CSE 451: Section 3 Handout
10/10/2019

Processes and Synchronization

Synchronization:

When should you use a spinlock vs a sleeplock?

The sleeplock `sleepacquire` function will call `sleep` if the lock is not available.
If you hold a spinlock and then call `sleepacquire` while the sleeplock is in use, it ends with a system panic.
Why does the system panic in this case?

Note: writei/readi may put the process to sleep when writing/reading to the disk. You cannot hold a spinlock when calling these functions.

Condition Variables in xk use channels. `sleep` and `wakeup` both take a channel as an argument.
What is the purpose of a channel?
How are the channel arguments in these functions used with respect to the proc struct?

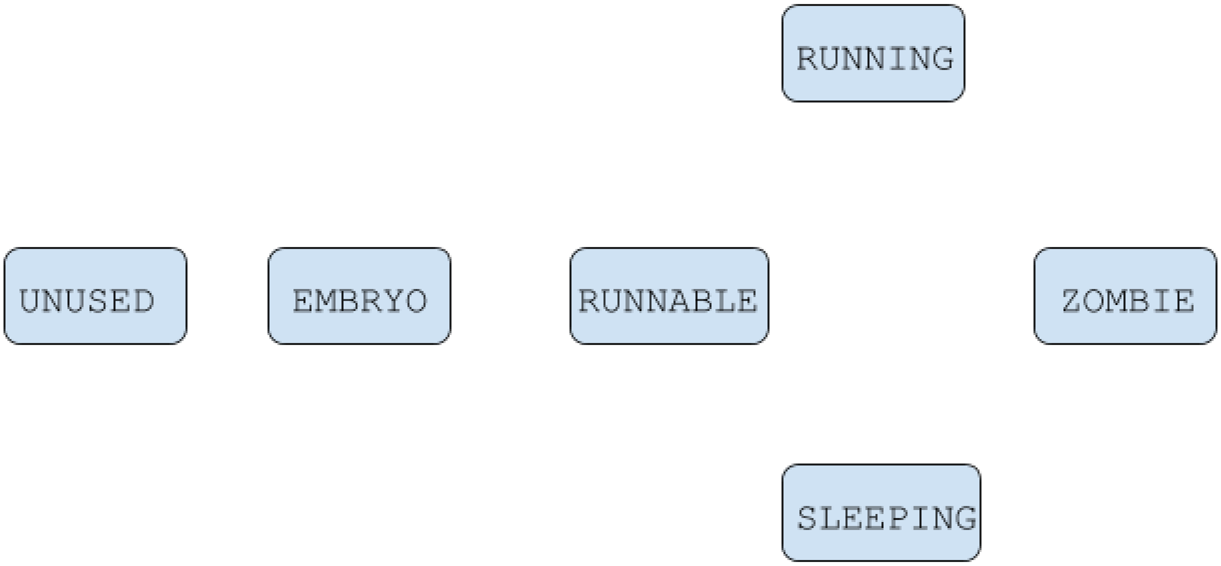
Wakeup is required to scan through the process table and change the processes sleeping on that channel to RUNNABLE.
What's a more efficient way to implement sleep/wakeup?

CSE 451: Section 3 Handout
10/10/2019

Processes and Synchronization

Processes:

Fill out the process state diagram below. Draw arrows from one state to another with the action that would result in that transition.



Why can a process not clear out it’s proc struct in `exit`?
If process A calls exit, who is responsible for clearing process A’s proc struct?

If a parent process calls `exit` before it’s child finishes executing, how does the child process need to be modified to guarantee that someone will wait for the child?