Fundamentals of Power Electronics
Prof. L. Umanand
Department of Electronics Systems Engineering
Indian Institute of Science, Bengaluru

Lecture - 13
Designing the circuit

In this video capsule, we shall look into the installation process for installing the open source software packages under our system such that we will be able to make effective simulation for this course. The plan is to install gEDA software tool set. This is the electronic design automation tool set and then we shall install NG spice a very popular package for circuit simulation and then after this we shall install octave. Octave is a MATLAB like environment which is very helpful in scripting and automating many of your design tasks.

So, these three set of software packages we need to install I will show; I will show you by taking a walk through and then configuring the system to our needs. I have a Fedora 23 Linux desktop. So, I will show you by show by taking a walk through the process in this particular operating system; my students have checked it out on Ubuntu also.

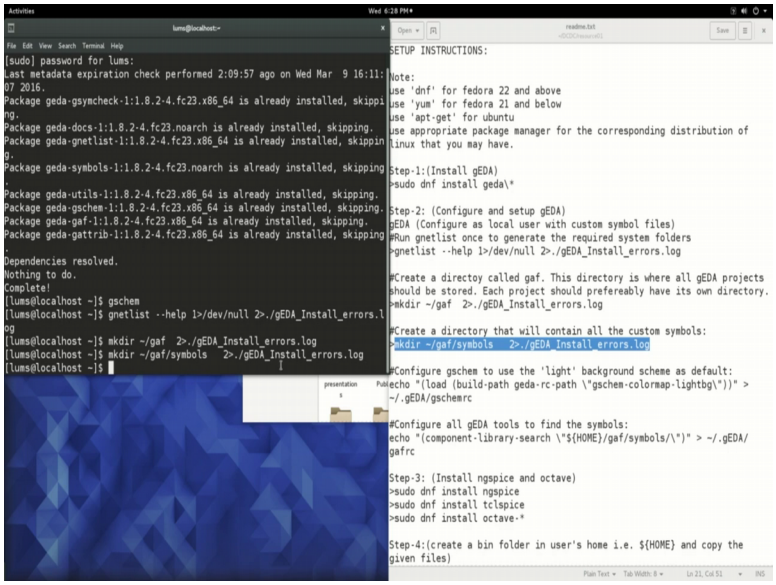
So, I guess that it should be fine even in other Linux distributions. On Windows I have not tried nor have my students tried. So, I am not too aware on how it will perform on Windows. So, I suggest that if anybody of you are trying it on Windows, that is EDA, ngspice and octave you can try and then let us know how it performs or you can use alternative packages whatever is available on the Windows platform.

With this we shall start now the process of installing these software packages. We shall begin the installation process by first downloading the resource from the Google drive. This is the Fedora desktop and let us open the folders and here I have created a DCDC converter folder and within that folder I have downloaded this resource01 dot zip folder file.

This is located in Google drive and the link for this is given on the course website. This has two folders and one file readme dot txt file there is a bin folder and the symbols folder of course, I will explain to you later what are the functions and how they will use

that one. So, for now extract unzip it and within that you see this folder. Repeat dot txt gives you a step by step instruction on how to go about this process of installation.

(Refer Slide Time: 04:25)

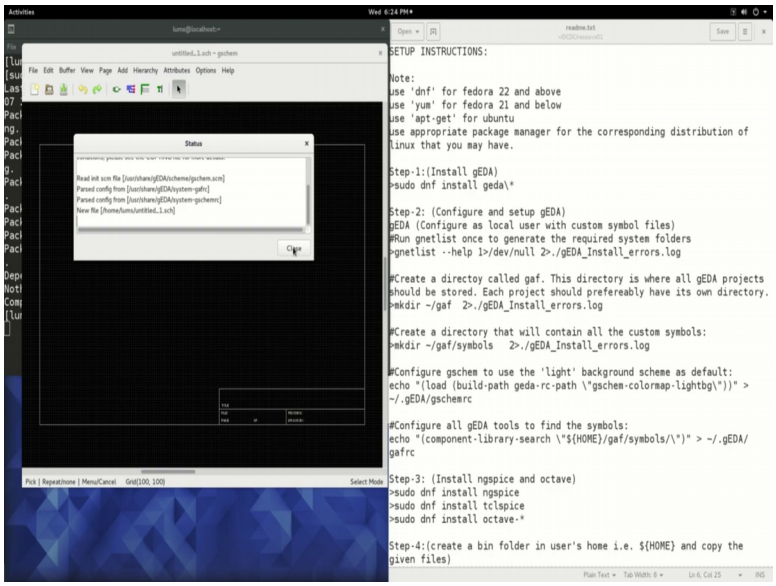


So, open that read me dot text file and keep it to one side. And all you have to do is follow through these steps the commands are given here you just may have to copy and paste it if it is a Fedora system or equivalently typing the command line commands if it is any other Linux system.

