Virtual computing

Name of student

Name of professor

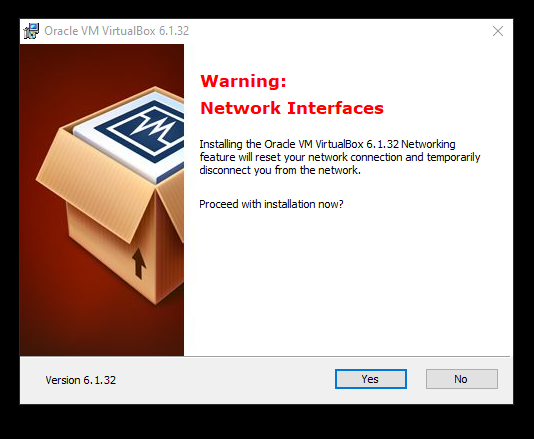
University

Course

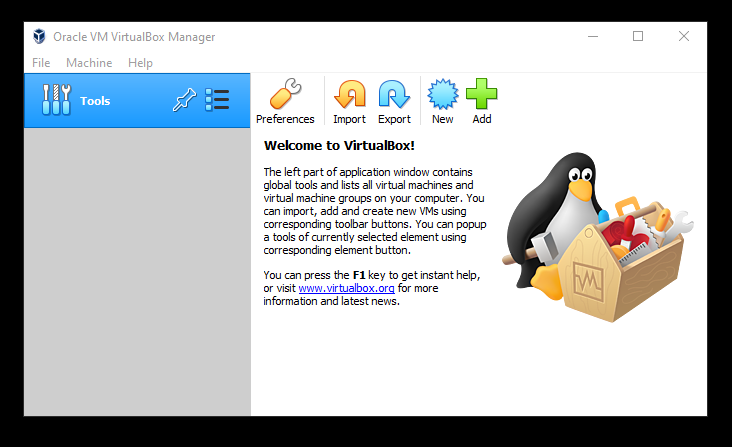
Date

**Installation:**

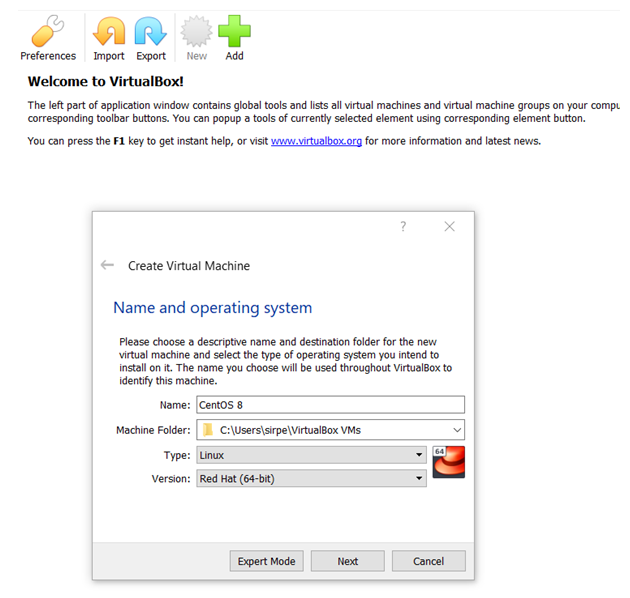
The below processes are followed when installing virtual box in the computer system. In this case it’s the windows operating system:



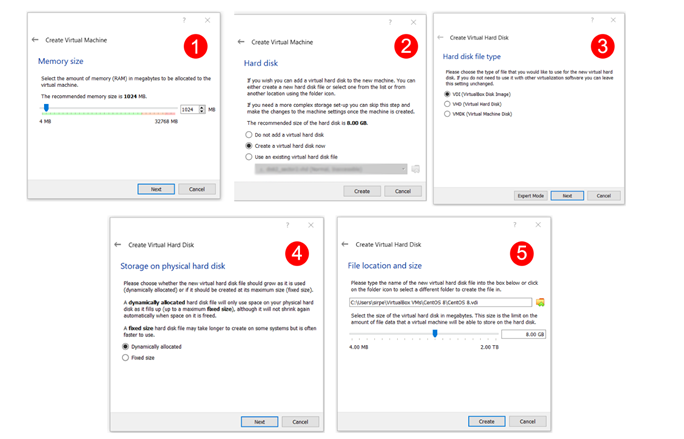
Once installed, the next step is to create the name of the machine and import it.



