

## lab - oodiff

### Problem:

It is simple to compare two text files with a command like `diff`. But what do you do when you have two copies of a document in open document format (as made, for example, by LibreOffice). You want to see if the content (as opposed to the formatting) is different between the two documents, but how can you compare them?

### Solution:

OpenDocument Format (ODF) saves its files in a format that we can dismantle so that we can get at various parts of the document. We want to get at the content of the document (not the formatting). Create a simple document in LibreOffice (or use one provided by the instructor) and look at its contents by using the `unzip -l` command. (that's "minus el", for "list")

Now how would you write a shell script that takes two `.odt` files as command line arguments? The script should create two (2) scratch directories and in each directory unpack an `.odt` file, then compare the contents of each. Make sure those directory names are unique to your job and don't interfere with anyone else's directories - that is, but some variability in the name -- like `$$` if you know what that means.

But, wait. The content contains some mark up tags that aren't part of the document's content. Our script should edit that out, maybe with `sed`. To avoid leaving all the remaining text on one line, let's have the script replace any `<...>` tag with a newline. (Assume that the open and close of any tag doesn't break across line boundaries.)

Be sure to clean up after yourself -- that is, have a `trap` to be sure that the scratch directories get removed no matter how the script exits.

Do some error checking to be sure that two `.odt` filenames are provided on the command line.

I'm looking for clean, readable design. Group certain tasks into functions. Avoid repetitive code - use functions or at least more careful logic.