**Sistemas Operativos**

Prof. Nuno Lau



**RELATÓRIO – Trabalho 2**

**Restaurante**

*David Filipe Ferreira Amorim, nº 112610, P5*

*Francisca Ferreira Pedro Silva, nº 112841, P5*

*Licenciatura em Engenharia Informática*

*Aveiro*

*2023/2024*

Índice

[Introdução 1](#_Toc154878700)

[1 – Processos 2](#_Toc154878701)

[1.1 – Grupos (Groups) 2](#_Toc154878702)

[1.2 – Rececionista (Receptionist) 5](#_Toc154878703)

[1.3 – Empregado de mesa (Waiter) 7](#_Toc154878704)

[1.4 – Chef 9](#_Toc154878705)

[2 – Testagem 10](#_Toc154878706)

[Conclusão 13](#_Toc154878707)

# Introdução

Enquanto estudantes da Licenciatura em Engenharia Informática na Universidade de Aveiro, no âmbito da cadeira Sistemas Operativos, tivemos como objetivo deste trabalho a compreensão dos mecanismos associados à execução e sincronização de processos e threads.

Deste modo, ao longo do presente relatório iremos explicar detalhadamente a abordagem que tomamos para a realização/resolução do problema e testagem para validação da solução que compõem o trabalho.

# 1 – Processos

Aquando da simulação deste restaurante, os grupos, o rececionista, o empregado de mesa e o cozinheiro são processos independentes, pelo que a sua sincronização é feita através de semáforos e de memória partilhada.



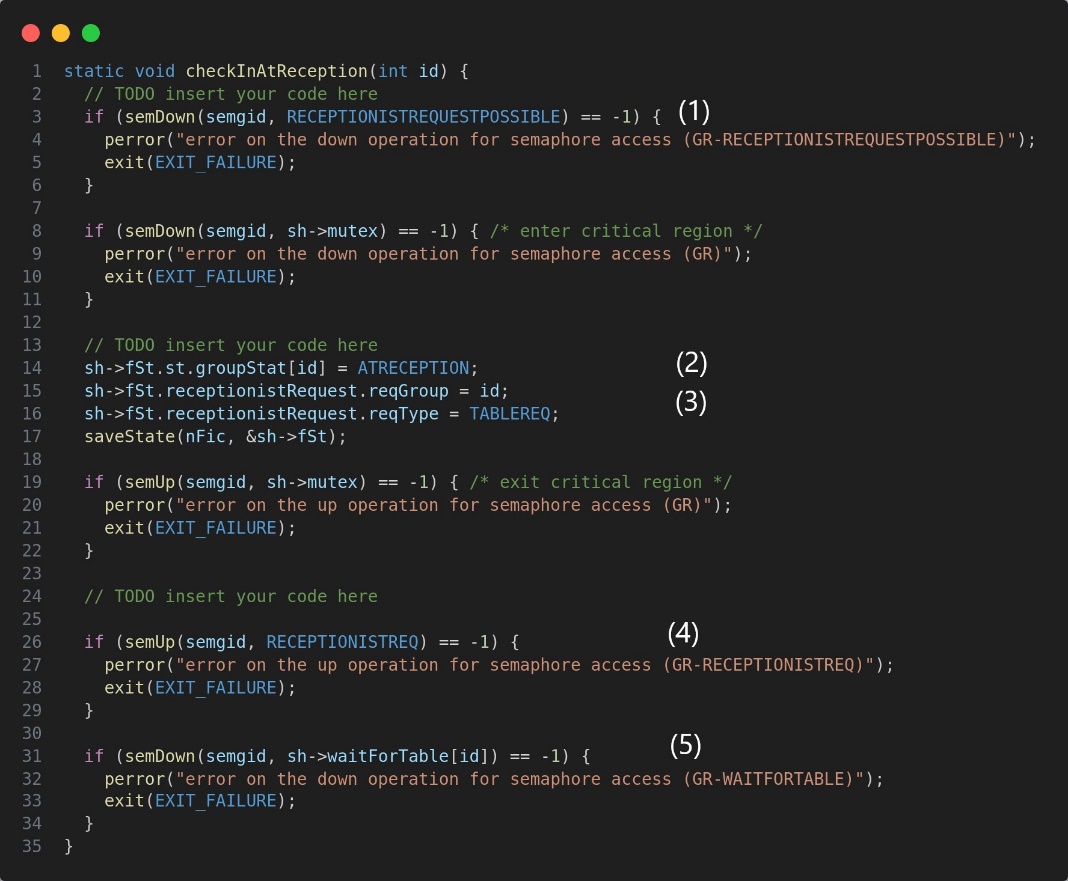
## 1.1 – Grupos (Groups)