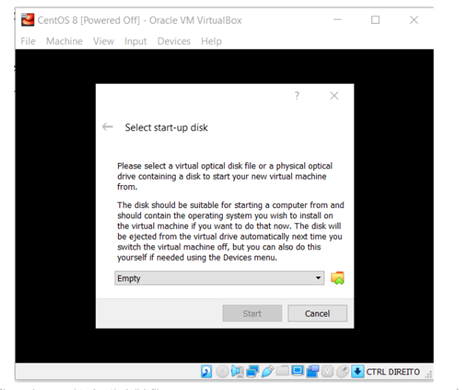
Then we select the operating System as Linux and then load the Red hat version in the next drop down



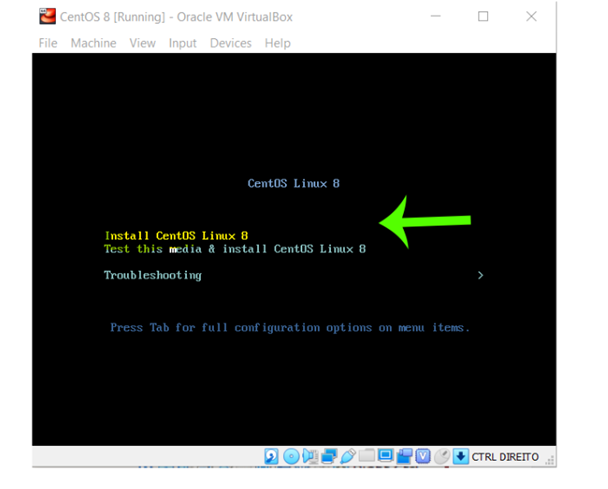
In the next steps, we allocate memery size of at least 15GB, this will be efficient to run the program.

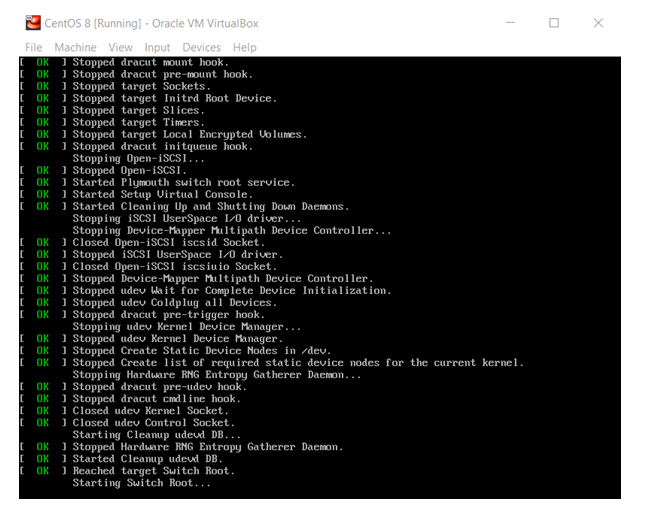


In the next we load our CentOS ISO file that we had downloaded from the official CentOS website

