

University of São Paulo  
Institute of Mathematical and Computer Sciences

SSC0140 - Operating Systems I  
Professor: Paulo Sérgio Lopes de Souza

**OER Documentation**  
**A12 Group**

Matheus de França Cabrini - N°USP: 8937375

Rita Raad - N°USP: 8061452

Rodrigo de Andrade Santos Weigert - N°USP: 8937503

## Description

The open educational resource (OER) “Sleep Wakeup Simulator” (SWS) simulates the producer/consumer scenario with sleep/wakeup as is presented in the book *Modern Operating Systems* (Tanenbaum, 2014, p. 129).

Such scenario consists of two processes and a fixed size buffer which is shared between them. One process, the producer, adds items to the buffer. The other, the consumer, removes them. The system calls *sleep* and *wakeup* are used for synchronization between the processes, preventing the violation of the problem’s semantic, for example, by a consumer “removing” an item from an empty buffer and thus causing an error. In this case, the consumer process would be blocked beforehand by a *sleep* call, and the producer would unblock it, when the buffer is no longer empty, with a *wakeup* call.

SWS allows the user to enter the role of a process scheduler and execute lines of the producer or consumer’s code, in whatever order desired. By doing that, the user gains the possibility of visualizing the execution of two processes running in pseudoparallelism in a multiprogrammed system, and also the *sleep* and *wakeup* semantics, its use for process synchronization and the possible race condition situations involved.

## Installation and Technical Details

The OER was developed for the web, being made purely with Javascript for the simulation, and HTML/CSS for the appearance and layout. Therefore, it does not require any installation. In order to be run, it only needs a browser that offers adequate support to the aforementioned languages - any recent browser should do so. Then, just access the OER through any of its many pages.

## How to Use

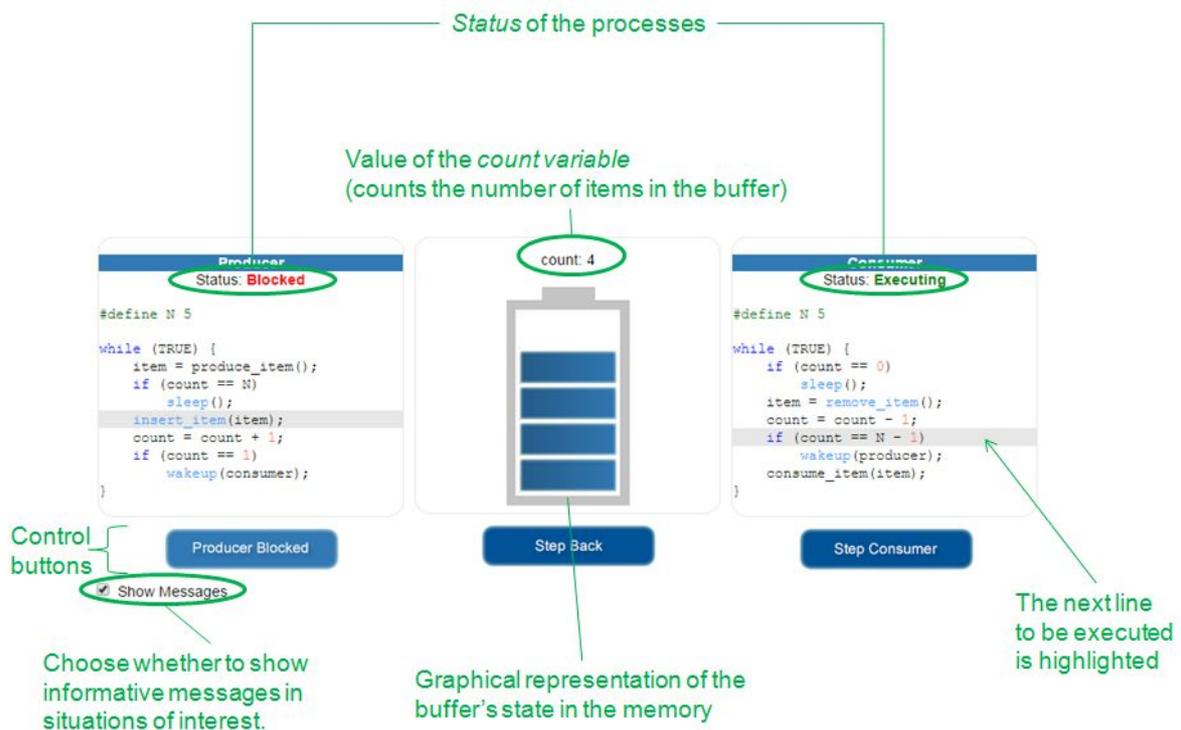


Figure 1: Simulation view, with its main elements explained

The simulation's view has three buttons. One of them relates to the producer, "Step Producer". When clicked, it executes the highlighted line of the producer's code - the highlight on a given line indicates that it is next to be executed. Another button, "Step Consumer", behaves in the same way, although for the consumer process. At last, the third button, "Step Back", undoes the last execution made.

