The “doit.sh” script

rev 7.21.23

This presented script is for those of us running Linux. The purpose is to more or less automate compiling source code. It assumes you have all of the needed library routines installed, and belong to the ‘sudo’ group. If needed, download and run <https://qso.com/qso/k7cwa/AAA_install_lib.sh>. This is Specifically Hams who use the Fldigi series of programs that make up the NBEMS suite of programs for digital communications. The (gz) compressed files are downloaded from <http://w1hkj.com> and are free. You also need to download ‘hamlib’ from https://sourceforge.net/projects/hamlib

Once a program is downloaded, then it must be compiled and installed. Since this is a Bash shell script it should work on any version of Linux. The following is how to set your system up and how the script works.

First thing to remember is to change the owner of “doit.sh” to your logon name. You can do it one of two ways. The first is to use the ‘File Manager’ and find the doit.sh file and then right-click and “Open as Root” and give your password. Then right-click on the doit.sh file, select Properties, then the Permissions tab. Now change the owner and group to your logon and any group you wish. Close it all and you should be good to go.

The other way to do this is from the command line. Using the File Manager go to where the doit.sh file has been placed. Right-click in a blank area and then “Open in Terminal”. Now left-click on the Terminal window and enter “sudo chown [your logon name]:[group name] doit.sh. An example: <sudo chown warren:warren doit.sh> (no brackets). With that having my name for the group, only I could use the program… or others could belong to your group and have access. You will be asked your sudo password of course, and you must belong to the sudo group to do that.

1. Create a folder/directory in which to put your downloaded .gz files. It can be any folder (directory) but I call mine “Nbems”.
2. Place the “doit.sh file in this folder also.
3. On each packaged file, right-click the mouse on each package in turn, and select “Extract here”. This will create a new folder with the programs name.
4. One at a time enter each folder and right-click on an open spot and select “Open in Terminal”, do hamlib first.
5. In the terminal window type (no quotes) “bash ../doit.sh” and hit <enter>. Now the script takes over.

It first asks how many cores to use. Normally you would use all you have, enter the number. If you do not enter a number it defaults to 4 cores. If you specify more cores than your computer has, it won’t hurt anything, it will just use all you have, but maybe take longer to run.

Next you will be asked if you want to hide the compiler output. The default is “Y” and you do not get to see all of the gibberish on your screen, but if there are errors, they get saved in a file called “errors.log” to be looked at later. If you select “N” then it all goes to your screen but any errors still go to the errors.log file. If “Y” (the default) is entered you sit in front of your screen with a message telling you it may take some time, and of course you wait. It does compile faster without the writing to the display screen. Another thing of note; what is defined as an error may in fact just be the result of a choice made in the source. Some programs produce “errors” that are not really errors, and the compiled program is good. If you try to install (next step), and the compiled code is bad, it won’t install, you would get an error message to your display.

When it is done compiling it will tell you if there were no errors, or if there were. If there were errors you are offered a chance to display them, or not. If yes, then the error.log is displayed. Either way, the next and final question is if you want to install the program or not. If not, it aborts. Also displayed is the total time it took to configure and compile the program.

One additional note: When you are looking at the properties / Permissions of doit.sh, if you put a check in the box “Execute: Allow executing file as a program” you can then run the program without putting the “bash” in front of it. Then you would run it with “../doit.sh”, or better yet, if you place the doit.sh file anywhere in the normal search path (like the bin folder) then all you need to do is enter “doit.sh” and it will run.

This script was written and supported by Warren Andreasen, K7CWA

Download the script file from: https://qso.com/qso/k7cwa/Scripts/doit.sh