Fundamentos de Sistemas de Operação -

Questão 2 (T2).

Considere que, no sistema Unix, se pretende realizar o esquema seguinte de comunicação por mensagens entre três processos concorrentes **P1**, **P2** e **P3**. Admita que as variáveis **P1**, **P2** e **P3** assumem, respectivamente, os valores dos identificadores daqueles processos no sistema Unix (*process identifiers*). Os processos estão obrigados a comunicarem entre si através do seguinte esquema, em que **fm1**, **fm2**, **fm3** são filas de mensagens Unix (message queues):

```
P1 --> fila fm1 ----->
                               procS -----> fila fm3 ----> P3
P2 --> fila fm2 ----->
```

em que: **P1** só tem acesso a **fm1**; **P2** só tem acesso a **fm2**; **P3** só tem acesso a **fm3**. O servidor auxiliar **procS** tem acesso às três filas e tem como função ir transferindo as mensagens para **fm3**, à medida que chegam a **fm1** ou a **fm2**. (Note: As acções de **procS** podem ser executadas concorrentemente.)

Pretende-se realizar as duas operações seguintes:

- **enviar(M)**: invocada pelo processo **P1** ou por **P2**, desencadeia o envio de uma mensagem **M**, de modo a que possa ser recebida pelo processo **P3**. O envio é não bloqueante. A mensagem **M** deve basear-se na convenção da comunicação por filas de mensagens (*message queues*) do Unix IPC System V.

- **receber(M, Pe)**: invocada por **P3**, recebe uma mensagem, devolvida no argumento **M**. A mensagem recebida depende do valor passado no argumento **Pe**:

- se **Pe** == 0 - deve receber a mensagem mais antiga que tenha chegado a **fm3**.
- se **Pe** == **P1** - deve receber a mensagem mais antiga enviada por **P1**
- se **Pe** == **P2** - deve receber a mensagem mais antiga enviada por **P2**

Esta operação de recepção é bloqueante.

==> Programa em C, usando chamadas ao Unix, as acções das operações **enviar()** e **receber()** e as acções do processo **procS** para transferir as mensagens. Não necessita de programar a criação das filas nem dos processos. Nesta questão 2 (T2), não pode utilizar **pipes**, **memória partilhada** ou **semáforos**.

Questão 3 (T2).

Admita que se pretende realizar um esquema de comunicação por mensagens entre três processos **P1**, **P2** e **P3**, como na Questão 2 (T2), mas implementado sobre comunicação por memória partilhada. Admita que, em vez de filas de mensagens, existem as seguintes áreas de memória partilhada (*shared memory Unix IPC System V*), exclusivas para a comunicação entre os processos:

```
P1 --> SM1 ----->
                               procS -----> SM3 ----> P3
P2 --> SM2 ----->
```

SM1 - exclusiva para comunicação entre **P1** e **procS** (onde **P1** põe mensagens)
SM2 - exclusiva para comunicação entre **P2** e **procS** (onde **P2** põe mensagens)
SM3 - exclusiva para comunicação entre **procS** e **P3** (onde **P3** obtém mensagens)

Como na Questão 2 (T2), o servidor auxiliar **procS** tem acesso às três áreas e tem como função ir transferindo as mensagens para **SM3**, à medida que são colocadas em

SM1 ou **SM2**. Admita mensagens de um tamanho fixo e que cada fila pode conter um máximo de MAX mensagens.

Admita que existem duas primitivas auxiliares '**inserir(fila,M)**' e '**remover(fila,M)**' para aceder às filas em memória partilhadas, mas que não estão protegidas contra acessos concorrentes.

Baseando-se em **semáforos**, apresente o pseudo-código das acções do processo **procs** para o encaminhamento das mensagens e o pseudo-código das operações seguintes:

enviar(M): invocada por **P1** ou **P2**, desencadeia o envio de uma mensagem **M**, de modo a que possa ser recebida pelo processo **P3**. O envio é não bloqueante.

receber(M): invocada por **P3**, recebe uma mensagem, em **M**. A recepção é bloqueante. Suponha que as áreas de memória partilhadas e os processos foram previamente criados. Nesta questão, não pode utilizar **filas de mensagens Unix nem pipes**.