copy-pasting directories using the shell

Imagine that you need to copy over some files or directories to another host, but the security policy or the connectivity doesn't allow to use standard file transfer protocols. Here's a quick and dirty solution to overcome such restrictions.

This little trick involves the ability to access two different hosts via SSH (or telnet, or even a serial console) and a terminal client supporting to capture screen output into a file or a copy-paste buffer.

So, on the source host, go for the file/directory, and **tar** it up, apply some compression to it as needed, send the output to stdout and pipe it directly through **base64**:

```
tar -czpvf - some/path | base64
```

This will give you some output and the base64-encoded representation of the data, which may look like this:

```
some/path
some/path/somefile1
some/path/somefile2
some/path/somefile3
some/path/somedir
some/path/somedir/someotherfile1
H4sIAGnAGVgAA+3POw7CQAwFwD1KbkC8kfY+kUIJK+XD+VklDVBAQ7qZ5lnyK+zp+rhs83if6i2d
pW9KKXs2n7nPkVsjIuccqY8hSqRuOO2iF9uyjnPXpbnW9Vvv1/74I46M+O+RAAAAAAAAAAA8O4J
FJs7gwAoAAA=
```

The "garbage" shown after the file and directory names is the base64 encoded contents from tar.
Don't bother decoding the output above, this is just some gargabe from /dev/urandom to illustrate this example ;-)

Now, copy-paste just the base64 output (or send it to a file, if your terminal client supports this).
Then, on the supposed-to-be target host, change to the directoy, where your files/directories should end up, then emter the command below:

```
cat|base64 -d|tar -xzpvf -
```

Don't worry, it'll "hang" on an empty line.
Now paste the buffer (or send the contents of the file captured before into the buffer).

This will looks similar to this:

```
cat|base64 -d|tar -xzpvf -
H4sIAGnAGVgAA+3POw7CQAwFwD1KbkC8kfY+kUIJK+XD+VklDVBAQ7qZ5lnyK+zp+rhs83if6i2d
pW9KKXs2n7nPkVsjIuccqY8hSqRuOO2iF9uyjnPXpbnW9Vvv1/74I46M+O+RAAAAAAAAAAA8O4J
FJs7gwAoAAA=
```

As soon as the buffer is flushed, output will string "hang", press CTRL-D to complete the transactions.

If done correctly, the input should be sent trough base64 to be decoded, and then passed on to tar to unpack. You should see the file and directory names accordingly.

```
cat|base64 -d|tar -xzpvf -
H4sIAGnAGVgAA+3POw7CQAwFwD1KbkC8kfY+kUIJK+XD+VklDVBAQ7qZ5lnyK+zp+rhs83if6i2d
pW9KKXs2n7nPkVsjIuccqY8hSqRuOO2iF9uyjnPXpbnW9Vvv1/74I46M+O+RAAAAAAAAAAA8O4J
FJs7gwAoAAA=
some/path
some/path/somefile1
some/path/somefile2
some/path/somefile3
some/path/somedir
some/path/somedir/someotherfile1
```

That's it, a while directory tree copied without involging file transer protocols.

Of course, the base64 encoding adds some overhead, so this doesn't work well for huge data loads as it's limited to the console speed. However this is a very quick solution if only a few files need to be copied quickly without bothering about possible restrictions.