



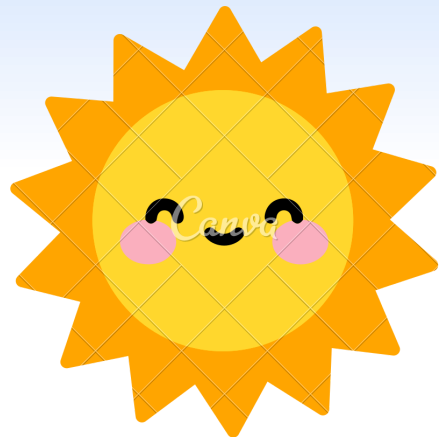
SLEEP



e



WAKEUP



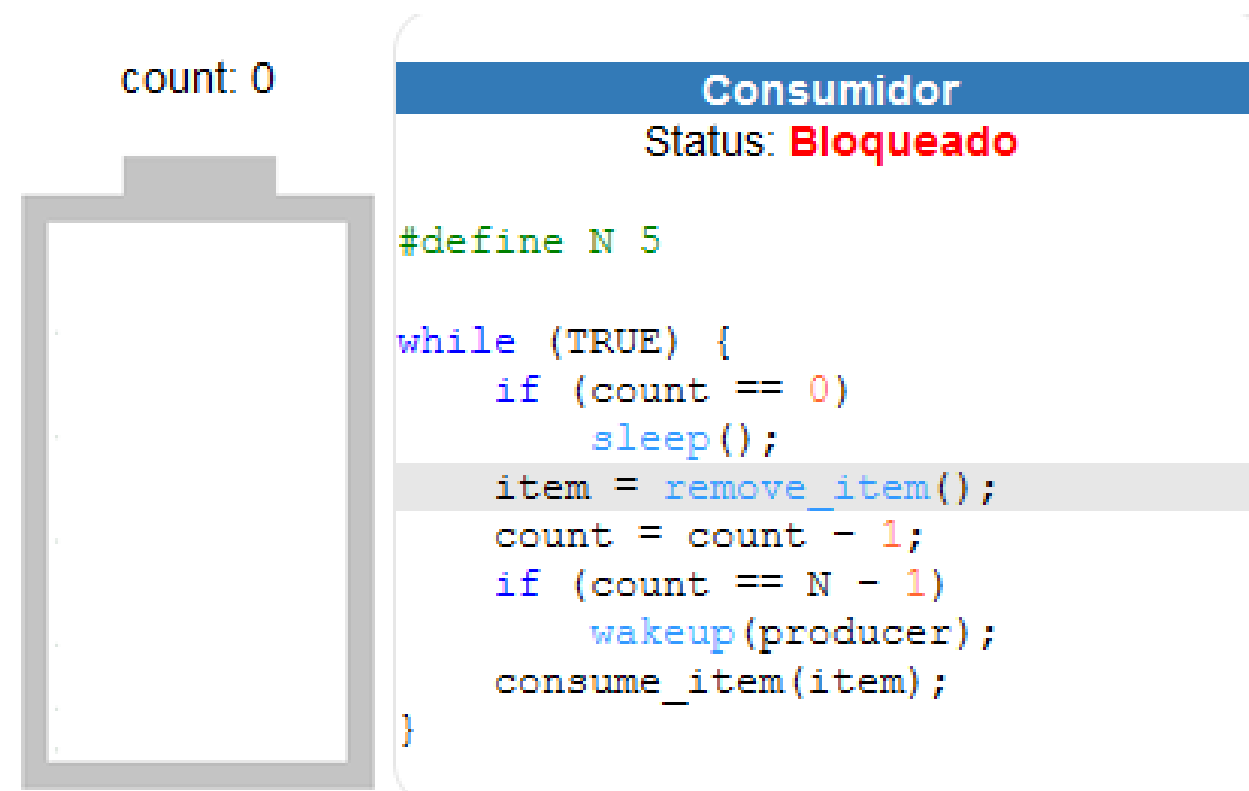
Sleep e Wakeup

- Primitivas de comunicação entre processos.
- Evitam a espera ocupada.
- **Abordagem** Bloquear a execução dos processos quando a eles não é permitido entrar em suas regiões críticas.
- **Sleep** Faz com que o processo que a chamou bloqueie, isto é, seja suspenso até que outro processo o desperte.
- **Wakeup**
 - Sinaliza (acorda) o processo anteriormente bloqueado pelo **Sleep**.
 - Tem como parâmetro o processo a ser desperto.

