K-splitter ADMINISTRATIVE MATTERS

STEFANIE TEUFEL

Constantly entering the famous/infamous Linux rule of three of configure, make, make install can become very tedious. If you'd like to avoid this task, or provide your less skilled friends with a graphical user interface for compiling Tar-balls, Kconfigure is just the thing.

Linux rule of three made easy

The program, by Javier Campos Morales, provides *configure*-fatigued users with everything necessary to compile and install from the source code in the usual KDE look and feel applications. The latest version of the compiler aid can be found on the homepage of the author at *http://kconfigure.sourceforge.net/*.

The way *kconfigure* works is simple: Open the program with a *kconfigure* in any terminal emulation of your choice, and marvel at the window as shown in Figure 1.

After that you have the choice as to whether you wish to use your graphical compiler assistant via the buttons or the menu bar. But first, you must trawl, by clicking on the folder icon, through the *unpacked* source directory of the program to be installed and there select the *configure* file. Unfortunately, unpacking the sources is not something *kconfigure* will do for you; here, the K-tool *karchiver* can help you along.

If you want to give the *configure* command an argument such *-with qtdir=/path/to/qt-directory*, it is advisable to take the route via the menu bar. To do this, select the item *Build/Configure with arguments…*. In the dialog window which then plops open (Figure 2) you can enter the required

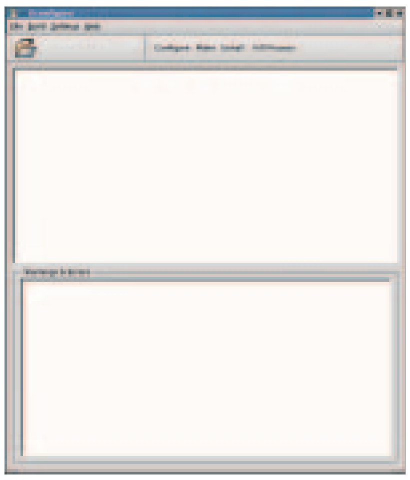


Figure 1: Compile me!

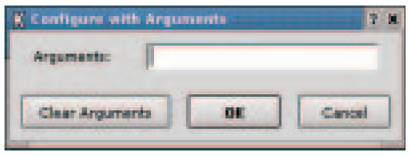


Figure 2: Always the right arguments

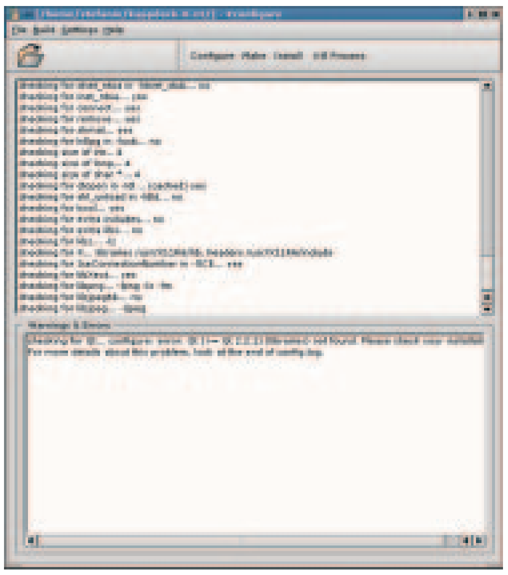


Figure 3: Once with and ...

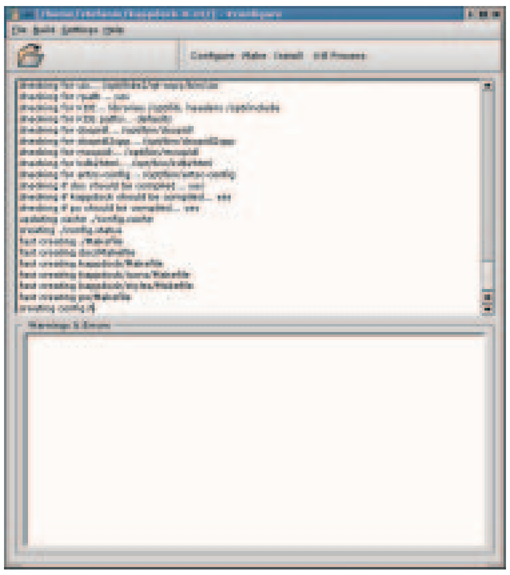
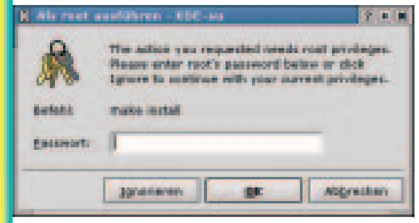


Figure 4: ... once without errors


Als root ausführen - KDE-su

The action you requested needs root privileges. Please enter root's password below or click Ignore to continue with your current privileges.

Befehl: make install

Password:

Figure 5: Identification, please

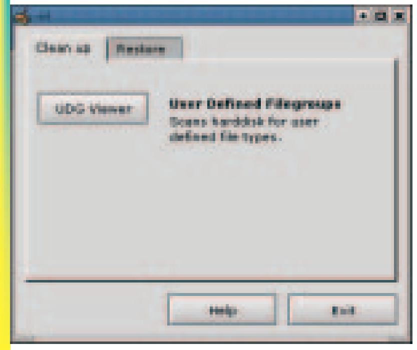
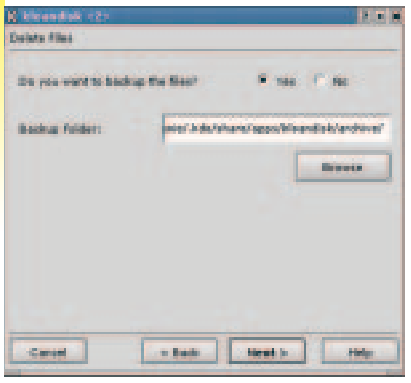

User Defined Filegroups
Scans harddisk for user defined file types.