Again go to the desktop and open a terminal let me adjust the size so, that I am able to see that. Now let us copy this and paste it here, it will ask for the password and then execute it. So, in my case I have already installed all these packages, it will search and look for the packages in the repository and install them. So, you have the gschem check the gschem docs, netlist symbols utils g shell the geda g schematic this is what we will be using at great length in this particular course.

So, dependencies resort nothing to do everything has been installed. In your case if you have not installed geda you will see that it will give a list of packages and ask you whether to install or not you say yes and it will get installed. Next we need to configure g e d m you can. In fact, open g schematics and see what happens what pops out. So, let me type gschem enter and you will see h u i something like this open up.

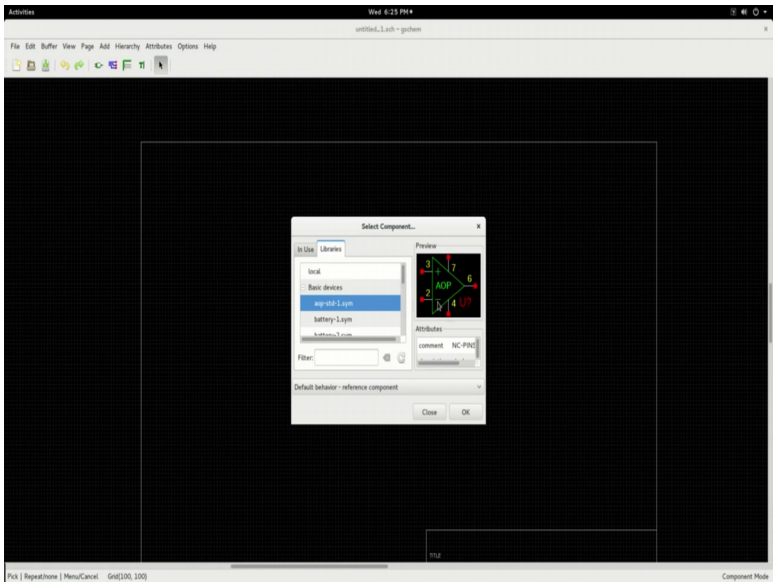
(Refer Slide Time: 06:29)



This is a status dialog which opens which just gives you an indication of the path which from where it takes this is the g schematic r c file gap for c file where it is located it is located and user share gEDA; however, these are in the this need root permission to modify these files.

However if you want to customize you need to have copy of these files in your user dot g e d f folder which I will tell you about shortly. Close this is how the g e d a user interface will look like. This is having a dark background some people may like this background fine, but I would prefer to have a light background and I would like to have the default gschem opening with a light b g e.

(Refer Slide Time: 07:33).



Now here is the components and this is where you have the libraries, where a list of components coming here and then you can choose these components to place onto the sheet ok. Close this for now we know that g e d a is installed working and running and it is opening up. At this point I would like to configure it I will show you how to configure it to make it into a light background 1, you may want to change it or you may not want to change it you may want to retain the dark background it is up to you, but let me show you how to go about doing that.

So, you see here the readme dot txt file gnetlist just go through this command syntax, copy and then paste. So, this is just a g g e d a command what it would do is that, it will create dot gEDA folder in the users root directory and into that directory is where we would like to have the r c files. So, if I go into home and to control h are alternate the hide it will show the hidden files with the dot prefix. So, you see that this has got created dot gEDA and into this you have logs and a geometry file. Here we have to put the gschemrc the g a f r c files anyway.

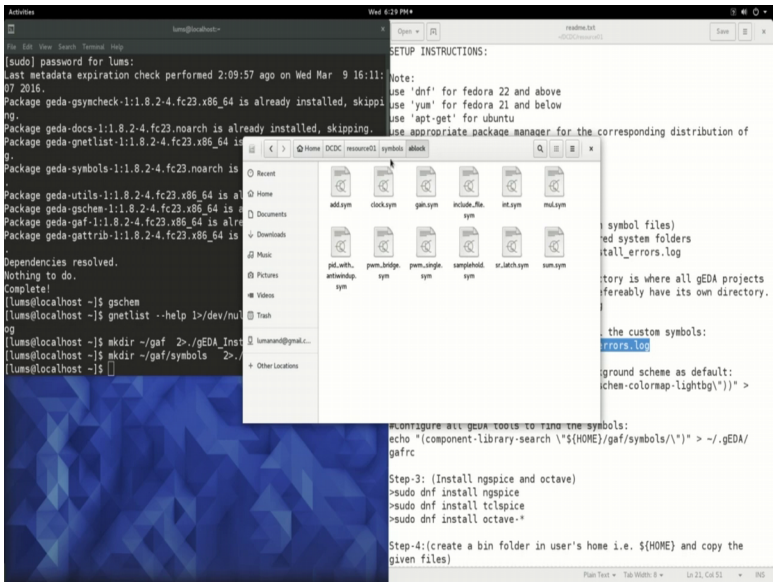
We will come back to that later and you will see what to do next. Next we would like to make a g a f directory and this is where we would like to place the symbols of g e d a symbol the custom symbols that we make that has to be placed within this. So, we will give it a place. So, just copy this and paste it here onto the terminal and press enter. So,

what it would have done now is create this g a f directory g a f. It is right now empty, but we want to fill it up with symbols custom symbols.

