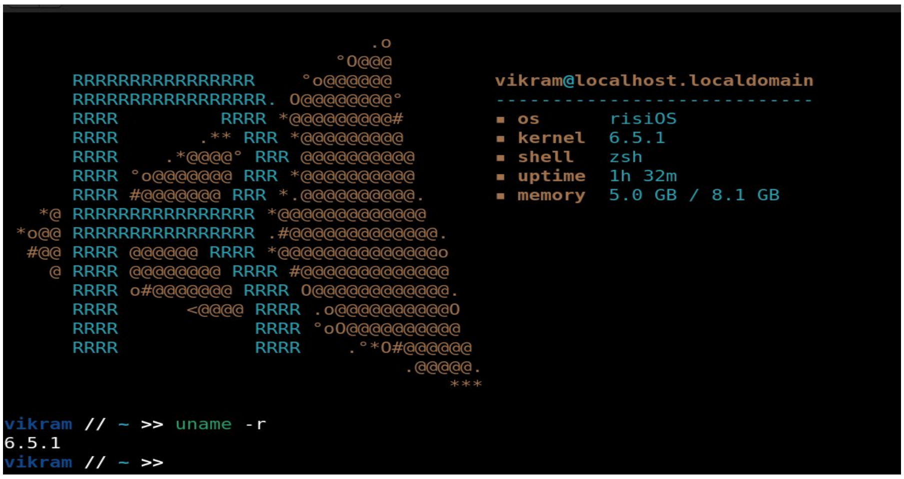
So far we have compiled the Linux kernel and installed kernel modules. Now let us install the kernel itself:

```
$ sudo make install
```

You have successfully installed the kernel. And the kernel boot option loads into grub menu

Updated Grub

Now the grub boot loader contains the new version of the kernel. You can boot into the kernel that you have just downloaded and run the $ uname –r command to check the current kernel version for making sure.



```
                                .o
                             °O@@@
      RRRRRRRRRRRRRRRR    °o@@@@@@         vikram@localhost.localdomain
      RRRRRRRRRRRRRRRRR. O@@@@@@@@°        ----------------------------
      RRRR         RRRR *@@@@@@@@@#        ■ os      risiOS
      RRRR       .** RRR *@@@@@@@@@        ■ kernel  6.5.1
      RRRR    .*@@@@° RRR @@@@@@@@@@       ■ shell   zsh
      RRRR °o@@@@@@@ RRR *@@@@@@@@@@       ■ uptime  1h 32m
      RRRR #@@@@@@@ RRR *.@@@@@@@@@@.      ■ memory  5.0 GB / 8.1 GB
   *@ RRRRRRRRRRRRRRRR *@@@@@@@@@@@@@
 *o@@ RRRRRRRRRRRRRRRR .#@@@@@@@@@@@@@.
  #@@ RRRR @@@@@@ RRRR *@@@@@@@@@@@@@@o
    @ RRRR @@@@@@@@ RRRR #@@@@@@@@@@@@@
      RRRR o#@@@@@@@ RRRR O@@@@@@@@@@@@.
      RRRR      <@@@@ RRRR .o@@@@@@@@@@@O
      RRRR            RRRR °oO@@@@@@@@@@@
      RRRR            RRRR    .°*O#@@@@@@
                                   .@@@@@.
                                       ***

vikram // ~ >> uname -r
6.5.1
vikram // ~ >>
```

<u>Conclusion</u>

Congratulations you have successfully installed and booted into latest kernel version alongside our current version of kernel.