Filas de mensagens

Este projeto faz parte do PPOS v2.

Uma fila de mensagens é uma estrutura usada pelas tarefas para comunicação através de troca de mensagens de tamanho fixo. Cada fila pode armazenar até N mensagens de tamanho fixo, respeitando uma política de acesso FIFO.

O acesso à fila é **bloqueante**, ou seja, uma tarefa que tenta enviar uma mensagem a uma fila cheia terá de esperar até que surjam vagas na fila; uma tarefa que deseja receber mensagens de uma fila vazia terá de esperar até que uma mensagem esteja disponível nessa fila.

As funções a implementar são descritas a seguir.

Cria uma fila de mensagens

```
int mqueue_create(int max_msgs, int msg_size)
```

Cria uma nova fila de mensagens com capacidade para receber até `max_msgs` mensagens de tamanho `msg_size` bytes cada, inicialmente vazia. Retorna o UID da nova fila ou -1 em caso de erro.

Exemplo: inicia uma fila para guardar até 100 números de tipo `long`:

```
int fila1 ;

fila1 = mqueue_create(100, sizeof (long));
if (fila1 < 0)
{
   printf("Erro ao iniciar a fila\n");
   exit(1);
}
```

Envia uma mensagem para a fila

```
int mqueue_send(int queue_id, void *msg)
```

Envia a mensagem apontada por `msg` para o fim da fila com ID `queue_id`; esta chamada é bloqueante: caso a fila esteja cheia, a tarefa corrente é suspensa até que o envio possa ser feito. O ponteiro `msg` aponta para um *buffer* contendo a mensagem a enviar, que deve ser copiada para dentro da fila. Retorna 0 em caso de sucesso e -1 em caso de erro.



Sugestão: copie a mensagem do *buffer* para a fila (ou vice-versa) usando funções C como `bcopy` ou `memcpy`.

Exemplo: Enviar o valor de uma variável para a fila `fila1`:

```
long x = 1234567890;

if ( mqueue_send(fila1, &x) )
{
```

```
   printf("Erro ao enviar mensagem\n");
   exit(1);
}
```

Recebe uma mensagem da fila

```
int mqueue_recv(int queue_id, void *msg)
```

Recebe uma mensagem do início da fila com ID `queue_id` e a deposita no *buffer* apontado por `msg`; esta chamada é bloqueante: caso a fila esteja vazia, a tarefa corrente é suspensa até que a recepção possa ser feita. O ponteiro `msg` aponta para um *buffer* que irá receber a mensagem. Retorna 0 em caso de sucesso e -1 em caso de erro.

Exemplo: Receber uma mensagem da fila `fila1` e colocar seu valor na variável `buf`:

```
long buf;

if ( mqueue_recv(fila1, &buf) )
{
  printf("Erro ao receber mensagem\n");
  exit(1);
}
else
{
  printf("Recebi o valor %ld\n", buf);
}
```

Encerra uma fila

```
int mqueue_destroy(int queue_id)
```

Encerra a fila de mensagens indicada pelo ID `queue_id`, destruindo seu conteúdo e liberando todas as tarefas que esperam mensagens dela (essas tarefas devem retornar das suas respectivas chamadas com valor de retorno -1).

Retorna 0 em caso de sucesso e -1 em caso de erro.

Número de mensagens

```
int mqueue_msgs(int queue_id)
```

Informa o número de mensagens presentes na fila com ID `queue_id`. Retorna 0 ou +N em caso de sucesso e -1 em caso de erro.

UIDs das filas de mensagens

Cada fila de mensagens é identificada pela aplicação por um UID inteiro positivo. Use um mapa genérico para gerenciar esses UIDs.

Observações:

Filas de mensagens são na verdade *buffers* limitados acessados por tarefas que produzem e consomem mensagens. O código do sistema produtor/consumidor é um bom ponto de partida para implementar as funções `mqueue_send` e `mqueue_recv`.