The simulation's flow boils down to updating accordingly the buffer's state, the count variable's value - whose job is to keep track of the amount of items stored in the buffer -, the next lines to be executed and the consumer and producer processes' status, which can be "ready", "executing", or "blocked".

# OER's Screenshots



Figure 2: Simulation tab.

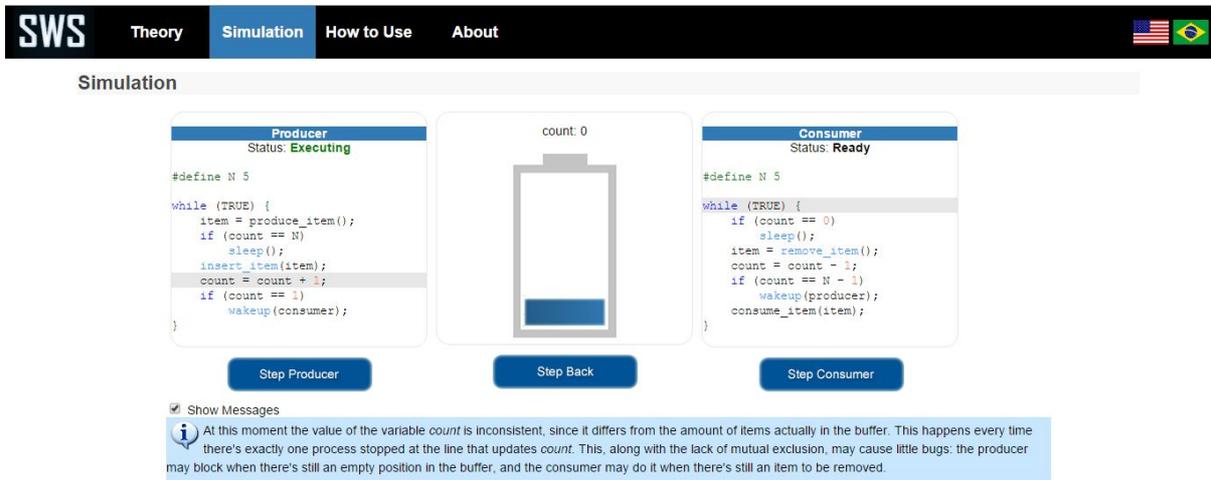


Figure 3: Informative messages (in blue background) are shown in appropriate situations.

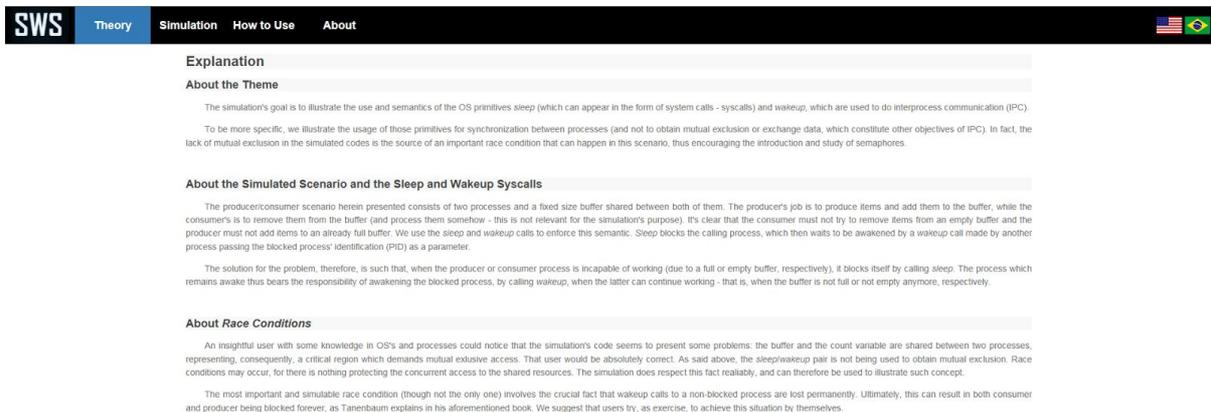


Image 4: Theory tab. Contains short explanations about the core concepts involved in the OER.

## About the Authors

Rodrigo Weigert - [rodrigo.weigert@usp.br](mailto:rodrigo.weigert@usp.br)



Rita Raad - [rita.raad@usp.br](mailto:rita.raad@usp.br)



Matheus Cabrini - [matheus.cabrini@usp.br](mailto:matheus.cabrini@usp.br)