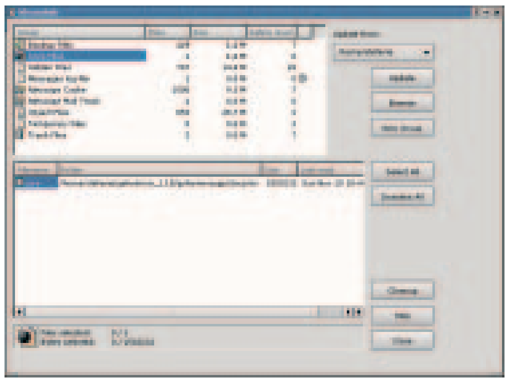
Figure 6: And off it goes


Kleandisk <2>

Delete files

Do you want to backup the files? [✓ Yes] No

Backup folder: ain/.kde/share/apps/kleandisk/archive/



[left] Figure 7: The big clean-up begins

[right] Figure 8: Better safe than sorry...

options. Anyone who is not really sure about the individual options can access these at any time, by clicking on *Build/Configure help* in the top half of the window *kconfigure*.

If you started the configuration process via the menu bar or by clicking on the *Configure* button, the program will immediately set to work without any further challenges. You can monitor progress in the upper half of the window, while error messages or warnings appear in the lower half (Figures 3 and 4).

The commands *make* and *make install* are treated exactly like the *configure* command by *kconfigure*. With one small but special difference: Before the actual installation the application checks which user you have logged on as. If you are travelling as a normal user, *kconfigure* will still at first confront you with a dialog box, as in Figure 5, in which you must enter the *root* password before it continues.

If you want to interrupt one of the commands you have given, all you need to do is click on the button *Kill Process*.

Cleared up

Anyone who has become too carried away by the simple handling of *kconfigure* may now break out in beads of sweat when taking a look at the amount of space occupied on their hard disk. If you are plagued by a bad conscience, you should risk a look at *Kleandisk*. The latest version of this easy-to-use disk cleaner can be found at *http://www.casema.net/~buursink/kleandisk/*.

Contrary to normal practice, though, at this point you should download, not the latest version *Kleandisk-2.0beta1*, but its predecessor *Kleandisk-1.2beta2*. The reason: The new version provides support for the first time for the removal of unused *rpm* packets. Unfortunately, though, this leads to problems with some versions of *rpm*, so that on various computers - for example on Red Hat 7.0 - the program will not compile.

Call up *kleandisk* either via the *K*-button/ *Applications/ Kleandisk* or by entering *kleandisk &* in a terminal emulation. After that you will see a window, as in Figure 6.

Click there on the button *UDG Viewer* in the *Clean Up* tab. The ominous abbreviation stands for User Defined Group. In the next window you can define the directory which *kleandisk* is to clean up for you, and also the file types, which the program is to give their marching orders. *kleandisk* then sets about searching for the less useful files on your system and sooner or later presents you with its inventory in the lower half of the window, as in Figure 7.

At that point I decided that I really do not need the **core-file** indicated below, and also informed the disk cleaner of this decision by clicking on the green box next to the core file. After that it is enough to click on the *Cleanup* button. *kleandisk* then begins to communicate cheerfully with you. It dutifully asks you window by window (as for example in Figure 8), whether you want to move, delete or archive the selected files, if you might perhaps prefer to make backups of the files to be deleted, and if so, where should they go.

In the last step, you find out how much space you are saving overall as the result of making these decisions. To get rid of the file now once and for all, click on *Finish*, which lets *kleandisk* off the leash...

Permission granted

Wanting rights and getting rights under Linux as a truly multi-user system are two different kettles of fish. As you may already have noticed, your system differentiates very precisely as to who exactly can read, write or execute the diverse files and programs on your computer.

And to avoid confusion, information is stored with each file as to whether the owner, group member or other users can read, write or execute the respective file. With the command *chmod* you can obviously change these access rights at any time.

There is a graphical front-end for this command at *http://www.leeta.net/kchmod/* in the KDE-Look named *kchmod*, with which the setting of access rights is twice the fun (Figure 9).

Simply select, via *File/Open*, the file you want to edit and choose between the options on offer. Is the file to become *writ(e)able*, *read(able)* or *executable*? The choice is yours. After that, quickly save the change with *File/Save* and it's a done deal - if only it were always so simple to guard one's rights. ■

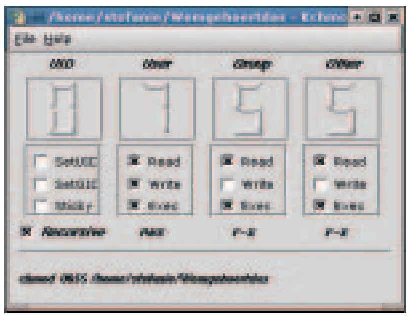


Figure 9: Everything legal?

Tar-ball: *The program tar is an archiving tool which is well-known under Unix. A collection of data packed together with this into a file is usually called a Tar-ball and has the file ending .tar.gz or .tgz, if it has been put together with tar and compressed with the program gzip.*

core-file: *The last memory retrieval of a crashed program is retained for posterity in files called core. Experienced programmers can find out the cause of the crash from these with the aid of a debugger, but for anyone else these files are simply a waste of space.* ■