Este processo é constituído por seis operações representadas pelas funções:

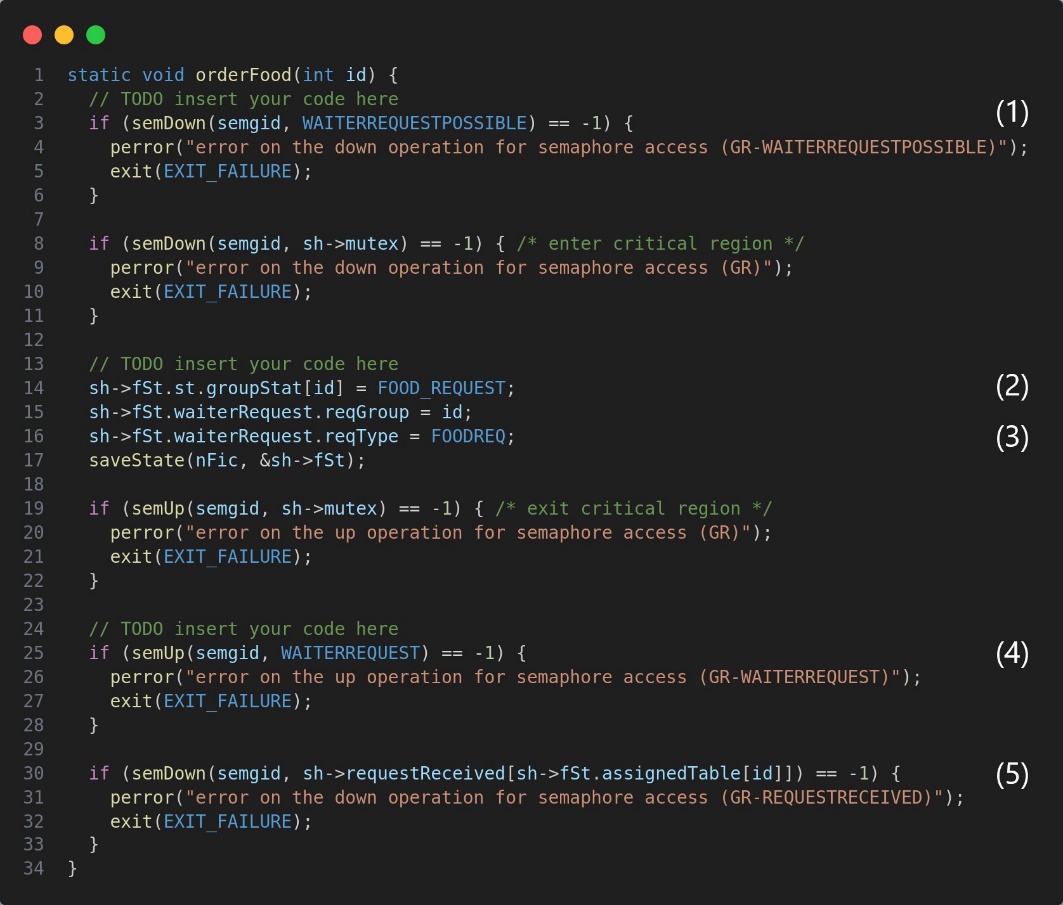
* goToRestaurant;
* checkInAtReception;
* orderFood;
* waitFood;
* eat;
* checkOutAtReception.

Neste trabalho foi solicitada a alteração de quatro destas funções.

