



- Já o segundo semáforo servirá para impedir que vários processos consigam escrever simultaneamente no ficheiro “Logs.txt” garantindo que a informação seja escrita de forma correta e não, por exemplo, de forma distorcida, isto é, as palavras não estejam escritas pela ordem correta.

Outro tipo de, digamos, mecanismo de sincronização que implementamos foi a existência de uma variável do tipo *int* dentro da *struct* correspondente a cada equipa. Esta variável serve para indicar se a box estará livre ou reservada ou cheia (caso haja uma avaria num carro da equipa ou este precise de reabastecer o depósito).

A partir de uma variável *int* “condition” presente na memória partilhada, os processos de todas as equipas ficaram à espera do comando “START RACE!” para criarem as suas threads, respetivas a cada carro. Assim que essa variável passar para outro estado 1 a corrida irá iniciar, sendo que quando esta variável voltar a 0 significa que nenhuma corrida está a decorrer.

Utilizámos uma outra variável *int* “ctrlc” para gerir a possibilidade de começar outra corrida, seja porque a atual foi interrompida pelo SIGUSR1 ou porque todos os carros chegaram à meta. Esta variável serve também para impedir de serem “aceites” comandos enquanto uma corrida está a decorrer.

Temos uma última variável *int* “finish” que controla quando um sinal SIGUSR1 ou SIGINT é recebido, indicando a cada carro que assim que chegar à meta deve parar a corrida e mudar para o estado “terminado”.

## Escolhas:

- Quando um carro fica na box, seja por abastecer o depósito ou reparação de alguma avaria, ele irá encher o depósito.
- Criamos um sinal SIGUSR2 para quando um carro precisar de ir à box avisar a equipa e conseguir reservar a box para ele, impedindo que outro carro possa entrar na box.
- Quando um sinal SIGUSR1 é capturado a corrida termina limpando também todos os dados que estejam na memória partilhada.
- Na variável “table” presente na memória partilhada não guardamos o top 5 de carros da corrida, sendo que esse top é apenas construído dentro da função executada pelo SIGTSTP.
- Assim que uma corrida termina, os dados na memória partilhada são limpos e é possível iniciar uma nova (a não ser que a corrida termine por ação do Ctrl+C).

## Horas de trabalho:

Visto que trabalhamos sempre em conjunto as horas de trabalho serão iguais para os dois elementos do grupo, devendo rondar as ~80 horas.