Operating Systems Problem session 2

Dibran Dokter 1047390 & Marnix Lukasse 1047400

September 27, 2019

6.10

To do this the process that tries to acquire the lock while the resource is not available will be blocked and placed in a queue. The process that holds the resource needs to signal that the resource is released and take the first process from the queue and continue it. To save the process to the queue it should save its PCB.

6.14

a) Number of processes can be read by multiple processes at the same time. This can cause a race condition.
Number of processes can be incremented and decremented at the same time because there is nothing keeping it from doing so. This would cause a race condition.

b) To prevent these race conditions the functions acquire and release should be placed around the increasing and decreasing of the number of processes.

c) Yes.

6.17

6.22

Fairness sees the writing and reading operation as being equal. In this case it would cause a writing operation to be executed only after all the readers are done reading. This would cause the writer processes to be starved. Throughput of operations tries to get all the operations done in the order they come in. And to get the best throughput it needs to stop the readers to let the writer get CPU time.

To solve the reader/writer problem without causing starvation we would need to ...

7.8

7.9

7.10

7.12