

From http://www.pcworld.com/businesscenter/blogs/linuxline/163019/10_expert_ubuntu_tricks.html

Recently I started work on a new [Ubuntu](#) tips book that will partner my existing title, *Ubuntu Kung Fu*. The new book is still being planned and won't be published until next year, but I thought I'd share 10 tips that are on my list to be included. If you have any others that you think would go into such a book, put them in the comments below. I'd love to hear them, as would other readers, I'm sure.

These tips aren't [Ubuntu-specific](#), but they've been tested on Ubuntu, and I can't guarantee they'll work on other distros. You might already know about some or all of them, but resist the temptation to gloat about this in the comments. You're just extra clever, OK?

With that said, let's start!

1. Open a Superuser Run Programs Dialog

You probably already know that hitting *Alt+F2* will cause a "Run Programs" dialog box to appear. Here you can type any program name to run it -- I often use it to run **gconf-editor**, which hasn't got a menu entry, for example. However, if you type **gksu** into a terminal window (with nothing following), a similar dialog box appears, but this time it will let you run the program as root (or any other user on the system). You can bind the **gksu** command to a keyboard shortcut (perhaps **Shift+Alt+F2**) using *System, Preferences, Keyboard Shortcuts*, to effectively produce a "superuser" Run Programs dialog.

2. Install Packages Quickly With GDebi

I spent a long time ignoring the GDebi program. To remind you, this is a GUI application that allows the installation of packages you've manually downloaded. It attempts to resolve dependencies using the repositories, which is a very useful feature. However, I've always preferred the good old-fashioned **dpkg** command. Then one day I tried GDebi at the command line and was shocked to find it also runs in a fuss-free non-GUI mode (i.e. **sudo gdebi package.deb**), yet still has the ability to resolve dependencies. Give it a try. Once you do, you may never want to use **dpkg -i** again.

3. CD Tricks

I'm sure that even bash experts occasionally slap themselves on the forehead and say,

"Wow! I never knew you could do that!". Recently I learned two tricks for the **cd** (change directory) command that had passed me by until now. Typing **cd** on its own will switch you back to your **/home** directory (i.e. the equivalent of **cd ~**). Typing **cd -** will switch you to the last directory you were browsing before you switched to the current one.

4. Add Directories To the Directory Stack With PUSH D

While we're on the subject of directories, let's talk about the directory stack. Quite simply, this is a list of directories stored by bash. The currently-browsed directory always tops the list, but the list is otherwise empty until you add a new directory, which you can do using the **pushd** command. For example, **pushd /usr/bin** will add **/usr/bin**. It will also switch you to that directory, so you could use it instead of **cd** to navigate (the **-n** command option will let you add a directory but stay where you are). The **dirs** command will show the directory list (remember that the currently-browsed directory will always be at the top). **popd** deletes the topmost entry in the list, and switches you to the next in the list. Give the directory list a try. It can be very useful if you're administering a system and have to leap from directory to directory, and are inclined to forget where important things are.

5. Kill Processes Quickly With PKILL

To kill a process I've always either used **top**, or the **kill** or **killall** commands along with **ps | aux** to discover process numbers/names. However, the **pkill** command removes a lot of work -- **pkill firefox**, for example, will search the list of processes for anything matching **firefox**, and then kill it (i.e. send a **SIGTERM**). **pstree** is also a pretty cool command, and will show all the processes in a family-tree arrangement, organized by who owns them.

6. Customize Gnome With Gnome Control Center

For those who have trouble giving up their Windows cravings, and who want a nice organized Control Panel-like experience, consider using **gnome-control-center** for your system configuration needs. Adding a desktop shortcut to this program can help newbies get to grips with Ubuntu, giving them a familiar Windows-like experience, and avoid forcing them to explore the scary System menu.

7. Launch OpenOffice.org Faster

If you use OpenOffice.org a lot, you might be frustrated at how long it takes to start each time. To get around this, open the Sessions program (called Startup Manager in **Jaunty**; whatever the case, click *System, Preferences, Startup Applications* to find it) and add a

new entry. In the Command field, type **openoffice -nodefault -nologo**. Then reboot. This will cause OpenOffice.org to be cached when the Gnome desktop starts up, so starting any OO.org application in the future will happen in a split second. Effectively, you're moving the OpenOffice.org startup delay to initial boot-up, but you'll hardly notice it.

8. Clear Up Disk Clutter In a Jiffy

Running short of disk space? Try typing **sudo apt-get autoremove** and then **sudo apt-get clean** into a terminal window. The first command removes any unused (redundant) dependencies from the system. The second removes all cached package files. Both are harmless. On a well-used system that's been updated a couple of times, you could free-up as much as a gigabyte using these methods. (Compare before and after using the **df -h** command.)

9. Figure Out Missing File Extensions

Been sent a file without an extension by e-mail? No idea what type of file it is? (Mac users are particularly guilty of the sin of considering file extensions optional.) Try the **file** command. Just specify the filename straight afterwards. Also, give the **strings** command a try. This will show any "printable strings" within a binary file (i.e. anything that isn't unprintable, which usually indicates data). The type of file is usually listed right at the top, so it's a good idea to pipe the output of **strings** into **head** (i.e. **strings filename|head**).

10. Don't Forget Your Man Pages

Check out these interesting but not-widely-read man pages: **intro** -- a beginner's guide to the command-line; **hier** -- a rundown of the filesystem hierarchy; **builtins** -- mini man pages for miscellaneous commands that don't have man pages of their own (including **pushd**, **popd** and **dirs**, as discussed above).

Keir Thomas is the award-winning author of several books on Ubuntu, including [Ubuntu Pocket Guide and Reference](#).