Na função checkInAtReception o grupo chega ao Restaurante onde pretende jantar e dirige-se ao rececionista, aguarda pela disponibilidade do mesmo (1), altera o estado para ATRECEPTION (2) e solicita uma mesa (3), o rececionista processa o pedido (4), enquanto o grupo aguarda até que se encontre alguma mesa disponível (5).

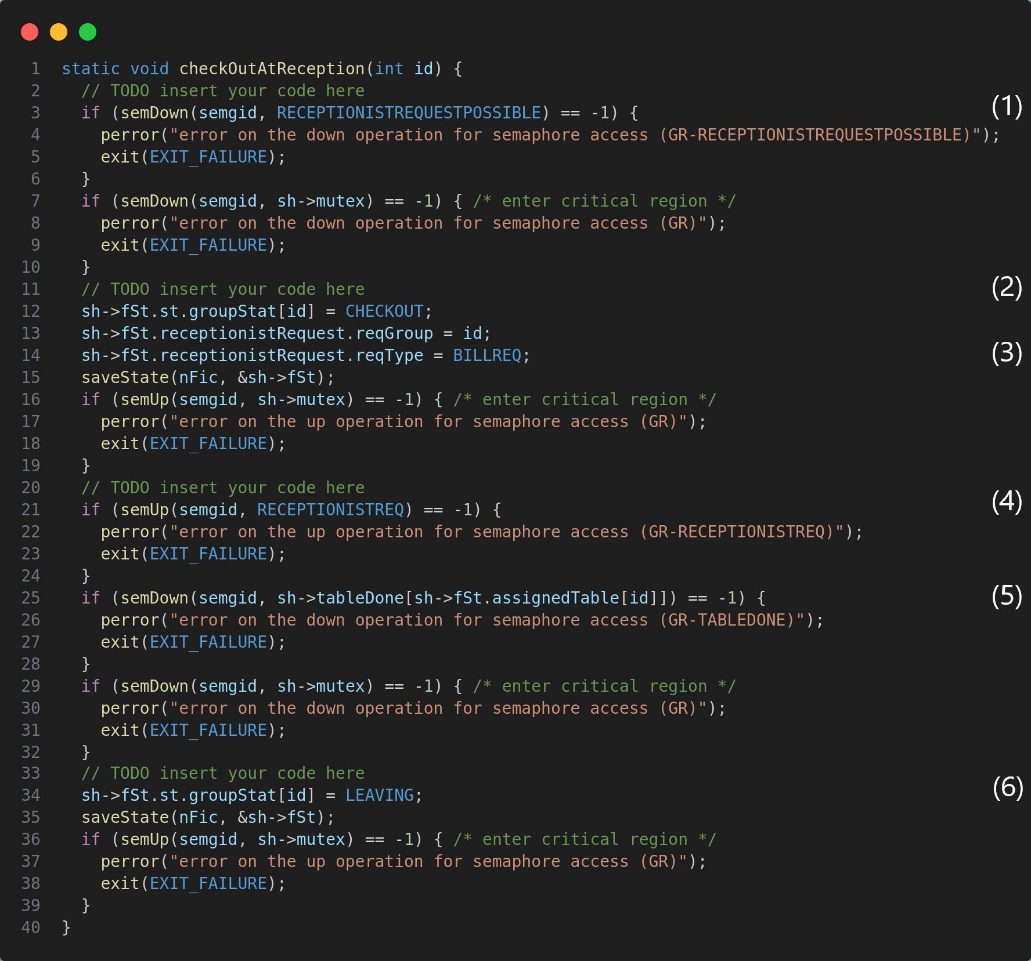


Na função orderFood o grupo aguarda pela disponibilidade do empregado de mesa (1), altera o estado (2), faz um pedido (3) e aguarda que o pedido seja recebido(4).



Na função waitFood altera o estado para WAIT\_FOR\_FOOD (1) e aguarda que a comida chega (2), após a chegada da mesma come e atualiza o seu estado para EAT (3).



Na função checkOutAtReception o grupo aguarda pela disponibilidade do mesmo (1), altera o estado para CHECKOUT (2) e solicita a conta (3), o rececionista processa o pedido (4), o grupo aguarda a conclusão do checkout (5), sai e atualiza o seu estado para LEAVING (6).

