How To: Use ‘find’ to List .iso files in a Directory.

In this article, we’re going to do a real world exercise involving the ‘find’ command, because you really shouldn’t process the output of the ‘ls’ command.

Using The Linux Find Command:

The conventional wisdom is that processing the ‘ls’ command’s output is a Bad Idea®. Don’t take my word for it, read this link (which will save me the time of repeating it). Seriously, read it. It’s well worth the read.

If you absolutely refuse to read the above link, then the reason you shouldn’t parse the output from ‘ls’ is because there are file names that will screw up the output something fierce and you won’t get accurate output. On the other hand, if you take care with how you name your files then it really shouldn’t be much of an issue.

So, how did we get here?

Winter is ebbing and it was not a very spectacular season this year. As winter winds down, it means that it’s time for Spring Cleaning. I highly recommend folks Spring Clean their digital lives as well as their real lives. Come on now, you and I both know that you have files that can easily be deleted and save you some space. In the days of large-capacity storage, we’ve become digital hoarders.

Once in a while you should muck out the stalls. Which is, coincidentally, what I was doing when I decided that this would make a good article. I had close to 300 .iso files, most of which were for varied Linux distros and most of which were no longer current. Hell, some of them were for distros that don’t even exist anymore. (Let’s just say that it had been a

while since I cleaned out that directory.)

So, I went to town and cleaned out about 200 files. Some of those files had outlasted entire hard drives, being backed up, backed up again, and backed up some more – migrating like herds of reindeer across the steppes just above the taiga. Unlike the reindeer, these files had no significant value to me and it pained me exactly none to delete them.

Now, all the files are more or less sensibly named. I could easily have just used the ‘ls’ command, but it’s just bad form to do so. For example, I could have used:

[code]ls -la | grep iso[/code]

I could have output them to a handy text file for storage with:

[code]ls -la | grep iso >> linux_images.txt[/code]

I did not. No. No, I did not. Instead, I behaved myself and used the ‘find‘ command. Using ‘find’ is a bit more complex, but it’s worth learning how to use it.

The find command is a holdover from Unix. It first showed up in 1976 with Version 5. It was designed along with, and meant to be used in unison with ‘cpio‘, which oddly didn’t appear in Unix until Version 7. The cpio was all about archiving and actually still exists though you’ve probably never used it.

Anyhow, ‘find’ is defined as thus:

> *find – search for files in a directory hierarchy*

That bit about a directory hierarchy? That’s actually a bit of a sticking point with the ‘find’ command. It insists you include the directory in the command. You can’t just open the terminal, navigate to the desired directory, and run the command. You might just as well stay right there in your home

directory and run the command right there! Going anywhere ain't gonna help!

I wanted to list the .iso files in the directory and so I ended up with this command:

[code]find /media/kgiii/elements4/Distros -iname "*.iso"[/code]

Yes, yes I do have 4 WD Elements external drives connected to the PC where I do a bunch of my work. Don't you judge me! It's 2021! I'll do what I want!

Anyhow, of all the modifiers to that command, the important one for this exercise is the '-iname'. That is pretty handy and it means you don't have to worry about case sensitivity.

> *-iname*
> *Like -name, but the match is case insensitive. For example, the patterns `fo*' and `F??' match the file names `Foo', `FOO', `foo', `fOo', etc. The pattern `*foo*` will also match a file called '.foobar'*

The rest is self-explanatory. I obviously want anything that ends with .iso. The wildcard '*' indicates that I want any characters in front of the .iso to be included for the purposes of making my list.

NOTE: As you can see, I used quotes around the text I wanted to be found. That's mandatory. You have to quote 'em, else 'find' doesn't do its thing.

So, what can you do with this? You can count them, if you'd like. You would have learned that back when we were talking about 'awk' and 'putting it all together'. You just need to use a pipe and then 'wc -l'. So, that command would look like:

[code]find /media/kgiii/elements4/Distros -iname "*.iso" | wc -l[/code]

(If you're curious, I'm down to just 98 .iso files in that directory. That's a pretty impressive cleaning job, if I do say so myself!)

And, what else can you do with it? Anything you want! You tell me what else you can do with it. How else can you apply this? How do you use the 'find' command? How am I supposed to know what else you do with it?!? You get to decide how you use this information. Seriously, leave a comment letting me know how you can use the 'find' command in your daily computing.

Like always, thanks for reading. This article is a bit longer than it really needs to be. Still, you can sign up for the newsletter and get silly articles like this at least every other day. I write 'em ahead of time and use the scheduling feature that WordPress has. I'm not sure why, but this seems to make writing articles on a schedule easier. Maybe it takes the pressure off because it gives me at least two days to write an article? Dunno, I ain't that kinda doctor!