Next we create within g a f another subdirectory called symbols that is where we shall place our symbols. Copy paste and enter this if two then redirected to errors not log is just an indication if there are any errors to see you can go and look at this log file see what happened. There are no errors nothing will be entered there.

So, now after having run this particular line within g a f you would have this and inside that let us place the symbols. So, what we shall do? We go into this DCDC converter folder and within that we have the resource folder and in that we have a symbols folder in the symbols folder, there are three subfolders one is called a block a bond and a comps.

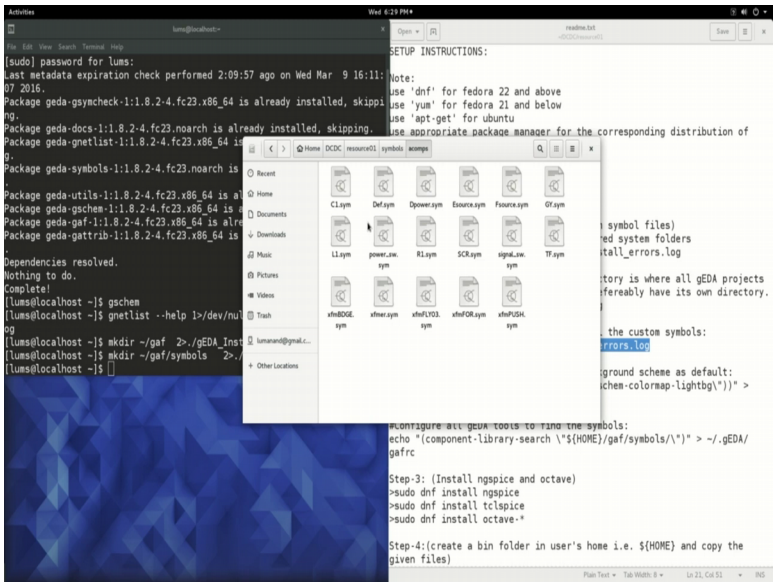
(Refer Slide Time: 11:45)



A block has many symbols associated with add clock again integration multiplication pid pwm bridge, pwm single phase sample and hold s r latch summations so, on and so, forth.

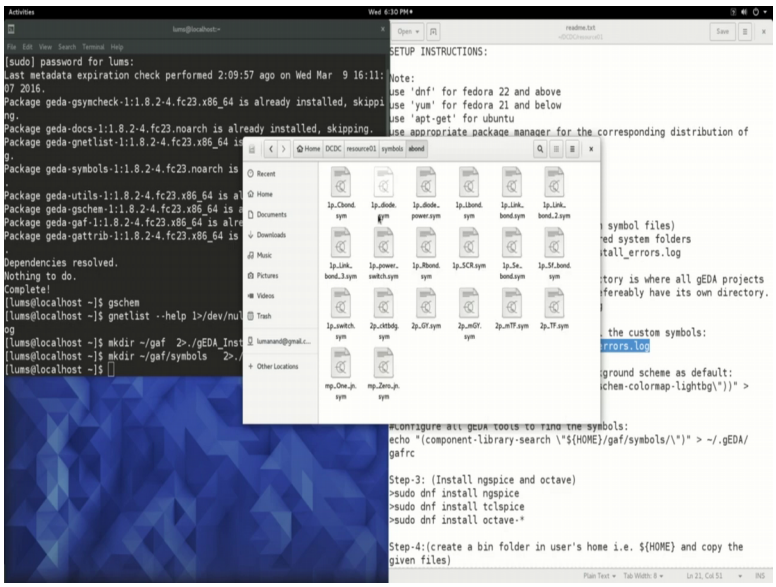
So, these are custom blocks more something more logic oriented not much of power flow and interesting to have them. I will come to this later we may not be using this, but anyway you I would like to share that with you.

(Refer Slide Time: 12:17)



The a comps are the component block, the default are the genetic components capacitor Default diode, power diode, source Flow source current source GYrator inductance, power switch resistance SCR, signal switch transformer a transformer bridge, transfer for bridge another flyback transformer, forward converter transformer PUSH pull transformer in the normal transformer. So, these are symbols which will be useful.

