

RELATÓRIO DO SEGUNDO TRABALHO PRÁTICO PARA DISCIPLINA DE  
SISTEMAS OPERATIVOS

FUMADORES

30/12/2019

Vinícius Benite Ribeiro (82773)  
Professor: José Nuno Panelas Nunes Lau

## Introdução

O presente trabalho visa implementar na linguagem C um programa que resolve o problema dos fumadores, proposto em 1971 por Suhas Patil.

Existem 3 processos independentes: fumador, agente e o watcher. A sincronização entre eles é feita através de semáforos e memória partilhada.

## Smoker

Suponha que um cigarro exija três ingredientes para fumar: tabaco, papel e fósforos. Há três fumantes ao redor da mesa, cada um com um suprimento infinito de um dos três ingredientes - um fumante tem um suprimento infinito de tabaco, outro tem papel e o terceiro tem fósforos.

**waitForIngredients():** esta função fica a espera dos ingredientes dados pelo agent. Primeiramente, faz-se o down do semáforo para que o smoke fique bloqueado.

Dentro da zona crítica (somente uma operação pode ser executada ao mesmo tempo), o estado do smoker passa a WAITING\_2ING, ou seja, a espera do sinal do watcher que os ingredientes estão na mesa. O agent pode, então, começar a distribuir os ingredientes.

Ao receber o ingrediente que lhe falta, faz-se up no semáforo do smoker. Os smokes que já possuem o ingrediente na mesa continuam bloqueados.

Se ocorrer erro nos semáforos, o programa termina.

Se o agent sinalizar que a “fábrica” está fechando, o método waitForIngredients() retorna falso. Caso contrário, retorna verdadeiro.

**rollingCigarette():** quando o smoker recebe do watcher o sinal que pode fumar (informSmoker()), o semáforo é bloqueado e, na região crítica, o estado do smoker passa para ROLLING. Temos um ciclo for para simular o uso de cada ingrediente (ingredients[]) e salvamos o estado do mesmo. Temos um sleep para esperar o smoker “enrolar o cigarro”. O semáforo é desbloqueado, avisando o agent que o cigarro está pronto e que ele pode produzir novos ingredientes.

**smoke():** o semáforo é bloqueado novamente, o estado do smoker passa a SMOKING. aumentamos o counter de número de cigarros do smoker, salva-se o estado e, por fim, desbloqueia-se o semáforo novamente.

## **Agent**

Um agente permite que os fumantes façam seus cigarros selecionando arbitrariamente dois dos suprimentos para colocar na mesa. O fumante que tem o terceiro suprimento deve remover os dois itens da mesa, usando-os (junto com o próprio suprimento) para fazer um cigarro, que eles fumam por um tempo..

**prepareIngredients():** o agente prepara dois ingredientes aleatoriamente. Coloca-se o semáforo a down. Na zona crítica, muda-se o estado do agent para PREPARING. Em seguida, gera-se ids para os ingredientes. Os semáforos são colocados a up para sinalizar o watcher que os ingredientes foram colocados a mesa. Temos semáforos diferentes para ingredientes diferentes.

**waitForCigarette():** faz-se o down do semáforo do agente, pois o mesmo está a espera do smoker enrolar o cigarro. Quando o smoker envia o “sinal positivo”, o semáforo waitCigarette vai a up.

**closeFactory():** aqui o agente termina a produção de ingredientes. Na zona crítica, o agentStat vai a CLOSING\_A e a variavel closing passa a true. Em seguida, todos os semáforos dos ingredientes, nIngredients, vai a up, para notificar o watcher que o agent está a fechar a producao.

## **Watcher**

O watcher é responsável por verificar se após a emissão de um novo ingrediente (um dos elementos do pacote produzido pelo agente) há algum fumador que possa fumar. Existe um watcher por cada tipo de ingrediente.

**waitForIngredient():** o watcher fica a espera do agent. O semáforo é bloqueado e o estado do watcher passa a WAITING\_ING. O semáforo de cada ingrediente é bloqueado, se ele estiver na mesa.

O watcher bloqueia o semáforo novamente, entra na zona crítica e verifica se o agent está a fechar, para, enfim, notificar os smokers. A função, neste caso, retorna falso.

**updateReservations():** este método retorna o id do smoker que possui todos os 3 ingredientes e está pronto para fumar. O semáforo é bloqueado, entra-se na zona crítica, o estado do watcher vai a UPDATING e ocorre a checagem se o ingrediente está reservado e se o smoker tem condições de fumar. Por fim, o semáforo é desbloqueado e a função retorna -1, caso nenhum smoker esteja apto a fumar.

**informSmoker():** o estado do watcher passa a INFORMING, na região crítica. Em seguida, o mesmo passa o semáforo a up e informa os smokers se eles podem começar a enrolar o cigarro.

## Fontes

Guiões e materiais das aulas P.

<https://pages.mtu.edu/~shene/NSF-3/e-Book/SEMA/TM-example-smoker.html>

[https://en.wikipedia.org/wiki/Cigarette\\_smokers\\_problem](https://en.wikipedia.org/wiki/Cigarette_smokers_problem)