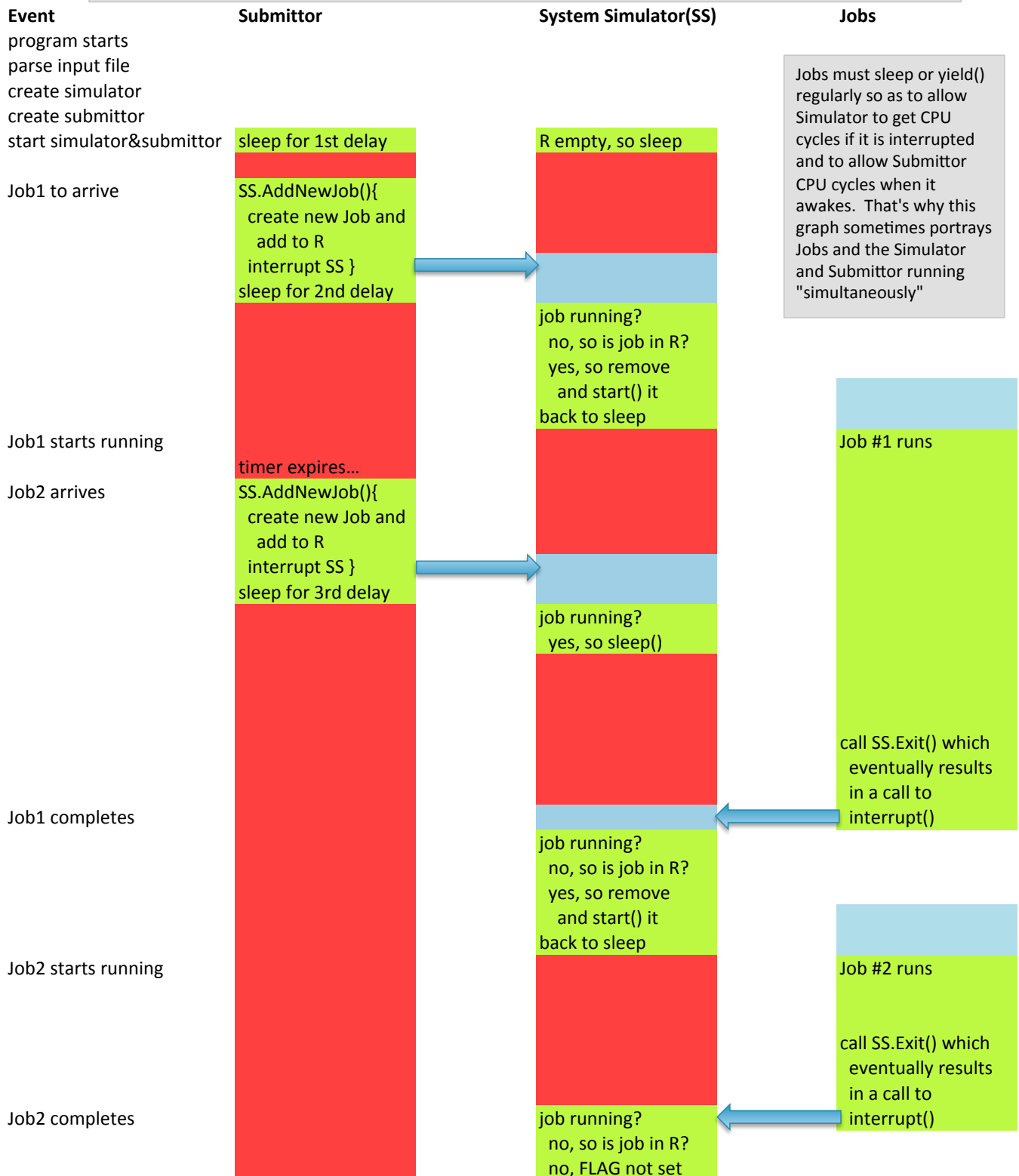This is a timeline showing the execution status of the threads comprising the Process Scheduling Programming Assignment. Green means a thread is running, red means sleeping, blue means runnable, but not yet selected bythe JVM to run.

Event	Submittor	System Simulator(SS)	Jobs
program starts			
parse input file			Jobs must sleep or yield() regularly so as to allow Simulator to get CPU cycles if it is interrupted and to allow Submittor CPU cycles when it awakes. That's why this graph sometimes portrays Jobs and the Simulator and Submittor running "simultaneously"
create simulator			
create submittor			
start simulator&submittor	sleep for 1st delay	R empty, so sleep	
Job1 to arrive	SS.AddNewJob(){ create new Job and add to R interrupt SS } sleep for 2nd delay	job running? no, so is job in R? yes, so remove and start() it back to sleep	
Job1 starts running	timer expires…		Job #1 runs
Job2 arrives	SS.AddNewJob(){ create new Job and add to R interrupt SS } sleep for 3rd delay	job running? yes, so sleep()	
Job1 completes		job running? no, so is job in R? yes, so remove and start() it back to sleep	call SS.Exit() which eventually results in a call to interrupt()
Job2 starts running			Job #2 runs
Job2 completes		job running? no, so is job in R? no, FLAG not set	call SS.Exit() which eventually results in a call to interrupt()

so….back to sleep

Job3 to arrive

timer expires

SS.AddNewJob(){
  create new Job and
    add to R
  interrupt SS }
sleep for 4th delay

job running?
  no, so is job in R?
  yes, so remove
    and start() it
back to sleep

Job3 starts running

Job #3 runs

timer expires

job4 to arrive

SS.AddNewJob(){
  create new Job and
    add to R
  interrupt SS }
no more jobs are to
arrive so set FLAG
and

(Submittor terminates)

terminate

job running?
  yes, so sleep()

call SS.Exit() which
  eventually results
  in a call to

job3 terminates

interrupt()

job running?
  no, so is job in R?
  yes, so remove
    and start() it
back to sleep

Job #4 runs

call SS.Exit() which
  eventually results
  in a call to

job4 terminates

interrupt()

job running?
  no, so is job in R?
  no, but FLAG is set,
  so this thread
terminates

(Simulator terminates)