(Refer Slide Time: 12:55)



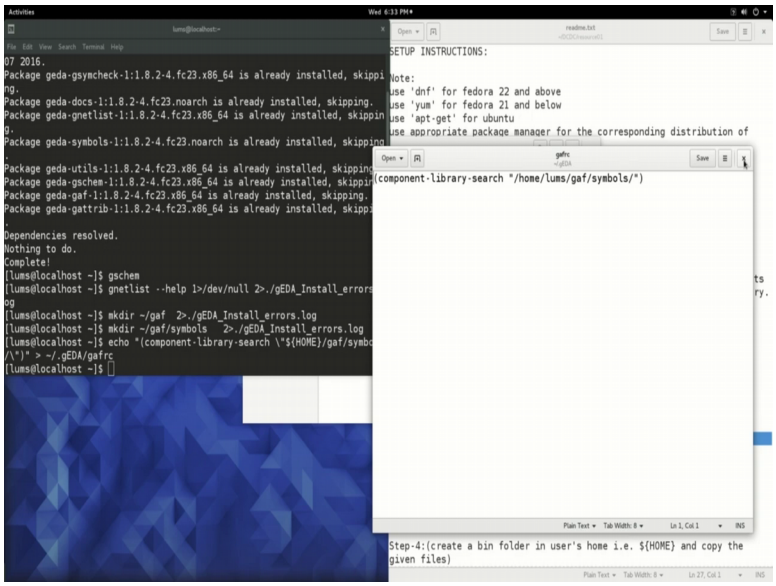
This is particularly symbols used for bond graph simulation. So, people who are familiar with bond graphs and have done bond graph modeling and simulation they may use it.

But right now we shall not use it, but; however, I will share this thing this folder also with you, you copy all these copy and go into the g a f go into symbols and paste them and now we are in business we have all the symbols in the proper directory.

Now, we have to assign the proper path. So, now, that we have done that, we will assign the paths and create the respective files. Now there are two commands here one is a command component library search, where it will search through this path dollar within braces HOME is a environment variable, which will take the root users r and into gaf and into symbols now this should be the path of all the component library search and they should get reflected in your g g schematic when you open it. And this is pushed into a file, which is residing it dot gEDAand into a g f rc file if gaf r c is not there it will get created.

So, this is your own custom gaf r c file it will look such far there and then see this and then set the path environment path. So, we can have this copy and paste it here and run that one. So, once you run that if you go down into the hidden dot gEDA file I have unhidden that. So, into that you will see that there is a gafrc file that is created. So, into the gafrc file see whatever we have typed.

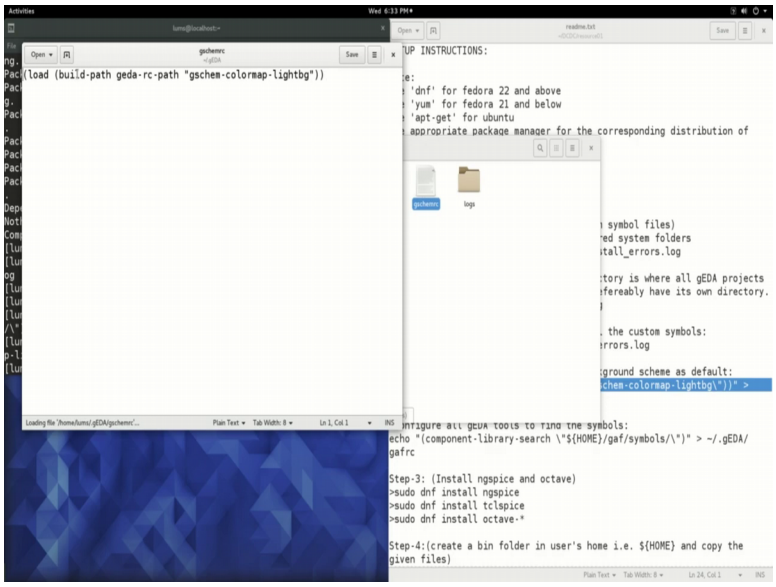
(Refer Slide Time: 15:27)



Common library search that would have come in here; so, when gEDA opens up it will look into this file see this path declaration and then appropriately remember the path. Next we would like to make the background light. So, that is this load build path gEDA