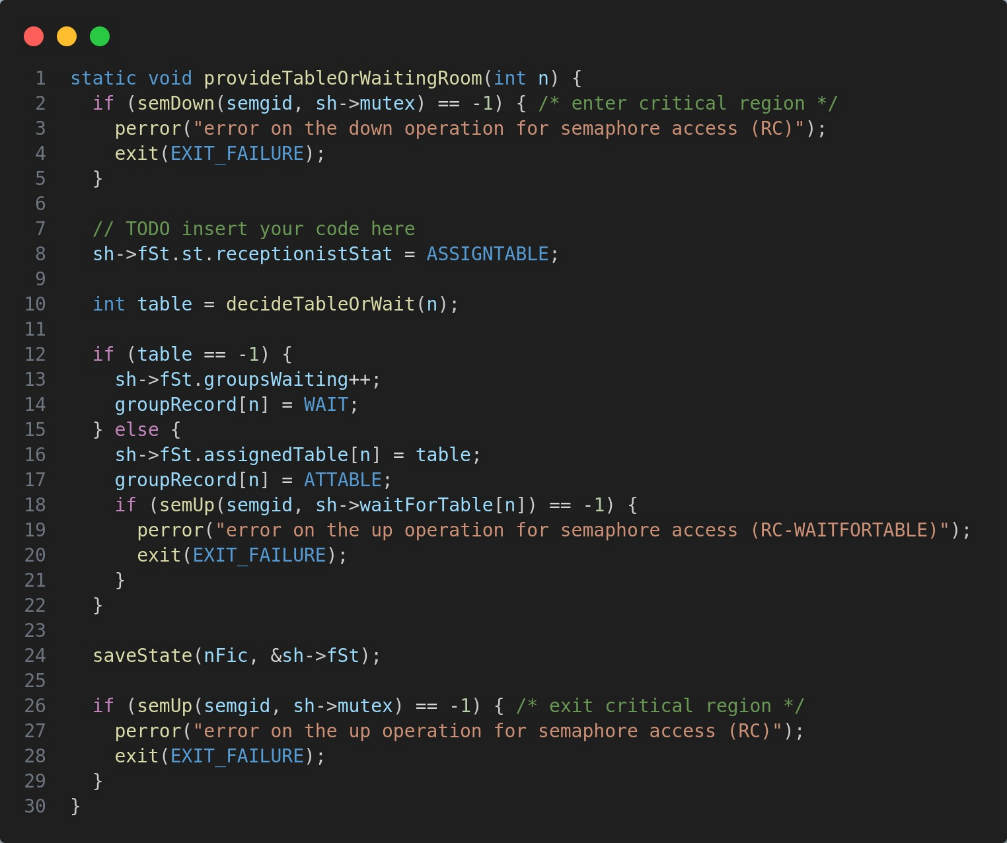
## 1.2 – Rececionista (Receptionist)

O rececionista executa a operação waitForGroup, provideTableOrWaitingRoom e receivePayment.

