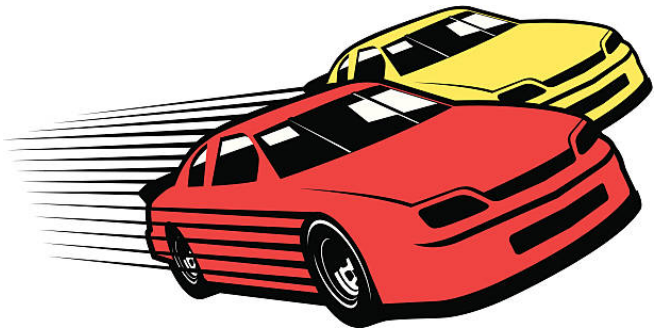
Faculdade de Ciências e Tecnologia Universidade de Coimbra
Departamento de Engenharia Informática
2020/21

Simulador de Corridas

Sistemas Operativos - Projeto Final

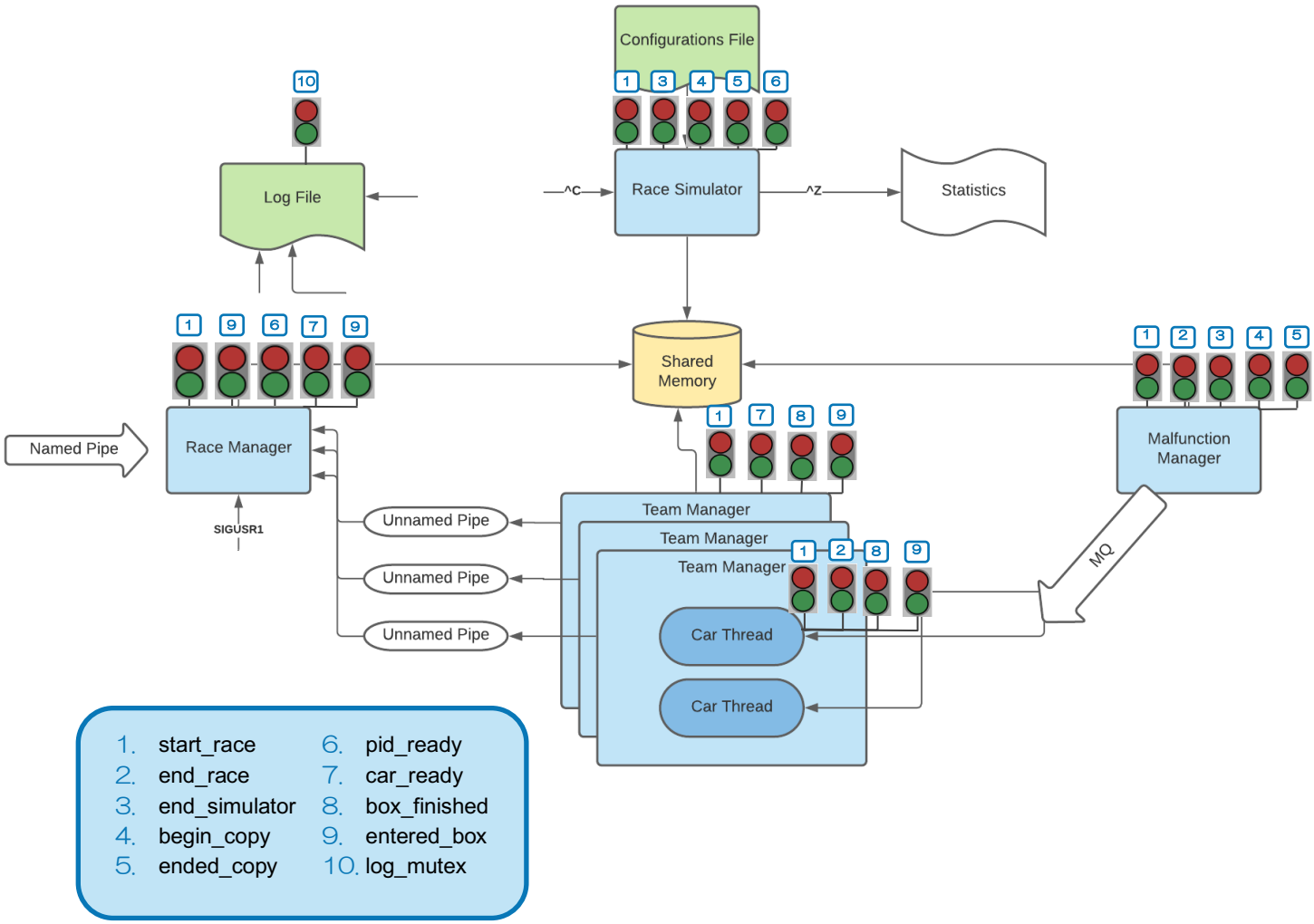


Trabalho realizado por:
André Guilherme Gonçalves Figo Carvalho n. º2019216156
Sofia Botelho Vieira Alves n. º2019227240

Professor Orientador:
Vasco Nuno Sousa Simões Pereira

23 de maio 2021

Arquitetura do cenário e os mecanismos de sincronização



Principais escolhas tomadas ao logo do projeto

Com o intuito de ter uma boa sincronização do tempo dentro da corrida, escolheu-se um dos processos para funcionar como “clock”, de acordo com o qual todos os carros e equipas iriam funcionar. Para isso, escolheu-se o gestor de avarias, visto que este é o processo que passa mais tempo inativo e não recebe qualquer tipo de sinal. Para este efeito, foram utilizadas variáveis de condição, uma para verificar se todos os carros estavam prontos para começar uma nova unidade de tempo (end_tunit) e outra para os informar que a poderiam iniciar (new_tunit).

Em memoria partilhada, são guardados 3 tipos de estruturas, uma para informações gerais da corrida, outra para as dos carros e outra para as das equipas.

Para o início da corrida, todos os processos e *threads* são notificados pelo gestor de corrida, utilizando um semáforo (start_race).

Para as entradas na *box*, os carros utilizam um semáforo para sinalizar que entraram numa *box* (entered_box). Para a saída acontece o contrário, sendo a *box* a recorrer a um semáforo para avisar que o carro está pronto para correr novamente (box_finished). Enquanto um carro estiver na *box*, é o processo da mesma que faz o controlo da passagem do tempo.