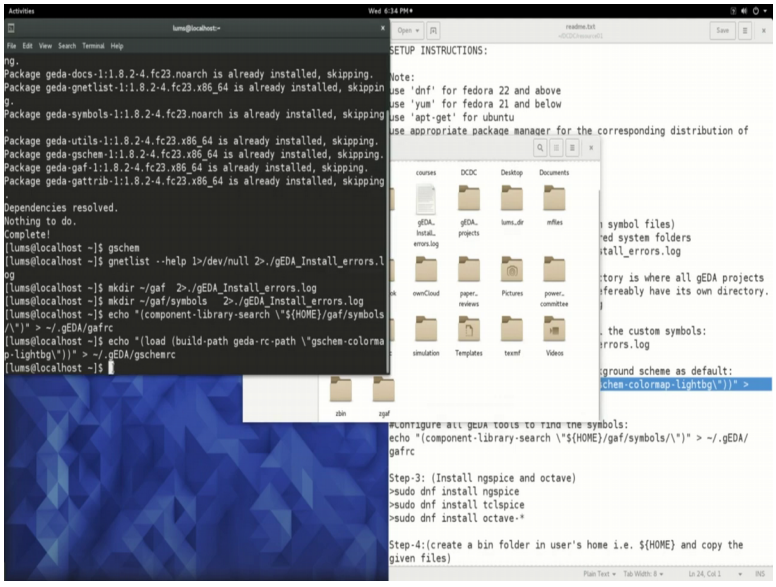
gschemrc into that gschemrc we will put this gschem color map lightbg. So, instead of that bg we want a light bg and then put that into dot gEDA gschemrc file. So, we will copy that and paste it here. So, that would have got entered in there. So, you see that gschemrc has been created and within that this has been copied.

(Refer Slide Time: 16:27)



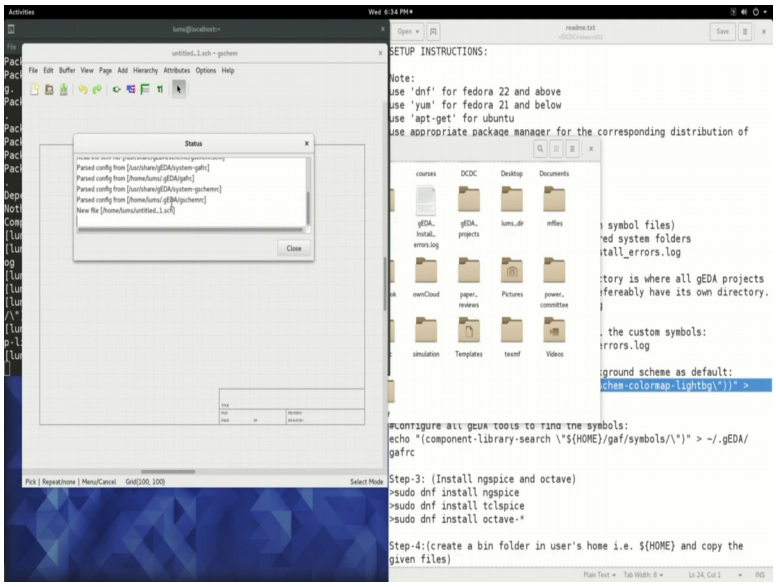
So, it will look in there and automatically put in the proper background color. Now I have to control h and hide all the dot prefix files. So, that it does not clutter up the folders.

(Refer Slide Time: 16:47)



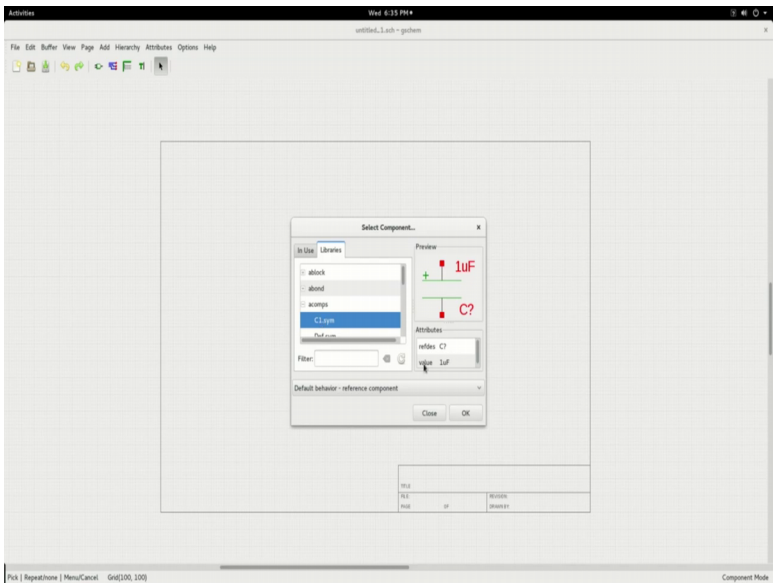
Now, again now you can run gschem.

(Refer Slide Time: 16:55)



Now, you see the background is light, see that it takes gafrc from the local users dot gEDA directory gschemrc also from the local. So, it is working properly you can open it to full screen you can use the wheel button of the mouse to zoom in and zoom out.

