Traversing deep directory structures

If you have to switch between two directories with long paths, these two techniques can make life better.

(A Nest of) Symbolic Links

A symbolic link is a special kind of file which points at another file. Also known as symlinks. The symbolic link takes up almost no space. It is not a copy of the original file.

You can refer to the symbolic link and in most cases* the results will be the same as if you specified the original file. (*=some commands treat symbolic links in ways you might not expect. cp, rm, rsync, tar Their man pages will discuss how they treat symlinks.)

You create a symlink like this:

```
ln –s realfile newname
```

The resulting files when listed with `ls –l`:

```
lrwxr–xr–x  1 tunison  wheel  8 Jan 11 10:12 newname@ –> realfile
–rw–r––r––  1 tunison  wheel  0 Jan 11 10:11 realfile
```

So a way to make use of symlinks is to create a directory of them, and refer to those when doing things like changing directories. I use the following scheme myself. You can name your directory whatever you want, such as “redirect” instead of the shorter ”r”.

```
mkdir ~/r
cd ~/r
ln –s /cms01/data/puf–free/VERICRED/2019 v2019
ln –s /cms01/incoming/c–myjhed–98765 in
ln –s ~/mycode/thatlanguage/src/yes–i–wrote–it pride–n–joy
```

Now you can use the symlinks named “v2019”, “in” “pride-n-joy” instead of the longer real directory names.

Traversing deep directory structures (cont'd)

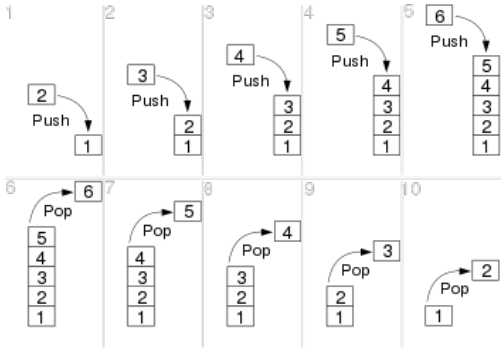
Symbolic Links (cont'd)

You can change directory with `cd ~/r/in` and you will wind up in `/cms01/incoming/c-myjhed-98765`

Or, You can copy downloaded source code into your home directory with
`cp ~/r/in/pkg-3.2.tar ~/mycode/thatlanguage/src/`

Pushd, Popd and Dirs Commands

If you're repeatedly working in several directories and don't need to open multiple windows to look at them simultaneously, these commands allow you to switch rapidly between them. They use a "stack" data structure. Think of a stack as a pile of plates in a cafeteria. When you want one, you usually take the one from the top. Then a fresh one is exposed. You've "popped" a plate from the stack. If you "push" three plates onto the stack, then the stack is deeper. You can access the top plate easily, but in this case you can also get at the third plate down.



`pushd directoryname`
changes your directory from the current one to directoryname, and creates a stack of two directories.

`dirs -v`
will list those directories in the stack.

`pushd`
by itself will switch you between the current directory and the top one in the stack. This is often the main way I use it.

`pushd +2`
will switch you between the current directory and the third down into the stack. (third because the index into the stack starts with 0 not 1)

`popd`
will cd back to the top directory in the stack while removing your current directory from the stack.

Traversing deep directory structures (cont'd)

Pushd, Popd and Dirs Commands (cont'd)

You can create a stack ahead of time using the –n option to pushd. That option adds the directory to the stack but does not change to it.

So if you used the nano text editor to add this to your .bashrc file, it would create a set of directories you can pushd between every time you log in!!!

```
# set up directory stack. Note that they appear in stack in reverse order than
listed here

# dirs –v will show their order (and index number)

for i in /cms01/data /cms01/outgoing /users/55548 /cms01/incoming; do pushd –n $i
1>/dev/null;done
```

Here we use that technique after adding it to our .bashrc file:

```
[~]$ source .bashrc

[~]$ dirs
~ /cms01/incoming /users/55548 /cms01/outgoing /cms01/data

[~]$ pushd +2
/users/55548 /cms01/outgoing /cms01/data ~ /cms01/incoming

[55548]$ pwd
/users/55548

[55548]$ dirs –v
 0  /users/55548
 1  /cms01/outgoing
 2  /cms01/data
 3  ~
 4  /cms01/incoming
```

Copying deep directory structures

The rsync command is much better than cp or mv

Rsync is a program which copies files from a source to a destination location. It has many available arguments but they aren't needed in most cases. Rsync's key utility is that it will compare the source and destination locations and only copy changed or missing files to the destination. If

A symbolic link is a special kind of file which points at another file. Also known as symlinks. The symbolic link takes up almost no space. It is not a copy of the original file.

You can refer to the symbolic link and in most cases* the results will be the same as if you specified the original file. (*=some commands treat symbolic links in ways you might not expect. cp, rm, rsync, tar Their man pages will discuss how they treat symlinks.)

```
mkdir ~/r
cd ~/r
ln –s /cms01/data/puf–free/VERICRED/2019 v2019
ln –s /cms01/incoming/c–myjhed–98765 in
ln –s ~/mycode/thatlanguage/src/yes–i–wrote–it pride–n–joy
```

Now you can use the symlinks named "v2019", "in" "pride-n-joy" instead of the longer real directory names.