REF:

https://boozlachu.medium.com/using-kconfig-for-own-projects-deb21e0d1804

# Using kconfig for own projects

# Intro

Every Linux professional write scripts. Someеimes light, linear. Sometimes complex script with functions and libs(yes, you can write your bash-library for use in other scripts).

But some of the scripts need a configuration file to work. For instance, I wrote a script that builds the ubuntu image for pxe, and I need to change the build process without build-script changes. The best way to resolve this task is to add configuration files.

Writing this config by hands is not a great idea:

* every user should know all available options
* hight risk of incorrect data entry
* uncomfortable
* maybe something else

Linux kernel gave us a great solution — KConfig. This system resolves all these troubles:

* New config generated automatically
* You can set the default value
* The previous version of config saved automatically
* text/graphical interface available

These are not all the benefits. You can see more [here](https://www.kernel.org/doc/Documentation/kbuild/kconfig-language.txt).

https://www.kernel.org/doc/Documentation/kbuild/kconfig-language.txt

# How it works

Simplified schema:

* All configuration parameters described in KConfig-format file (including available ones)
* Interpreter (mconf, gconf, qconf, others) reads this file, current config (if it exists), and shows the menu
* The user selects needed options and selects the SAVE option
* KConfig moves old config to .config.old and creates new config (default name .config)

After that, you get a ready-made config that can be used in scripts(or for project build).

# Example project

Now, I’ll show you the demo-project. I placed it on [github](https://github.com/skif-web/demo-kbuild).

https://github.com/skif-web/demo-kbuild

# Overview

It’s a simple script that shows the information about the last logins. Available options:

* Select destination (file or console)
* If the selected destination is a file, the user can set the filename
* Shows information about both all and only current users

Just clone this repo and go to its directory:

boozlachu@debian:~$ git clone https://github.com/skif-web/demo-kbuild.gitCloning into 'demo-kbuild'...remote: Enumerating objects: 8, done.remote: Counting objects: 100% (8/8), done.remote: Compressing objects: 100% (6/6), done.remote: Total 8 (delta 0), reused 5 (delta 0), pack-reused 0Unpacking objects: 100% (8/8), done.boozlachu@debian:~$ cd demo-kbuild/boozlachu@debian:~/demo-kbuild$ lsgetLast.sh KConfig README.md

# KConfig interpreter notes

KConfig is just a language. You have to use the interpreter to show the menu. In the Linux kernel, the interpreter is built when you run the “make menuconfig” command. But I’m a lazy potato and will use the kconfig-frontends package. It is available in Debian/Ubuntu, and you can also build it from the source. Or add an interpreter build from source to your project.

# KConfig file

The main file here is KConfig. KConfig supports a lot of operators, but I’ll use only some of them.

Various types of variables are available there. The most useful are:

* string to save string data
* bool to save the bool value (y or commented string in config)
* tristate (\*/m/commented)

boozlachu@debian:~/demo-kbuild$ cat KConfigchoice SELECT\_DESTINATIONprompt "Select data destination"default DESTINATION\_TTYconfig SELECT\_DESTINATION\_TTYbool "show data in console"config SELECT\_DESTINATION\_FILEbool "save data to file"endchoiceconfig SELECT\_DESTINATION\_FILE\_FILENAMEstring "destination file"depends on SELECT\_DESTINATION\_FILEdefault "last.log"helpWrite destination file with relative or full pathconfig SHOW\_ONLY\_CURRENT\_USERbool "show data only for current user"default yhelpscript will get data only for current user

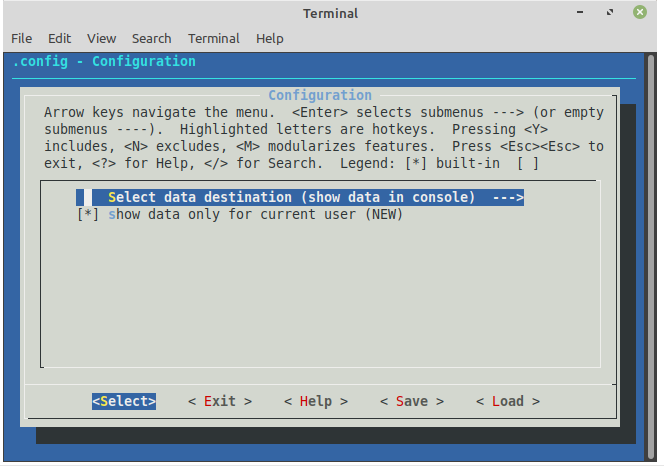
Let’s see it. Choice-endchoice pair allows creating SELECT of variants. Inside these keywords, I set definition (prompt), default value (default), and added available variants (SELECT\_DESTINATION\_TTY, SELECT\_DESTINATION\_FILE).

The next option (SELECT\_DESTINATION\_FILE\_FILENAME) will only be shown if the destination is the file. The keyword “depends on” allows us to do this. Also, the available keyword “selects” that works in the opposite direction.

Option SHOW\_ONLY\_CURRENT\_USER will always be shown.

Run kconfig in text mode:

boozlachu@debian:~/demo-kbuild$ kconfig-mconf KConfig



And with qt:

boozlachu@debian:~/demo-kbuild$ kconfig-qconf KConfig

# Test run

After configuring it, we get .config file with options:

boozlachu@debian:~/demo-kbuild$ cat .config## Automatically generated file; DO NOT EDIT.# Configuration#CONFIG\_SELECT\_DESTINATION\_TTY=y# CONFIG\_SELECT\_DESTINATION\_FILE is not setCONFIG\_SHOW\_ONLY\_CURRENT\_USER=y

As you can see, non-selected options are commented on. It’s useful for my script.

I run it with this config:

boozlachu@debian:~/demo-kbuild$ ./getLast.shboozlach pts/0 192.168.2.4 Sun Jul 19 06:00 still logged inboozlach pts/0 192.168.2.4 Sun Jul 19 05:58 - 05:58 (00:00)wtmp begins Thu Apr 16 05:48:31 2020

I changed the parameters via KConfig and got another result. Now the date was saved to a file, instead of being shown in the console:

boozlachu@debian:~/demo-kbuild$ cat .config## Automatically generated file; DO NOT EDIT.# Configuration## CONFIG\_SELECT\_DESTINATION\_TTY is not setCONFIG\_SELECT\_DESTINATION\_FILE=yCONFIG\_SELECT\_DESTINATION\_FILE\_FILENAME="last.log"CONFIG\_SHOW\_ONLY\_CURRENT\_USER=yboozlachu@debian:~/demo-kbuild$ ./getLast.shboozlachu@debian:~/demo-kbuild$ cat last.logboozlach pts/0 192.168.2.4 Sun Jul 19 06:00 still logged inboozlach pts/0 192.168.2.4 Sun Jul 19 05:58 - 05:58 (00:00)

# Conclusion

As you can see, KConfig is very powerful and friendly util for configuring. It can be used for scripts, package building, and other variants.

If you want to see an improved variant of KConfig using a makefile in the real project — write in comments.

Originally published at [*https://habr.com*](https://habr.com/ru/post/515398/).