Para tratamento dos sinais SIGUSR1 e SIGINT, são utilizadas variáveis em memória partilhada para informar que a corrida deverá interromper ou parar. Os outros processos depois irão consultar o valor destas variáveis para saber se há ou não a necessidade de recomeçar a corrida. Depois de uma interrupção, não são aceites novos carros.

Caso a corrida ainda não tenha começado, o SIGINT irá terminar todos os processos e libertar todos os recursos. Para isso acontecer, é feita a verificação de que todos os processos já foram criados, usando um semáforo (pid_ready).

A finalização da corrida é feita pelo controlo do número de carros em estado terminado ou desistência, tanto pelo gestor de avarias como pelo gestor de corrida. Apenas em casos em que recebe um SIGINT ou quando a corrida termina normalmente, serão notificadas as *boxes*, através do envio de um carro inválido para cada *box*, e o simulador de corrida, através de um semáforo (end_simulator). Assim, todos os processos poderão terminar de forma controlada e todos os recursos serão libertados.

Para o controlo da escrita no ficheiro de log é utilizado um *mutex* (log_mutex). Para além deste *mutex*, são usados vários outros no controlo do acesso à memória partilhada, da escrita nos *pipes* e das variáveis de condição.

Para imprimir as estatísticas da corrida através do SIGTSTP, espera-se que comece uma nova unidade de tempo para poder fazer uma cópia dos dados dos carros naquele instante. Feita a cópia, o gestor de avarias é notificado e a corrida pode voltar ao normal, sem que o cálculo e a impressão das estatísticas impeçam o decorrer da corrida. Para tal são usados dois semáforos (begin_copy e end_copy).

Tempo total despendido por aluno

- André Carvalho: 30-40 horas
- Sofia Alves: 20-30 horas