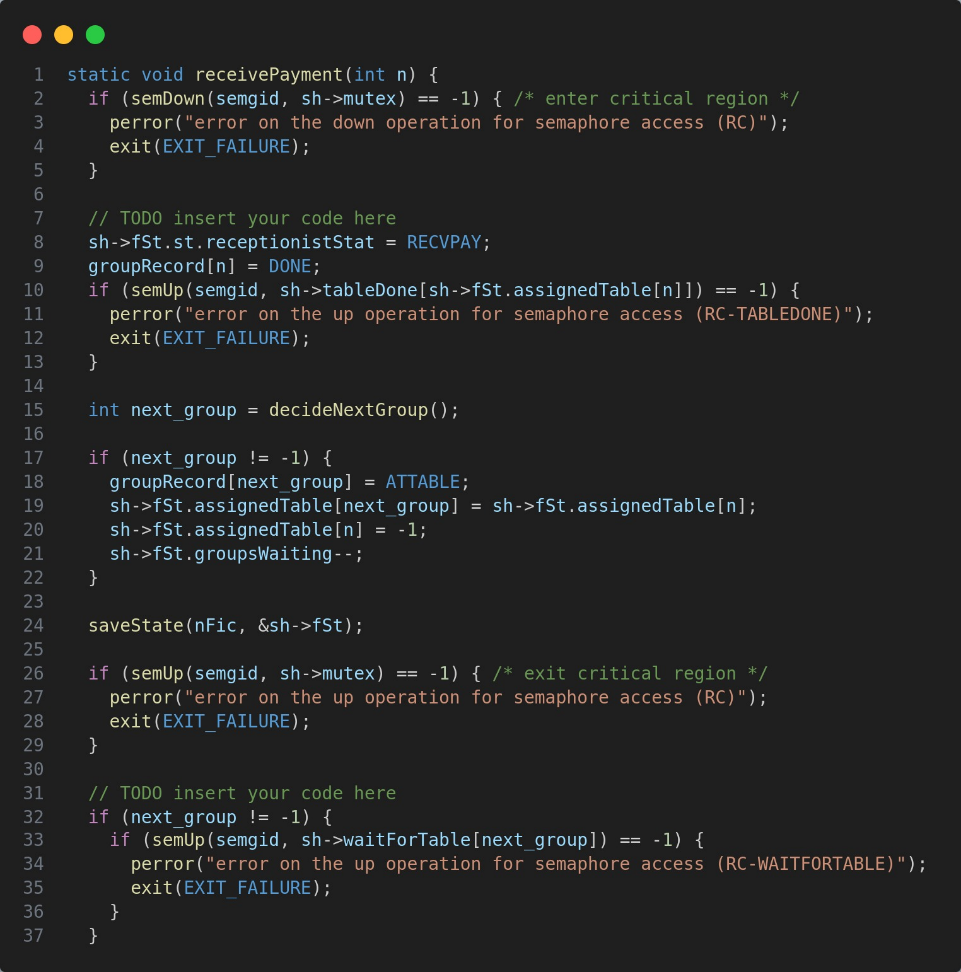
A função waitForGroup atualiza o estado da função para WAIT\_FOR\_REQUEST, e lê o pedido e diz se é possível realizar um novo pedido.



Por sua vez, a função provideTableOrWaitingRoom, à semelhança da função anterior, atualiza o estado da função para ASSIGNTABLE e decide qual a localização para onde o grupo se deverá dirigir. Esta decisão recorre à função decideTableOrWait e caso a mesa esteja disponível o rececionista deverá indicá-lo.



Aquando do pagamento, a função receivePayment atualiza o estado para RECVPAY, recebe o pagamento e indica que o pagamento foi realizado com sucesso e, caso existam grupos à espera indica que há uma mesa disponível.