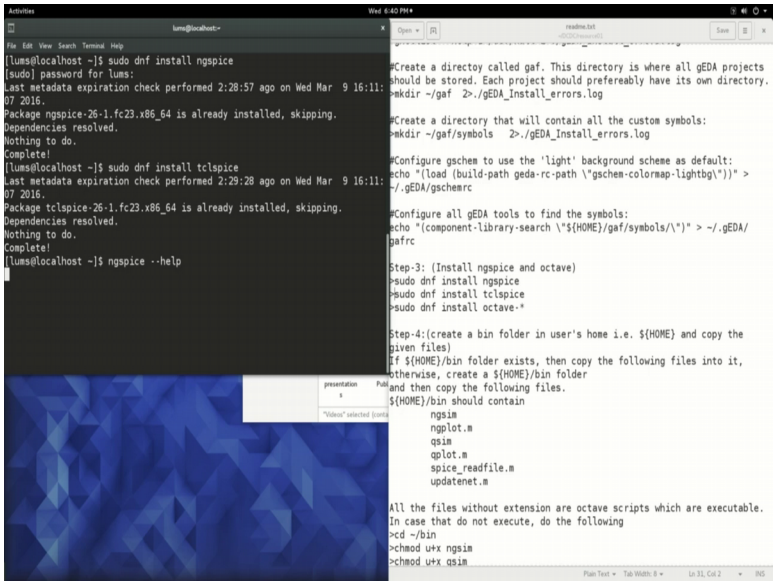
(Refer Slide Time: 17:21)



And then go into the components, now you see that our custom libraries ablock aborn acomponents are reflecting here. So, if I open it they are all coming in here the symbols

and that is very nice and then the component now close this, and back again to the terminal going back to the readme dot txt file.

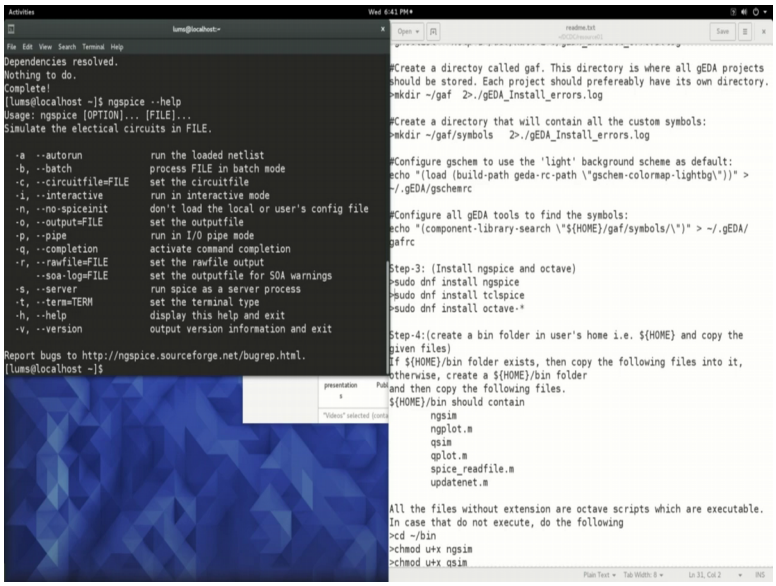
(Refer Slide Time: 17:59)



The next part of the simulation is installing ngspice. So, installing ngspice is pretty straightforward you will just copy this, dnf install ngspice and let me clear this terminal screen control l and that is cleared test. So, sudo they are not install ngspice enter the password and of course, in my case here it is already installed and in case you yours is not installed, it will say that this particular package has to be installed whether to install or not click yes and then you will be through.

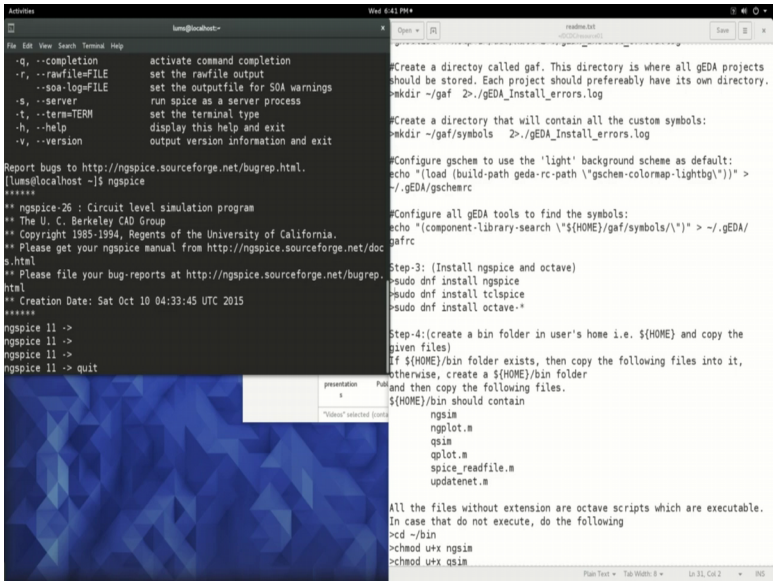
And then after that you need to install tclspice also though actually you may not used tclspice, there are some libraries in tc lspice that are useful for ngspice and kindly to install tclspice two and here again it is already installed here and then you will get such a message. So, now ngspice is also in place and if you want to test that you just type in ngspice dash dash.

