

Traversing deep directory structures (cont'd)

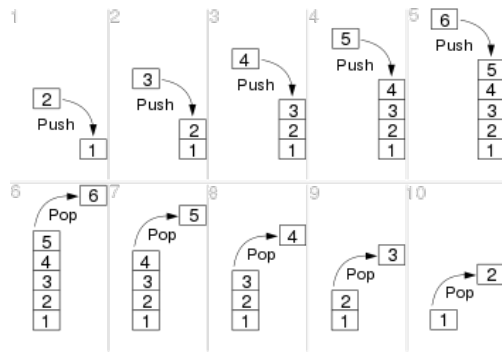
Symbolic Links (cont'd)

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
```

