FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D COIMBRA

Sistemas Operativos

2022/2023

Relatório da meta final de Recurso do trabalho prático

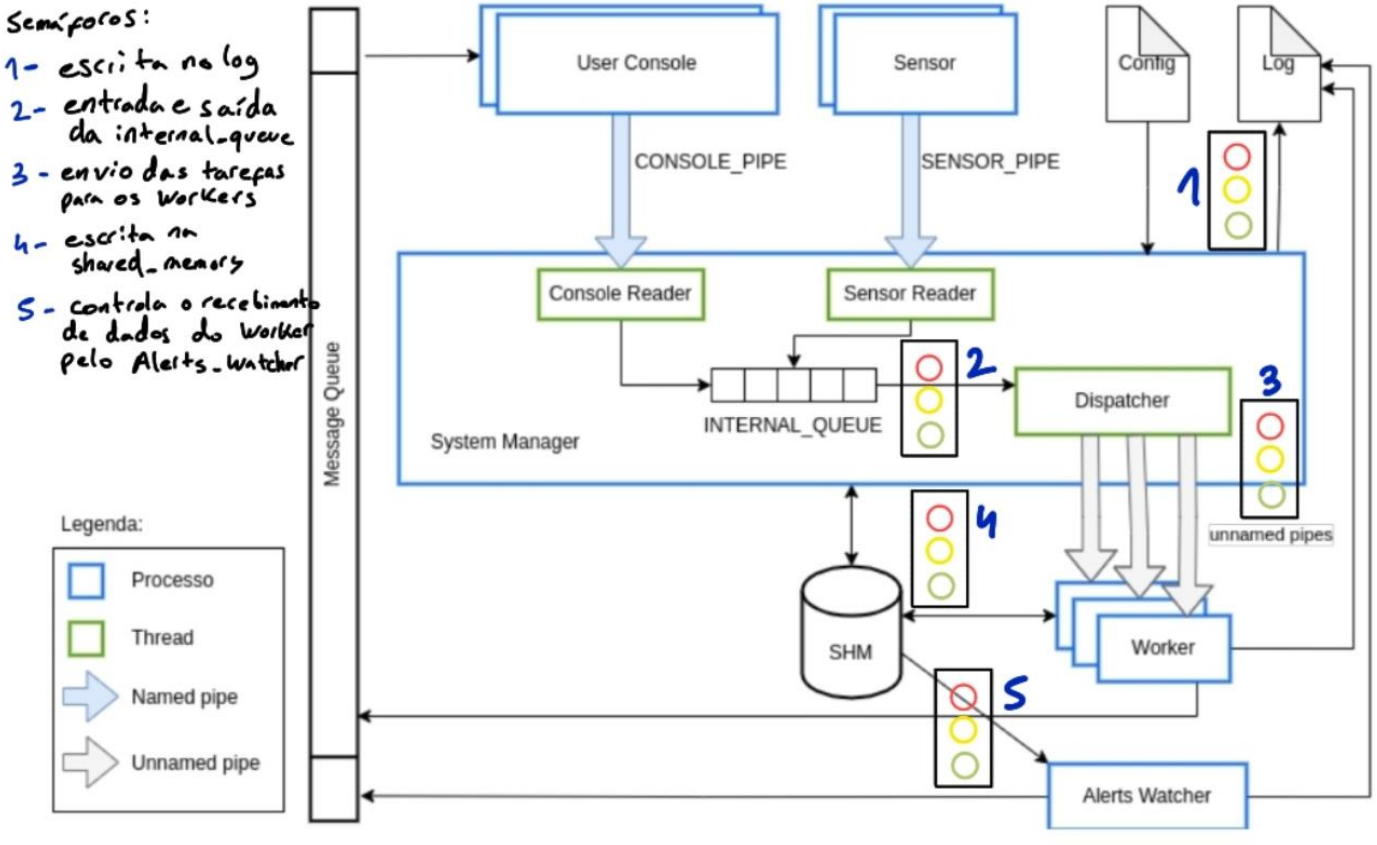
Simulador para Internet das Coisas em contexto habitacional

Trabalho realizado por:

Pedro Ramalho nº 2019248594

André Pinto nº 2021213497

Esquema da arquitetura do programa



Opções tomadas na construção do programa

Criamos três ficheiros, um para o processo System Manager,o system_Manager.c, outro para o processo Sensor, o sensor.c, e um para o User Console, o userConsole.c.

Utilizamos o os dados do ficheiro configs.txt para inicializar as ariáveis QUEUE_SZ, N_WORKERS, MAX_KEYS, MAX_SENSORS e MAX_ALERTS.

O tamanho da shared memory é calculado a partir da seguinte expressão: SHM_SIZE MAX_SENSORS * (sizeof(Infos_sensor)) + MAX_ALERTS * (sizeof(Alerta)) + N_WORKERS * (sizeof(int)) , reservando assim o espaço para o máximo de dados fornecido pelo ficheiro.

O acesso à shared memory é realizado através de três ponteiros, sensor, alert e worker_valid_dispatcher, que apontam para o primeiro elemento de cada zona de memoria que é reservada para cada estrutura dentro do espaço reservado para a memoria partilhada.

Para os pipes utilizamos os tipos que são solicitados no enunciado do projeto.

Para a INTERNAL_QUEUE optamos por uma lista ligada, ou seja, uma estrutura com uma variável data e um ponteiro que aponta para o próximo elemento da fila. Que permite enviar os dados gerados pelo sensor e os comandos inseridos no User Console pelos respetivos named pipes, SENSOR_PIPE e CONSOLE_PIPE para a thread dispatcher.

No dispatcher damos prioridade às mensagens enviadas pelo UserConsole. Criamos um unnamed pipe para cada worker, que transportam as mensagens.

Nos workers, lemos a mensagem e realizamos as tarefas pedidas, consoante o tipo de comando, atualizamos a shared memory, escrevemos no ficheiro log e enviamos a resposta aos comandos pela message_queue de volta para o user console.

O estado dos workers é guardado na shared memory, utilizando a seguinte logica: é colocado a 1 quando está ocupado e a 0 quando não está a realizar nenhuma tarefa.

No processo alerts watcher verificamos se algum dos valores ultrapassou os limites provenientes do ficheiro e caso tenha ultrapassado é enviado um alerta para a fila de mensagens.

A Message_queue é uma fila de mensagens do tipo SysV, permitindo assim o envio das respostas por parte dos processos alerts watcher e workers com destino ao User Console.

No processo sensor são gerados os valores aleatórios, com base nos intervalos fornecidos pela linha de comandos, que são enviados para o System Manager

No processo User Console é apresentado o menu com as diferentes opções que o utilizador pode realizar. As inserções feitas pelo utilizador são verificadas conforme descrito no enunciado.

Nos três ficheiros é feita a verificação dos sinais. Caso seja pressionado o Ctrl + C (SIGINT) os programas chamam a respetiva função cleanup que liberta toda a memoria utilizada no programa e encerra as tarefas em curso. No processo sensor caso seja pressionado Ctrl + Z(SIGTSTP) é mostrado o número de mensagens enviadas até ao momento.

Todas todos os eventos relevantes são escritos no ficheiro “log.txt” acompanhados da sua data e hora.

A execução do programa é controlada pelos semáforos e mutexes representados no diagrama.