

## Re: \_\_del\_\_ pattern?

---

*Source:* <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2005-08/msg03011.html>

---

- *From:* "BranonZ" <[zarnovican@xxxxxxxxx](mailto:zarnovican@xxxxxxxxx)>
  - *Date:* 19 Aug 2005 10:18:39 -0700
- 

Bryan Olson wrote:

>> Use file that is writeable by A and B in a directory that is  
>> writeable only by root.  
>  
> Is that portable?

I have the feeling that you are asking if it works on Windows.  
No idea! I have only user experience with Windows.

On UNIX it is as portable as 'flock', which means all modern  
Unices (be careful about NFS).

> What's the sequence the program should try?

1.  
open a file, which name was previously agreed on  
(like /var/tmp/<prog-name>-<user-name>)

If it fails, report error and exit. System error or  
somebody has created unaccessible file by the same name.

2.  
Try to aquire a flock on the descriptor from step 1.

If it fails, some running process already has the lock, exit

3.  
lock will be released and lockfile closed automaticaly by OS  
on process exit.

```
import sys, fcntl
```

```
try:  
lockfile=open('/var/tmp/test1', 'w')  
fcntl.flock(lockfile.fileno(),  
fcntl.LOCK_EX | fcntl.LOCK_NB)  
except IOError:  
print sys.exc_info()[1]  
sys.exit(-1)
```

Re: \_\_del\_\_ pattern?

You can flock any open file, no matter if it is read/write/append.

BranoZ

---

• **References:**

- ◆ [del pattern?](#)
    - ◇ *From:* Chris Curvey
  - ◆ [Re: del pattern?](#)
    - ◇ *From:* bryanjugglercryptographer
  - ◆ [Re: del pattern?](#)
    - ◇ *From:* BranoZ
  - ◆ [Re: del pattern?](#)
    - ◇ *From:* Bryan Olson
- 
- Prev by Date: [Re: Idempotent XML processing](#)
  - Next by Date: [Re: global interpreter lock](#)
  - Previous by thread: [Re: del pattern?](#)
  - Next by thread: [Re: del pattern?](#)
  - Index(es):
    - ◆ [Date](#)
    - ◆ [Thread](#)

Re: \_\_del\_\_ pattern?