Produtor-consumidor com Sleep e Wakeup

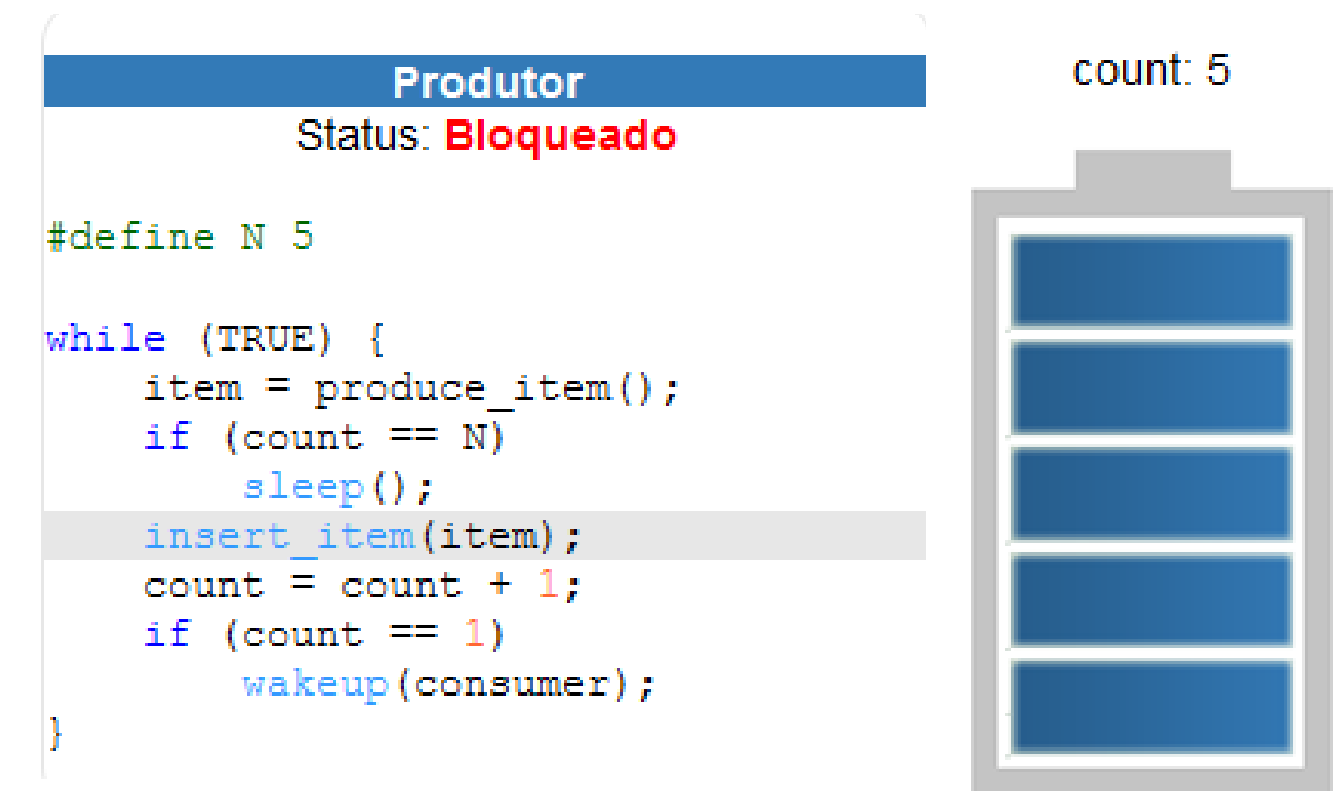
Consumidor

Dorme quando o buffer está vazio.



Produtor

Dorme quando o buffer está cheio.



O problema do Sleep e Wakeup

- ✧ **Condição de Corrida:** o acesso a *count* não é restrito.
- ✧ **Sinal Perdido:** o produtor tenta acordar o consumidor, mas este não está dormindo.
- ✧ Podem ocorrer situações em que ambos dormem para sempre.

Exemplo

Buffer: []
Count = 0

1. Consumidor verifica se o buffer está
vazio.

Buffer: []
Count = 0

Produtor

2. insere um item no buffer

3. incrementa count para 1.

Buffer: [X]
Count = 1

Consumidor:

Enquanto verdadeiro:

se count == 0:

dormir()

item = buffer.remove()

count = count - 1;

se (count == N - 1)

acordar_produtores()

consumir_item(item);

Produtor:

Enquanto verdadeiro:

item = produzir_item()

se count == N:

dormir()

buffer.adicionar(item)

count = count + 1;

se count == 1:

acordar_consumidor()

Exemplo

4. **Produtor** verifica se `count = 1` e tenta acordar o **consumidor**.

Buffer: [X]
Count = 1

5. O **consumidor** não está dormindo e o sinal é perdido.

6. Quando o **consumidor** volta a ser executado, executa *sleep* e é bloqueado.

7. **Produtor** ganha a CPU e, ao encher o buffer, irá dormir também.

Consumidor:

Enquanto verdadeiro:

se `count == 0`:

`dormir()`

`item = buffer.remove()`

`count = count - 1;`

se `(count == N - 1)`

`acordar_produtores()`

`consumir_item(item);`

Produtor:

Enquanto verdadeiro:

`item = produzir_item()`

se `count == N`:

`dormir()`

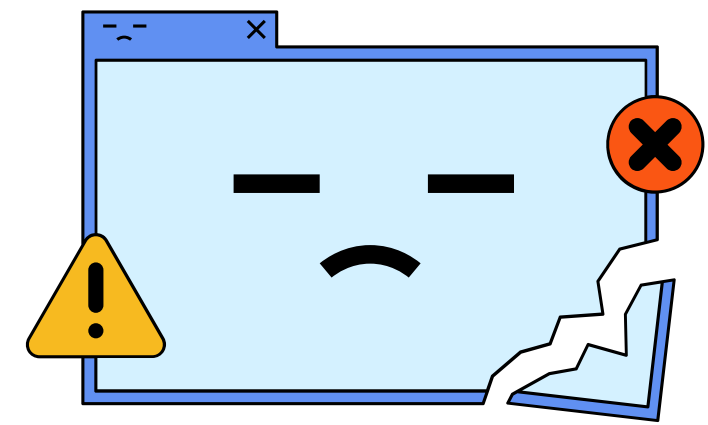
`buffer.adicionar(item)`

`count = count + 1;`

se `count == 1`:

`acordar_consumidor()`

Simulação Sleep e Wakeup



<http://lasdpc.icmc.usp.br/~ssc640/grad/bcc2015/grupoa12/simulation.html>



Simulação Sleep e Wakeup

NEW



<https://github.com/software-concorrente-distribuido/art-attack/tree/main/SleepWakeupSimulator>

Referência

TANENBAUM, A.; BOS, H. Sistemas Operacionais Modernos. 4ª Edição. São Paulo: Pearson Education do Brasil, 2016.