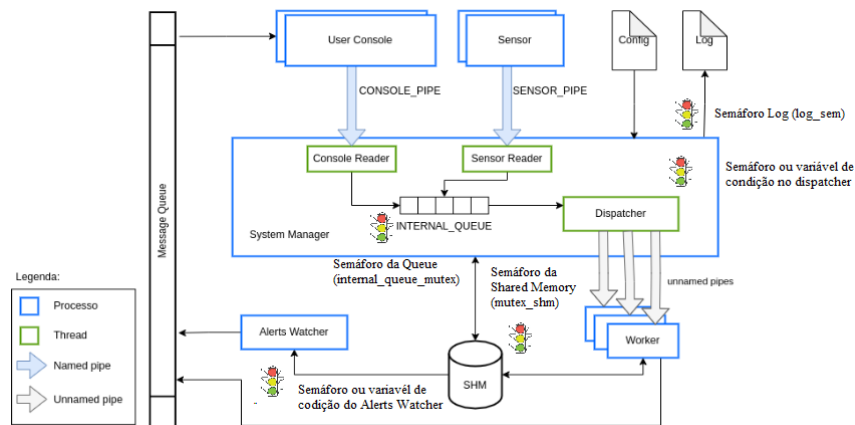


Arquitetura:



Semáforo, variáveis de condição e Mutex:

Os semáforos são utilizados para sincronizar o acesso a recursos compartilhados. No nosso caso, os semáforos são utilizados para sincronizar o acesso à memória partilhada, à fila de mensagens e ao ficheiro de log. A seguir iremos explicar como é que os semáforos são utilizados em cada um dos casos.

As variáveis de condição permitem que uma thread espere até que uma condição seja satisfeita. No nosso caso, as variáveis de condição serão utilizadas para que o dispatcher espere até que a internal queue tenha elementos para serem lidos. Poderá também vir a ser necessário utilizar variáveis de condição no alert watcher, em vez do atual semáforo para evitar o busy wait.

O Mutex é uma forma de implementar a exclusão mútua, que garante que apenas uma thread execute uma parte crítica do código de cada vez. No nosso caso, o mutex é utilizado para sincronizar o acesso à fila de mensagens interna.

Memória partilhada:

A memória partilhada tem comunicação com o System Manager, Worker e Alert Watcher sendo utilizada para guardar os dados dos sensores e enviar os dados para o servidor. Para que os dados não sejam corrompidos, é necessário sincronizar o acesso à memória partilhada, para isso, é utilizado um semáforo. O semáforo que usamos tem o nome de `'mutex_shm'` e é criado no System Manager.

Alert Watcher e Worker:

Criamos um semáforo chamado `'check_alert_sem'` que é acionado quando se adiciona uma nova key, ou se atualiza uma key existente. O semáforo é colocado em espera quando o alert watcher verificou os alertas, e é libertado quando o worker recebe uma nova key.

Internal Queue:

Na internal queue colocamos um mutex, da forma `pthread_mutex_t`, para sincronizar o acesso à fila. O nome da variável é `'internal_queue_mutex'`. A este mutex, vamos fazer lock quando fazemos alguma operação na fila, e unlock quando terminamos a operação. Por exemplo, quando adicionamos um elemento à fila, fazemos lock, adicionamos o elemento, e depois fazemos unlock. Quando retiramos um elemento da fila, fazemos lock, retiramos o elemento, e depois fazemos unlock. Achamos que também será necessário um variável de condição para indicar que a internal queue tem elementos a serem lidos para o dispatcher, para assim podermos economizar recursos.

Ficheiro de Log:

O ficheiro de log tem comunicação com o System Manager servindo para escrever todas as informações que o System Manager recebe e serem analisadas posteriormente. Para que as informações não sejam corrompidas, é necessário sincronizar o acesso ao ficheiro de log, para isso, é utilizado um semáforo. O semáforo que usamos tem o nome de `'log_sem'` e é criado no System Manager.

Futuro:

Para o futuro, poderão ser necessários mais semáforos para sincronizar o acesso a outros recursos partilhados como no Worker e no Alert Watcher, bem como para evitar possíveis busy waits. Poderá também vir a ser necessário trocar o semáforo do alert watcher por uma variável de condição.