As filas de mensagens devem ser genéricas em relação aos tipos de mensagens suportadas. Em outras palavras, a implementação deve suportar o envio de qualquer tipo de dado de tamanho conhecido, não sendo restrita a um tipo ou dois.

O *buffer* interno da fila pode ser implementado com uma estrutura de buffer circular, ou pode ser usada a fila genérica já implementada.



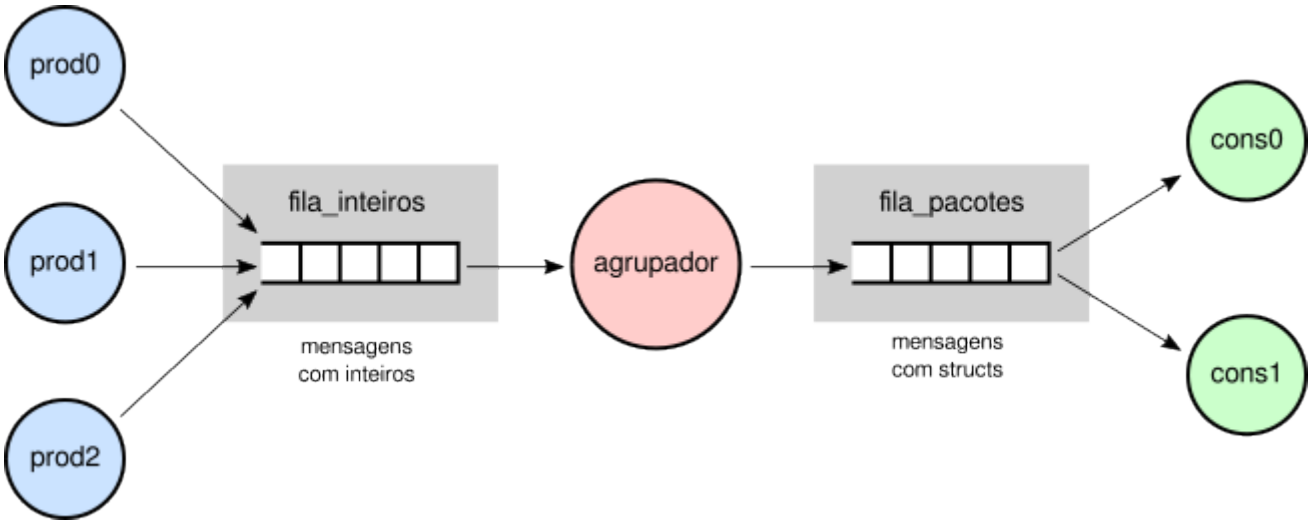
Cada valor enviado a uma fila de mensagens deve ser **copiado** para ela, pois o emissor pode alterar a variável após o envio, o que alteraria o conteúdo da fila. O mesmo raciocínio se aplica à recepção de valores da fila.

Arquivos

Os seguintes arquivos são relevantes para este projeto:

- `kernel/mqueue.c`: implementação das filas de mensagens
- `test/pingpong-mqueue.c`: programa de teste
- `test/pingpong-mqueue.txt`: saída esperada do teste

O programa de teste implementa um sistema produtor-consumidor complexo, com processos que produzem e consomem valores de acordo com a arquitetura apresentada na figura abaixo:



Outras informações

- Duração estimada: 6 horas.
- Dependências:
  - Tarefas cooperativas
  - Despachante de tarefas
  - Preempção por Tempo

- Tarefas que esperam
- Tarefas que dormem
- Spinlocks e Semáforos
- Fila genérica
- Mapa genérico

From:
https://wiki.inf.ufpr.br/maziero/ - **Prof. Carlos Maziero**

Permanent link:
https://wiki.inf.ufpr.br/maziero/doku.php?id=ppos-v2:filas_de_mensagens

Last update: **2026/07/29 18:00**