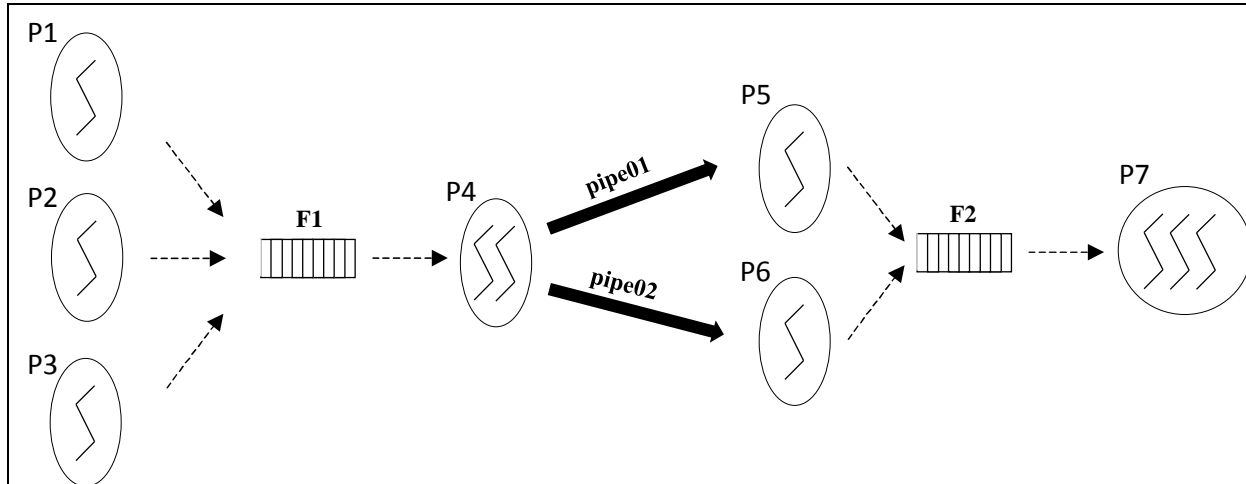


Atividade Prática - Unidade III

Implementar um programa que realize o processamento ilustrado na figura abaixo:



Os processos P1, P2 e P3 são *singlethreads*. Cada um desses processos se comunica com o processo P4 por meio de uma área de memória compartilhada (*shared memory*). Essa área compartilhada deve ser usada como uma fila (F1) do tipo FIFO (*first-in-first-out*), a qual armazena valores do tipo inteiro. A capacidade da fila é de 10 valores.

Os processos P1, P2 e P3 são produtores e o processo P4 é consumidor nessa comunicação. P4 sempre aguarda um sinal (SIGUSR1) para consumir dados da F1. Esse sinal é enviado para P4 quando F1 estiver cheia, ou seja, com 10 valores. O sinal deve ser enviado para P4 pelo processo que inseriu o décimo valor em F1. Note que neste caso F1 somente receberá valores após P4 retirar o último (décimo) valor de F1, deixando-a vazia para receber novos valores.

Os valores inteiros inseridos em F1 devem ser gerados de forma aleatória na faixa de 1 até 1000.

O acesso a F1 deve ocorrer de forma exclusiva utilizando o mecanismo de semáforo entre os processos envolvidos.

O processo P4 possui duas *threads* (t1 e t2), onde ambas as *threads* retiram valores de F1 e enviam para P5 (t1) e P6 (t2), respectivamente, utilizando o mecanismo de *pipe*. Os processos P5 e P6 ao receberem valores de P4 os enviam para P7 utilizando da fila F2, também implementada como *shared memory*. Neste caso, o controle ao acesso de F2 deve ser implementado usando um mecanismo de exclusão mútua baseado em espera ocupada (*busy wait*).

Diferente de F1, a fila F2 deve ser consumida na medida em que entram novos valores. Se a fila estiver cheia (máximo 10 valores), então os processos P5 e P6 devem aguardar a retirada de ao menos um elemento da fila, por P7, para inserirem novos valores.

Todas as três *threads* de P7 retiram valores de F2 e os imprimem na tela.

Após P7 imprimir 10000 valores, o programa deve imprimir o seguinte relatório:

- Tempo total de execução do programa.
- Quantidade de valores processados por P5 e P6.

- c) Dentre os valores impressos, informar o valor que mais se repetiu (moda), o valor mínimo e o valor máximo.

obs: para gerar números aleatórios use

```
#include <stdlib.h>
int rand(void);
int rand_r(unsigned int *seedp);
void srand(unsigned int seed);
```