(Refer Slide Time: 16:29)



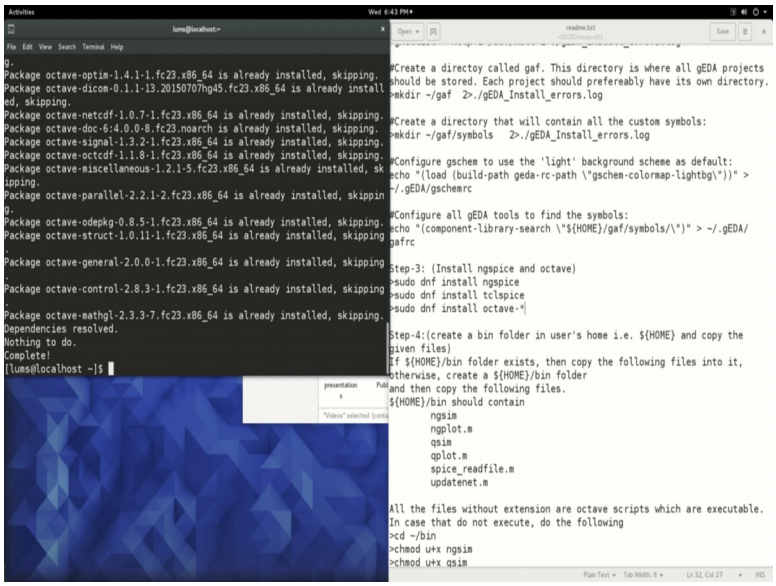
So, that will give you a menu of options and how to use go through that and I would recommend that you download the ngspice manual and read through it, if you are not if you do not know how to use spice. It is very very similar to p spice of the windows it is actually a word out of the bugrep spice and quite universally used by many people..

(Refer Slide Time: 20:05)



You can type just plain ngspice and it will go into the ngspice environment. It is ngspice version 26 plus and a developed by Berkeley CAD Group and then you have the ngspice environment prompt. So, I will quit from this here knowing that ngspice is working.

(Refer Slide Time: 20:29)

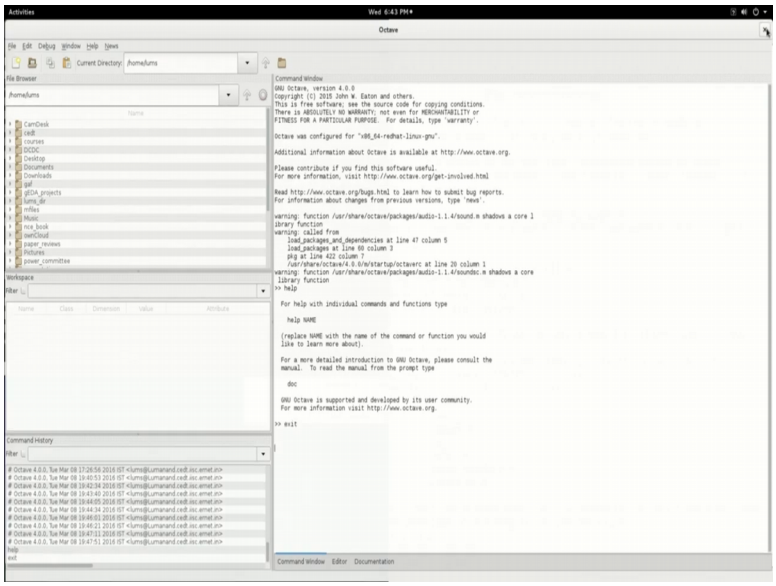


And now we have one more package to install and that is octave. Octave is a very very powerful tool you should have that it is a open source equivalent of MATLAB sometimes I find it much more powerful than MATLAB, but anyway that is a good package to have on your system.

Paste it here, I have used a dnf install I have used dnf everywhere doing this process because this is Fedora 23; Fedora 22 onwards we have to use dnf for the package installer as the package installer before Fedora 22 it used to be yum. So, people having Fedora version earlier than 22 kindly use yum y u m. And the case of Ubuntu people can use the apt get package installer.