## 1.3 – Empregado de mesa (Waiter)

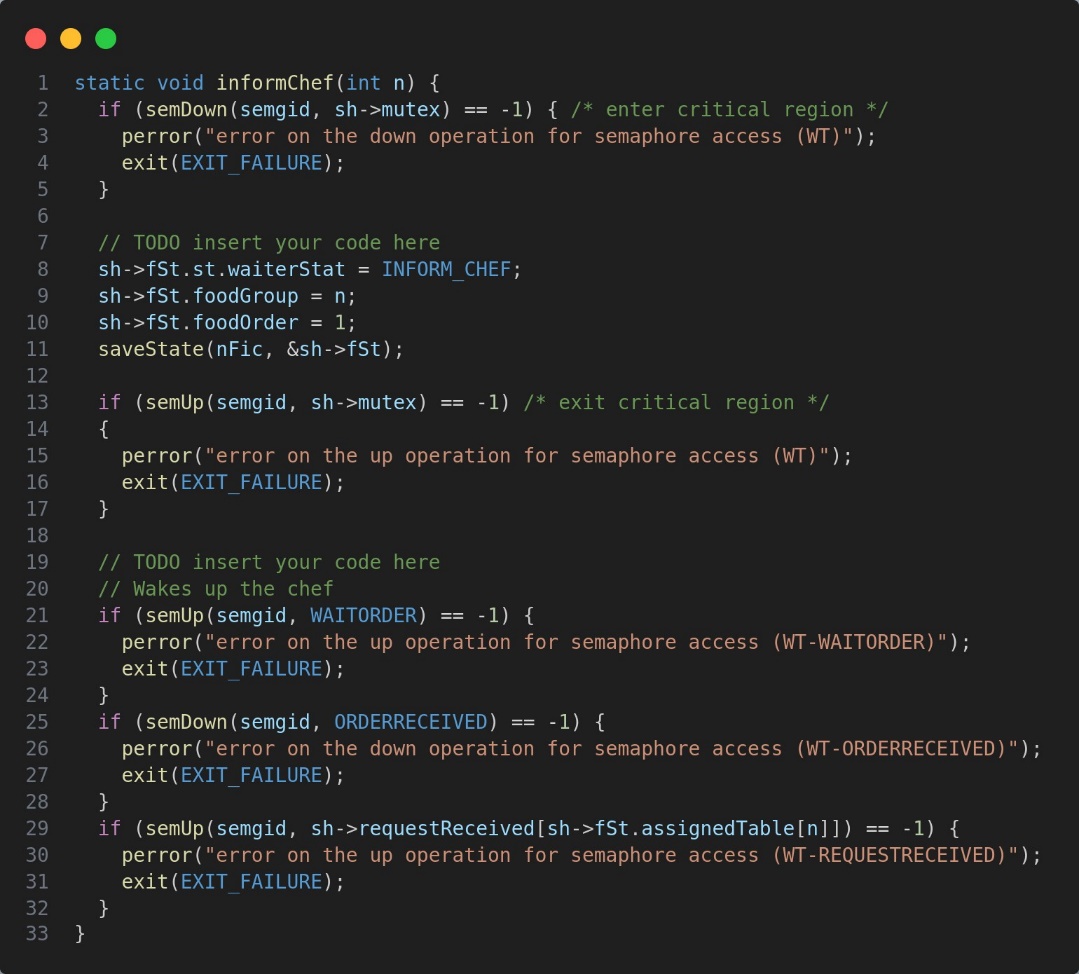
Este processo é constituído por duas operações representadas pelas funções:

* waitForClientOrChef
* informChef
* takeFoodToTable

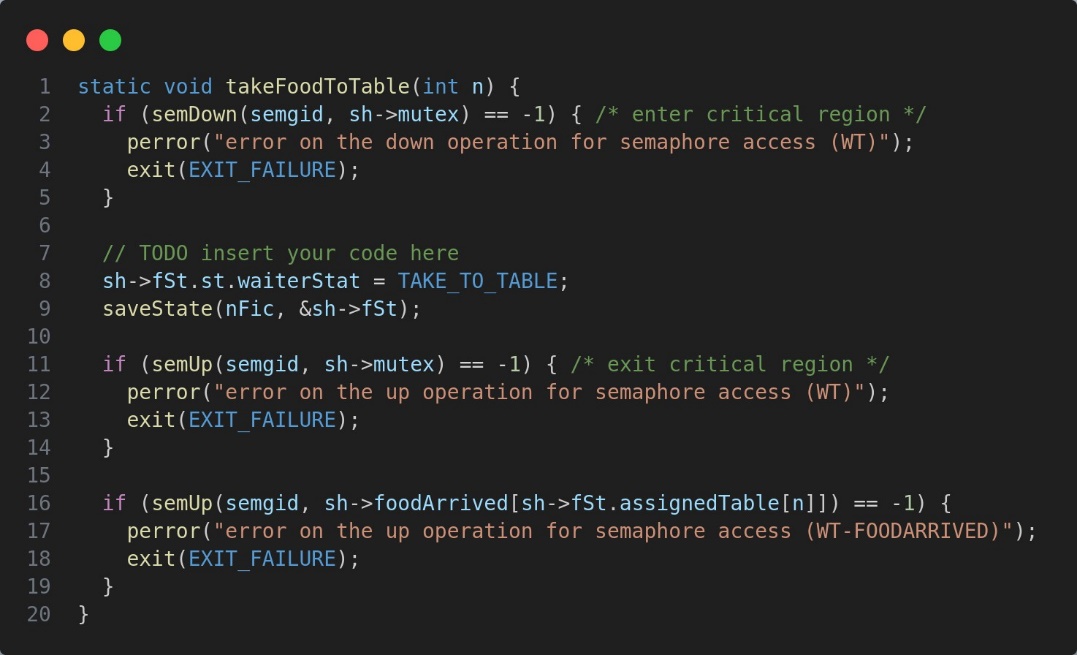
A primeira função é muito semelhante à função waitForGroup do processo Rececionista.



