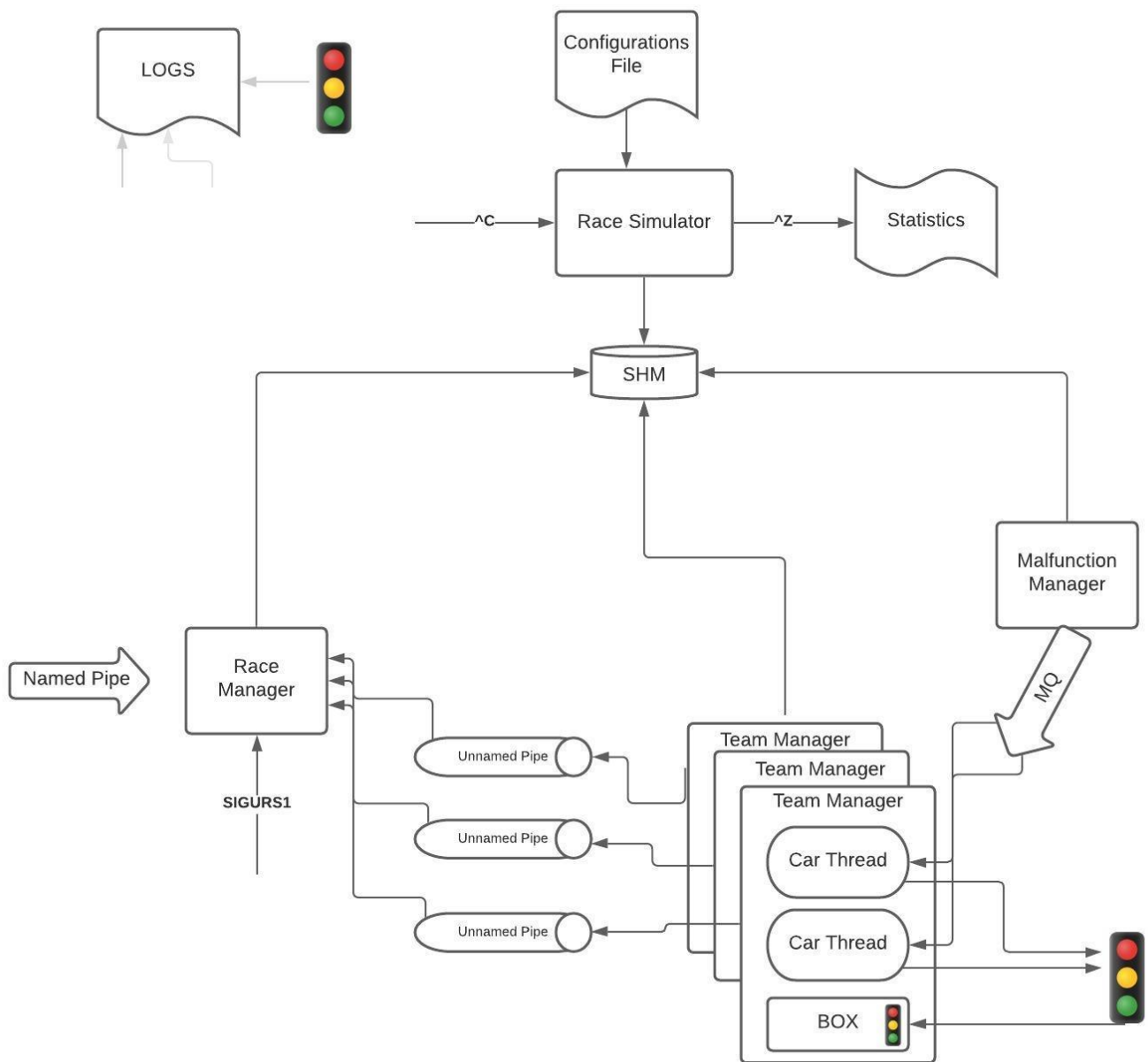


Relatório SO



Trabalho realizado por :

Francisco António Da Cruz Faria 2019227549

Iago Silva Bebiano 2019219478

Horas despendidas: 20 horas cada um.

Simulador de corrida:

- Foi usada a função *signal* para tratar os sinais *ctrl c* e *ctrl z*, ignorando-os inicialmente, até que todos os valores necessários sejam inicializados, posteriormente, ao serem recebidos correm a função *termination_handler*;

Gestor de corridas:

- Altera a reação ao sinal SIGUSR1 que previamente estava a ser ignorado;
- De seguida cria todas as equipas usando o *fork()*;
- Após todas as equipas e respetivos *pipes* serem criados, entra num ciclo *while* infinito no qual recorre à função *select* para ler dos diversos *pipes*;
- Depois de receber o comando “START RACE!” este processo verifica se existem pelo menos 3 equipas, em caso afirmativo, notifica todos os outros processos usando o *pthread_cond_broadcast* desta maneira evita-se a espera ativa;
- Para evitar a existências de mais semáforos, apenas este processo poderá alterar todas as variáveis em memória partilhada exceto as equipas e o *recieved_ctrl*.

Threads Carro:

- Depois de sair da variável de condição esta *thread* reinicializa todos os valores necessários e entra num ciclo *while*;
- Tenta ler da *message queue*, usando a *flag* *IPC_NOWAIT* de modo a não bloquear enquanto espera mensagens, continuando a corrida como desejado;
- Na entrada da box, o *sem_trywait()* tenta bloquear o semáforo sem entrar em espera ativa, depois corre o *sem_post(&go_box)* o que permite ao gestor de corrida perceber que esta um carro na box para arranjar .
- Após alterar o semáforo *go_box* utiliza-se o *sem_wait()* para esperar que o gestor de equipas acabe de arranjar o carro em questão .

Gestor de equipas:

- Antes de criar as threads este inicializa os semáforos referidos anteriormente.
- De seguida espera pelo início da corrida;
- Entra num ciclo *while* que irá permitir a gestão da box e a reparação dos carros quando estes necessitarem;
- Após receber o sinal SIGUSR1 entra no *termination_handler()* onde irá esperar que todos os carros terminem a volta limpando todos os semáforos anteriormente utilizados;

Gestor de avarias:

- Tal como as *Threads* e o gestor de equipas este espera pelo *broadcast* e depois começa a gerar erros para todos os carros inicializados tendo em conta a *reliability* dos mesmos.

Ficheiros de log:

- De modo a garantir a qualidade da informação obtida, a função que escrever no ficheiro log apenas permite a escrita por um processo de cada vez, utilizando um semáforo.