

## Re: Concurrency Issues

**Source:** <http://unix.derkeiler.com/Newsgroups/comp.unix.programmer/2004-05/0336.html>

---

**From:** Nick Landsberg (*hukolau\_at\_NOSPAM.att.net*)

**Date:** 05/12/04

Date: Wed, 12 May 2004 18:30:46 GMT

Barry Margolin wrote:

> In article <bc42e1a.0405120624.1a40b791@posting.google.com>,  
> bernardpace@yahoo.com (Xarky) wrote:  
>  
>  
>>Hi,  
>> I am writing a small program, where any number of users can have  
>>access to a file apparently concurrently. Since certain operations  
>>done on the file such as ADD at EOF, Modify, Delete, make rise to race  
>>conditions, I designed a small algorithm to ensure that only one user  
>>is using the file at one time. Basically I share a variable between  
>>all users and when a user requests an operation, user checks its value  
>>and upon its value, he acts.  
>  
>  
> Why don't you use file locks or semaphores?  
>  
>  
>> But by using the algorithm designed, I have lost the concurrency  
>>aspect. Is there a way to let users use the file apparently  
>>concurrently and avoiding the problem of race conditions?  
>  
>  
> Whenever concurrent applications use a shared resource, they have to  
> perform some kind of mutual exclusion to prevent races. There's no way  
> to get absolute concurrency, all you can do is minimize the size of the  
> critical regions where you access to the shared resource, which should  
> minimize the amount of waiting that takes place when two processes try  
> to access it.  
>  
> If you use file locking, the lock can be specific to a particular  
> section of the file. So process 1 can lock bytes 10-20, and process 2  
> can lock bytes 100-120, and they each can then modify their portion of  
> the file without interfering with each other?  
>

What Barry said is absolute