A segunda função atualiza o estado para INFORM\_CHEF, entrega um pedido ao chef, o empregado aguardar que o chef confirme a receção do pedido e verifica que o chef está a tratar do pedido.

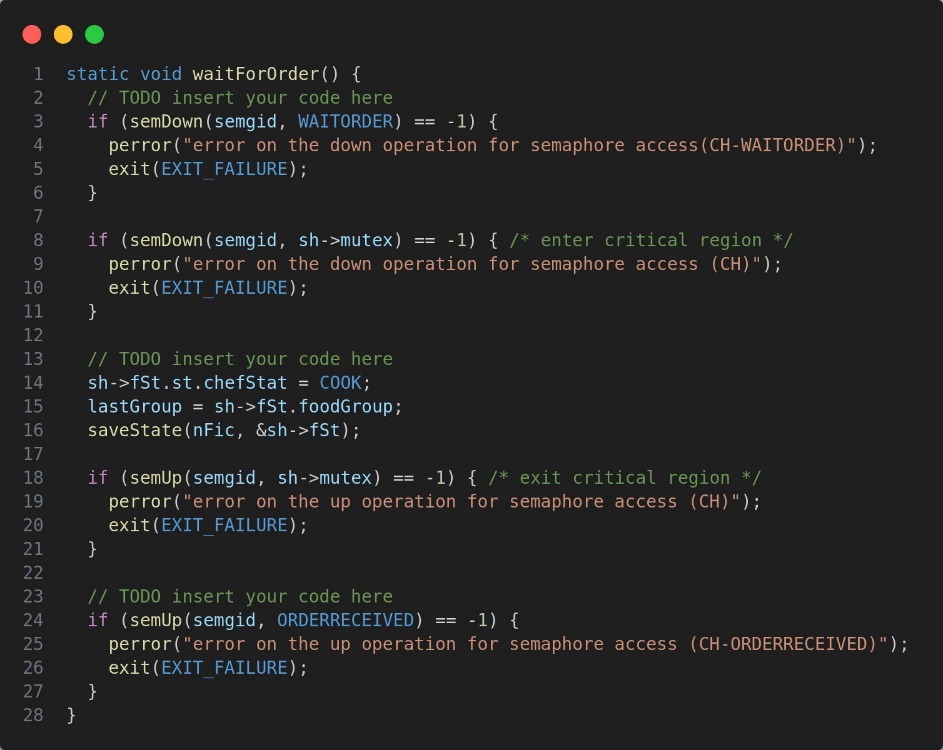


Na terceira função, o estado da função é atualizado para TAKE\_TO\_TABLE, e o empregado de mesa entrega a comida ao grupo.