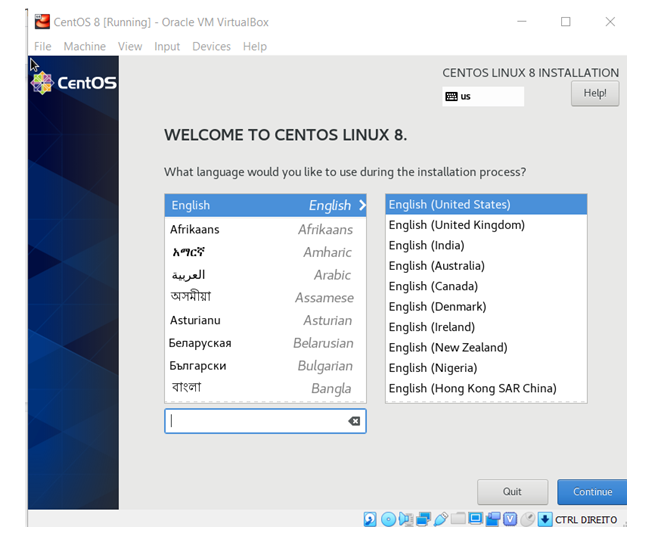


Then the CentOS begins to install

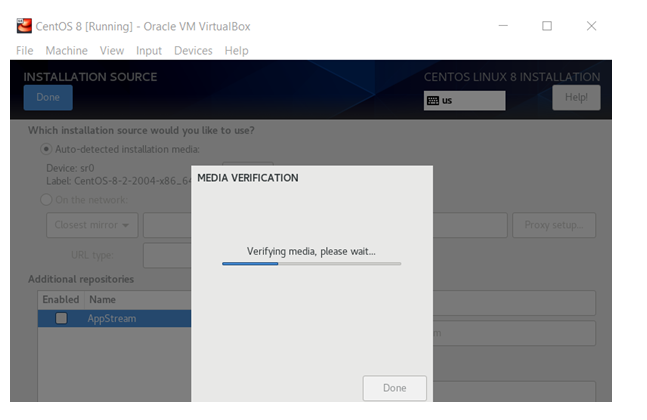


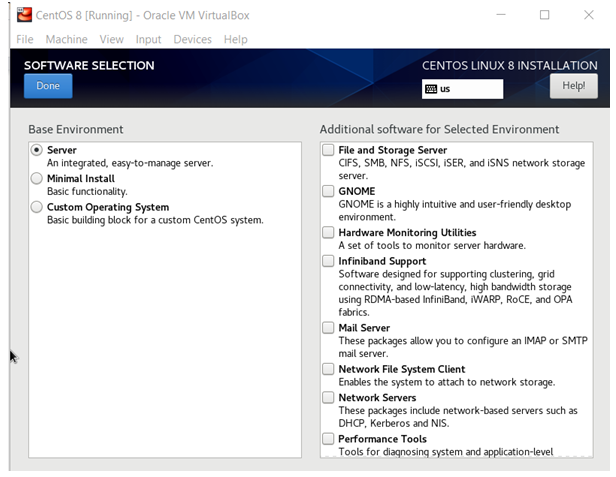


During the installation process, we are asked for language to select, in this context, we chose English

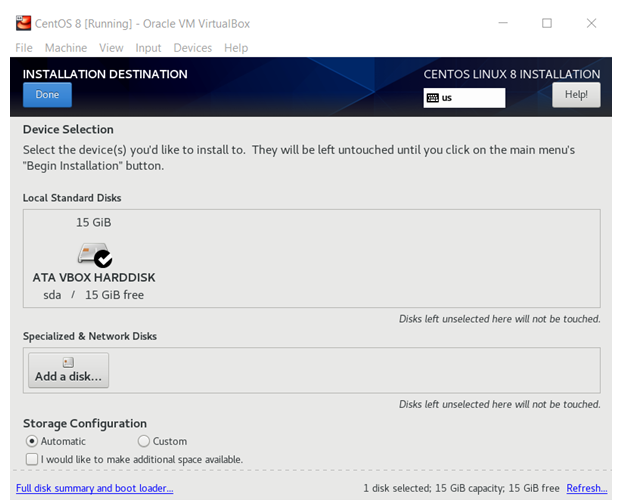


Then the OS installation verifies that all media capabilities at machine state are all set.

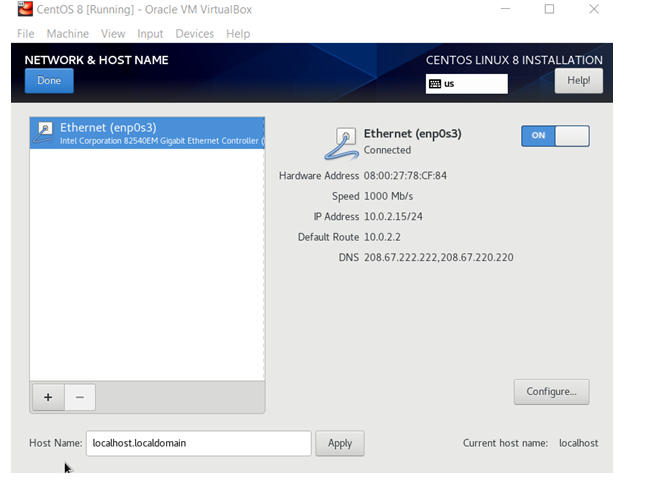




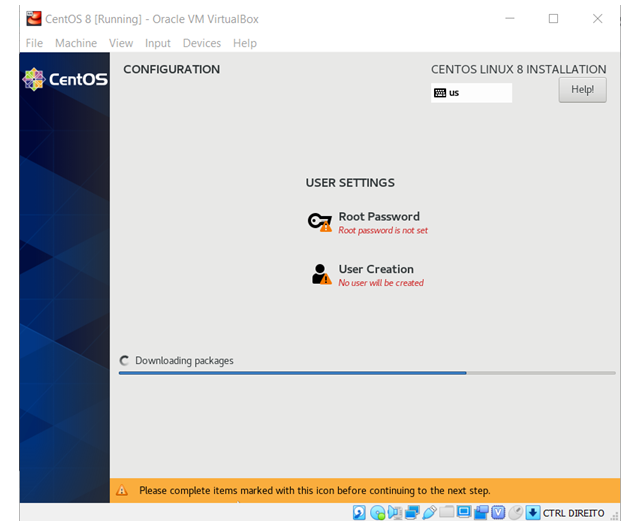
In the next step, we chose the installation directory, so we select the 15GB memory space we had earlier created



