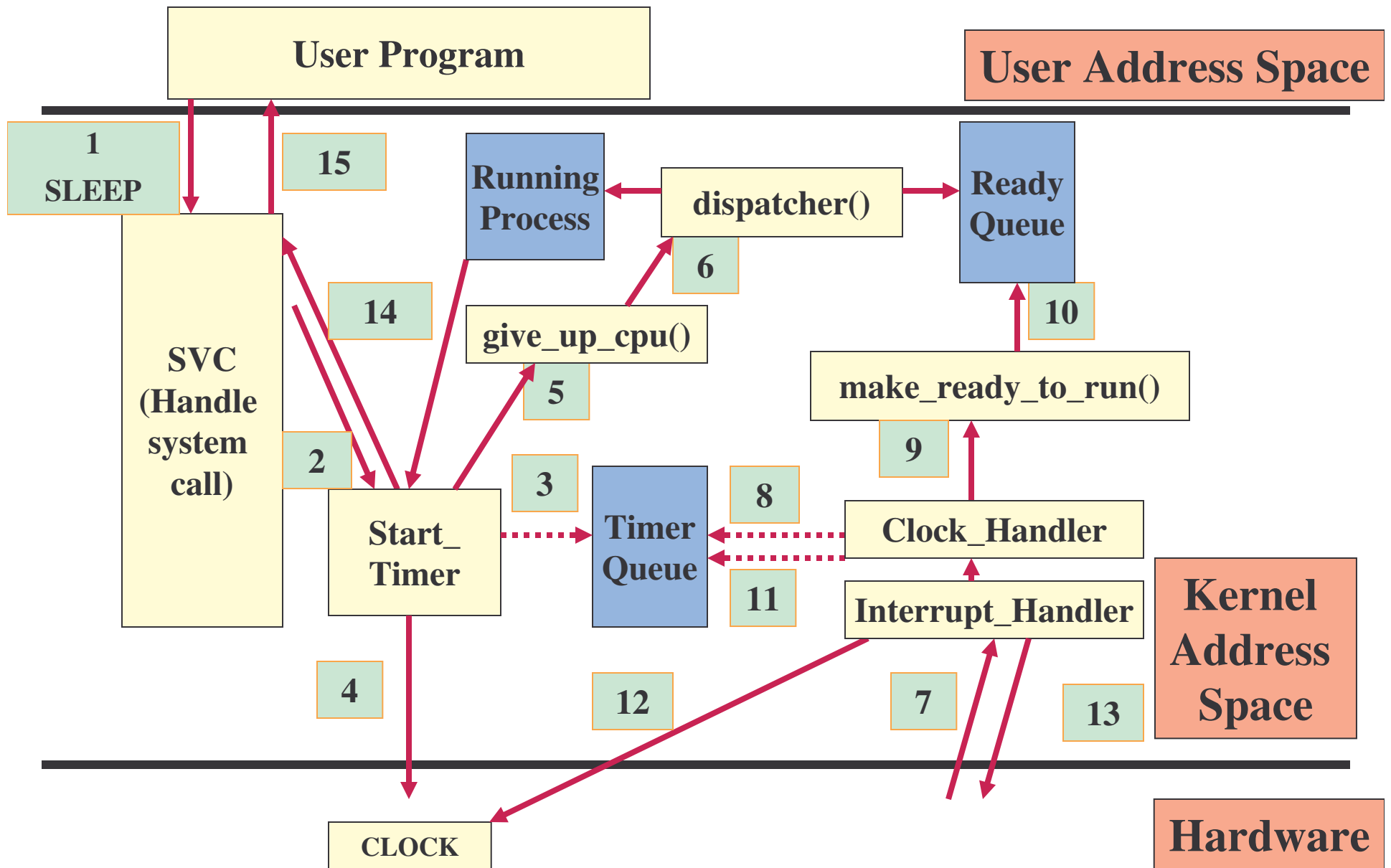


The Execution Of Sleep()



The Execution Of Sleep()

- Step 1:** The user level program does a sleep() system call.
- Step 2:** The system call traps to the kernel, ending up in a predetermined location that's equipped to handle system calls. The system call handler determines the request is a sleep and passes the request to a routine Start_Timer.
- Step 3:** The Start_Timer routine enques the PCB onto the timer queue.
- Step 4:** The Start_Timer routine invokes the hardware clock and asks it to interrupt at some time in the future.
- Step 5:** The process has nothing to do until the sleep completes, so it calls give_up_cpu().
- Step 6:** give_up_cpu() unloads the process from the processor and calls the dispatcher to find someone else to run. The dispatcher looks on the ready Q to find the next process to run. If there's noone to run, then it loops waiting for something to appear on the ready Q. If there is something there, run it.
- Step 7:** At some time, the hardware clock interrupts. The hardware interrupt routine determines that it's the clock that'd done the interrupt.
- Step 8:** The clock interrupt handler goes to the Timer Queue and extracts the PCB for the process that originally generated the interrupt.
- Step 9:** The clock handler takes that PCB and calls make_ready_to_run.
- Step 10:** make_ready_to_run puts that PCB on the ready queue in the appropriate order.
- Step 11:** The clock handler looks on the timer queue to see if some other process wants to use the timer.
- Step 12:** If yes, start the timer with the new request.
- Step 13:** The interrupt is now complete. The user process that was interrupted can now continue.
- Step 14:** At some point the dispatcher is called on to find the next process to run. The PCB of the process that was sleeping is taken off the Ready Queue and is loaded on the processor.
- Step 15:** The subroutine calls now unwind, going back through start timer, back through the system call handler, and finally back to the user process. It starts there at the line after the sleep() call.

Note the names used here are generic and are not meant to imply a particular routine on a particular OS.