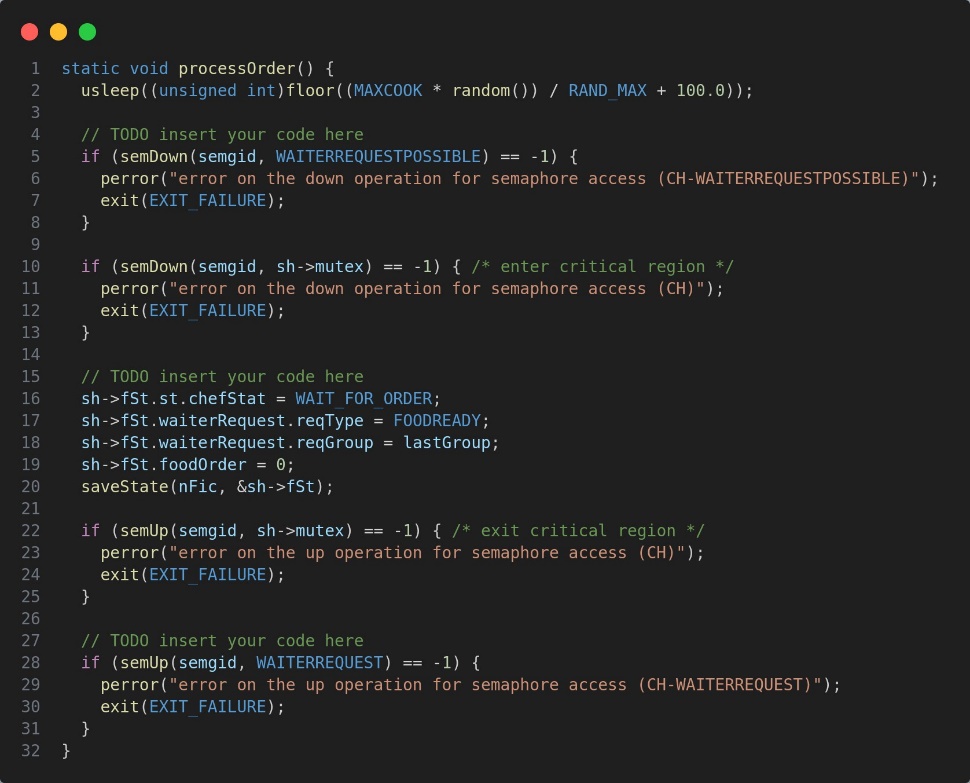


## 1.4 – Chef

Por um lado, o chef, na função waitForOrder, espera que cheguem pedidos e, quando há, atualiza o seu estado para COOK, confirma a receção do pedido para que o empregado de mesa tenha esse conhecimento e o passe ao grupo.



Por outro lado, na função processOrder, o chef processa o pedido e aguarda que o empregado de mesa esteja disponível, atualiza o seu estado para WAIT\_FOR\_ORDER e indica que a comida está pronta.



# 2 – Testagem

De modo a confirmar se os resultados que obtivemos estavam de acordo com o espectável, verificámos se existia algum deadlock, pelo que constatámos que eram inexistentes e analisámos vários resultados de várias execuções, como mostra o seguinte exemplo:





A fim de verificar se existia algum deadlock, com o auxílio do script run.sh, corremos o programa múltiplas vezes, verificando que nunca ocorreu nenhum deadlock.

Na figura anterior conseguimos ver como interagem as várias entidades num restaurante com duas mesas e cinco grupos.

CH, WT e RC são, respetivamente, o Chef, o Waiter (empregado de mesa) e Rececionista. G00-G04 representa cada um dos grupos que foram ao restaurante. gWT corresponde ao nº de grupos à espera por uma mesa para jantarem.

Para melhor aferirmos a fiabilidade dos resultados alterámos múltiplas vezes o número de grupos que foram ao restaurante. As análises realizadas foram feitas afim de verificar linha a linha se as mudanças de estado faziam sentido.

Comparámos algumas das nossas execuções com o resultado da execução pré-compilada do professor e, como era de esperar, observámos que eram bastante semelhantes.

# Conclusão

Os conhecimentos para a realização deste trabalho foram adquiridos essencialmente nas aulas práticas e teóricas, não tendo sido necessário qualquer pesquisa da nossa parte.

Um dos maiores desafios que tivemos foi perceber a interação entre as entidades e quais os semáforos a usar aquando dessa interação, isto é, quais os semáforos que deveriam estar ativos e/ou bloqueados, e como poderíamos confirmar se o que fizemos estaria em conformidade com o proposto.

Os programas foram implementados com sucesso, não havendo nenhum deadlock e, tendo passo em todos os testes a que foram propostos.

Este trabalho permitiu-nos desenvolver habilidades de organização, gestão de tempo e perseverança. Enfrentámos desafios ao longo do caminho, mas com dedicação e determinação conseguimos superá-los.