

Re: Using os.system() and string concatenation

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2004-10/2071.html>

From: Donn Cave (donn_at_u.washington.edu)

Date: 10/12/04

Date: Tue, 12 Oct 2004 12:29:49 -0700

In article <mailman.4764.1097606614.5135.python-list@python.org>, Wayne Witzel III <wwitzel3@gmail.com> wrote:

> *Using Python 2.3*
>
> *Currently I process three files and build the output of those files in*
> *to lists using for statements.*
>
> *I think take those lists and provide them to an os.system() call.*
>
> *cmd = "/usr/sbin/useradd"*
> *os.system(cmd + list1[0] + list1[1] + list2[0] + list3[0])*
>
> *This executes without any errors, but doesn't execute the command*
> *supplied to os.system(). Now if I place them all in cmd first, then*
> *supply it to os.system() it executes just fine. So there really isn't*
> *a problem, just wanting to know what would cause such behavior.*

As already mentioned in other followups, could be white space missing.

On the other hand, this looks to me like a very good place to use `os.spawnv` instead of `os.system`. It might resolve the present problem, but much more importantly, it will avoid more dangerous problems of a similar nature. When you construct a shell command out of data from files, data becomes shell syntax, and there is in theory the possibility that the result will be worse than just invalid, it may execute a different command or different parameters than you had in mind, with a wide range of potentially unpleasant results.

But `spawnv()` executes the command directly, with the parameters you supply, unlike `system()` which invokes the shell. So with `spawnv()` you don't need white space, but more importantly the only command that can run as a result is the one you specify.

```
os.spawnv(os.P_WAIT, '/usr/sbin/useradd',  
          ['useradd', list1[0], list1[1], list2[0], list3[0]])
```

comp.lang.python: Re: Using os.system() and string concatenation

Donn Cave, donn@u.washington.edu