

Does Python File Writing Lock?

Duo Tao

California Institute of Technology

September 12, 2018

Writing Lock

Just in case you do not know: when you have two processes writing to the same file, typically you want them to write sequentially, not interleaving into each other. For example, if one process writes "Python", another writes "programming". You want to get "Python programming" instead of something like "Ppryogtrhoamminng". However, if the two processes run together, they write to the file together, so you will get something like that.

That is where locks come in: when one process writes to the file, it obtains the lock so that another program has to wait for the lock before writing.

Does python file writing lock?

Thus, the main question being answered here is that "when a Python processes write to a file, does it lock the file?"

Testing I wrote a piece of program writing 100 a's to a file, pausing 10 milliseconds for each.

```
import time
f = open('test.txt', 'a+')
for i in range(100):
    f.write("a")
    print("a")
    time.sleep(.010)
f.close()
```

I wrote another piece writing 100 b's to a file, pausing 10 milliseconds for each.

```
import time
f = open('test.txt', 'a+')
for i in range(100):
    f.write("b")
    print("b")
    time.sleep(.010)
f.close()
```

Results We run the programs with, starting both processes at the same time.

```
python test1.py & python test.py
```

The results are

- ▶ In terminal, a's and b's interleaving into each other.
- ▶ In test.txt, we have 100 a's and then 100 b's.

Conclusion

It seems safe to have multiple Python programs writing to the same file. The programs do not wait, according to the fact that the terminal printing interleaves. There might be some buffering mechanism inside that controls the file writing process to keep things in order. Interesting.