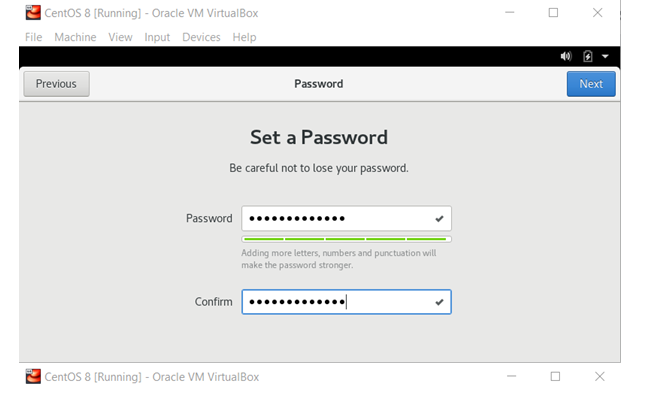
In the next step, we configure the network capabilities of the OS. It detects the local IP address, the route network and the DNS services address.



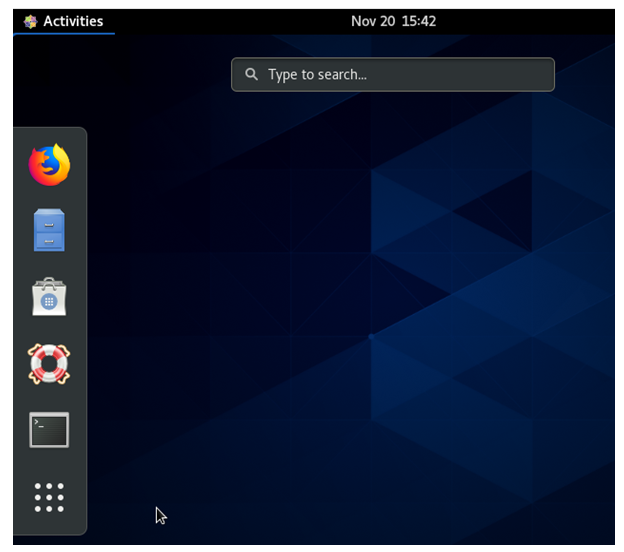
In the next step, we configure the user settings of name and passwords, this is required every time a new installation is done. By default, this becomes the admin user.



Then we use the new details to login to the OS.



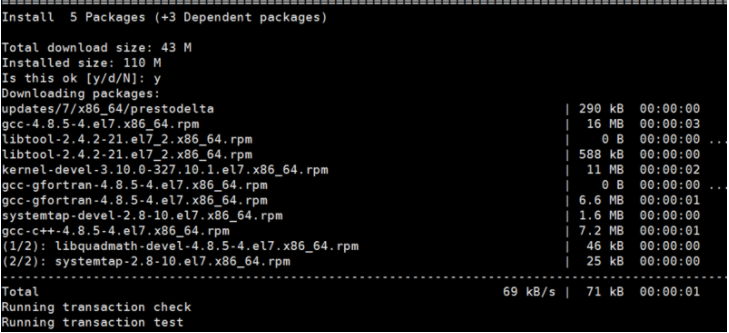
First screen after login appears as below:



Next we fire up terminal and begin to install development tools using the commands below:

yum groupinstall 'Development Tools'

This shall prompt the system to show the full list of programs to install.



Once installed, we can write our first program by writing a simple C++ program as below, to print “Hello World to the screen”

#include <stdio.h>

int

main (void)

{

  printf ("Hello, world!");

  return 0;

}

We write the program and then save it in file called hello.cpp. Then we navigate to gcc and invoke it with the below command. This will print a simple text “Hello World”, on the screen.

gcc -Wall hello.c -o hello

In the next step, we write a C++ program to print processes on to the screen using the fork() method. In this case, we are targeting to print the string:

(“These processes are being forked !! \n"). There are a total of three fork() methods that have been called so assuming we generate the first parent, the second line will print two children and the third children will print 4 process ids. A total of 6 processes on the screen:

Forking this program to print multiple IDS:

#include

#include "unistd.h"

int main()

{

fork();

fork();

fork();

printf("These processes are being forked !! \n")

return 0;

}