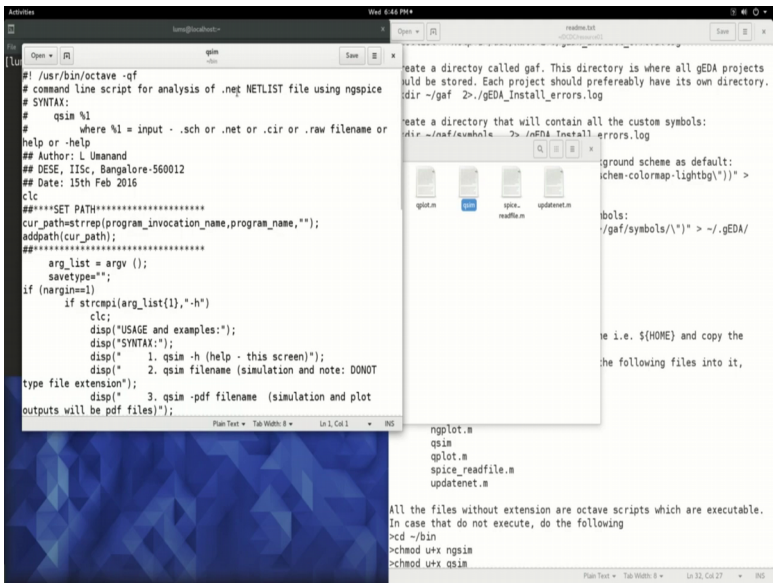
So, use this press enter and the rest the package installer will do the job and here again you see everything is already installed and dependencies are resolved nothing else to do. Now, to test whether an octave is running properly or not just type octave; you will now see a GUI popping up very nice GUI.

(Refer Slide Time: 21:53)



It looks much like the MATLAB GUI this is where the workspace window is and this is where you will probably do all the work. So, octave is working this is version 4 plus. Most of the earlier versions octave they may not have a GUI integrated, but it will work on the terminal just like the ngspice environment. So, if you want to work in such an environment which I also prefer you type octave dash dash no gui.

(Refer Slide Time: 22:37)

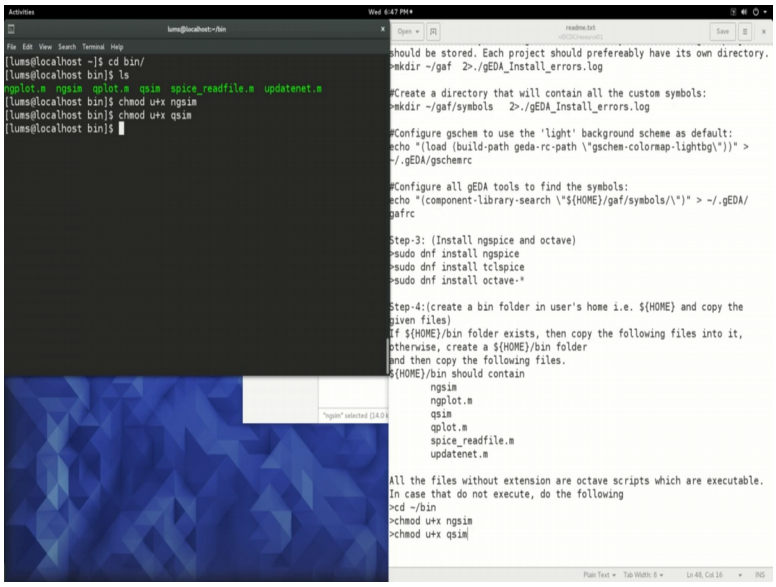


So, then you will get it in the terminal this is very fast and you will be able to do equivalently, good programming with this terminal environment to. So, let me quit that

and with this we have installed three packages one is the EDA Electronic Design Automation the next is the spice that is the ngspice and tcl spice have been installed and then octave and we have installed it on a federal system. I hope that you will not have a problem and other distributions to kindly follow it and then look into help on the net to if you run into problems and there is of course, a forum to discuss. There is one last job which we need to do which is the step 4 and that is something which you need not do, but something that I would like to share and which may be of help

So, this is not compulsory. So, what I request you to do is, if you do not have a bin folder in your user home create one create a bin folder. So, you will see that a bin folder is created. Then go into the DCDC there is this bin folder there are six files here; ngplot, ngsim qplot, qsim, spice read file and this update net dot m. So, you just copy all of them and go back there into the bin and paste fine; ngsim and qsim should be made executable, they both are octave scripts octave scripts meaning that it starts with a hash exclamation user bin octave. So, it calls it runs octave directly from the terminal. So, these are useful scripts I will tell you more about this at the time and they are needed, but its better you keep it ready.

(Refer Slide Time: 24:59)



So, to make these two executable in the readme file, I have put these two commands change mod u plus x qsim. So, they will be made executable. So, what you have to do?

Go into the terminal cd bin. So, you see that it has the six files chmod u plus x ngsim. So, that will be made executable chmod u plus x q sin that also will be made executable.

So, that is it your software installation portion is over, next we can start on building our simulation schematics and simulate the circuits. This portion this installation portion is one time. So, it is worth putting the effort to do this work neatly and properly now and then forget about it next only you can think on circuits and simulation.