

Re: Problem using grep through the unix shell: doesn't accept a regular expression in filename

Source: <http://unix.derkeiler.com/Newsgroups/comp.unix.shell/2004-11/0315.html>

From: Stephane CHAZELAS (*this.address_at_is.invalid*)

Date: 11/09/04

Date: Tue, 9 Nov 2004 20:21:09 +0000

2004-11-9, 11:59(-08), Francois:

[...]

> #####

> #\bin\bash

Should be:

#!/bin/bash -

> *grep "mypattern" < \$1*

>

> #####

>

> *However it works perfectly when i execute it on the command prompt by*

> *replacing \$1 by *.txt for example, but when I call the script using*

>

> *myscript.sh *.txt*

>

> *The *.txt only takes into account the first file matching the regular*

> *expression and not the others! It seems quite weird, does anyone had a*

> *similar problem?*

[...]

It's normal.

In:

myscript.sh *.txt

The shell parsing that command line expands *.txt to the list of .txt files in the current directory and passes as many arguments to myscript.sh.

Moreover, the "<" operator accepts only one file (it opens a file for reading, and affects that to the standard input of grep, the standard input is one file descriptor, there is only one

comp.unix.shell: Re: Problem using grep through the unix shell: doesn't accept a regular expression in filename

standard input, so you can have only one file open. If you want the concatenation of the files, then use cat), so it's no use trying to pass several ones.

What you probably want to do is:

```
#!/bin/bash -  
exec grep 'mypattern' -- "$@"
```

"\$@" means the list of arguments verbatim (not just the first one).

Or, if you want grep to work on its standard input being the concatenation of the files:

```
#!/bin/bash -  
cat -- "$@" | grep 'mypattern'
```

(but beware you'll have troubles for files named '-')

(note that with both solutions, if the script is not given any argument, then the script (actually grep or cat) will read from its standard input).

--
Stephane