Filas de mensagens

Video deste projeto

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.

Inicia uma fila de mensagens

```
int mqueue_init (mqueue_t *queue, int max_msgs, int msg_size)
```

Inicia a fila de mensagens apontada por `queue`, com capacidade para receber até `max_msgs` mensagens de tamanho `msg_size` bytes cada, inicialmente vazia. Retorna 0 em caso de sucesso e -1 em caso de erro.

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

```
mqueue_t fila1 ;

if ( mqueue_init (&fila1, 100, sizeof (double)) )
{
   printf ("Erro ao iniciar a fila\n") ;
   exit (1) ;
}
```

Envia uma mensagem para a fila

```
int mqueue_send (mqueue_t *queue, void *msg)
```

Envia a mensagem apontada por `msg` para o fim da fila `queue`; 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`:

```
double pi = 3.1415926535897932 ;

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

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

Recebe uma mensagem da fila

```
int mqueue_recv (mqueue_t *queue, void *msg)
```

Recebe uma mensagem do início da fila `queue` 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`:

```
double buf ;

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

Encerra uma fila

```
int mqueue_destroy (mqueue_t *queue)
```

Encerra a fila de mensagens indicada por `queue`, 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 (mqueue_t *queue)
```

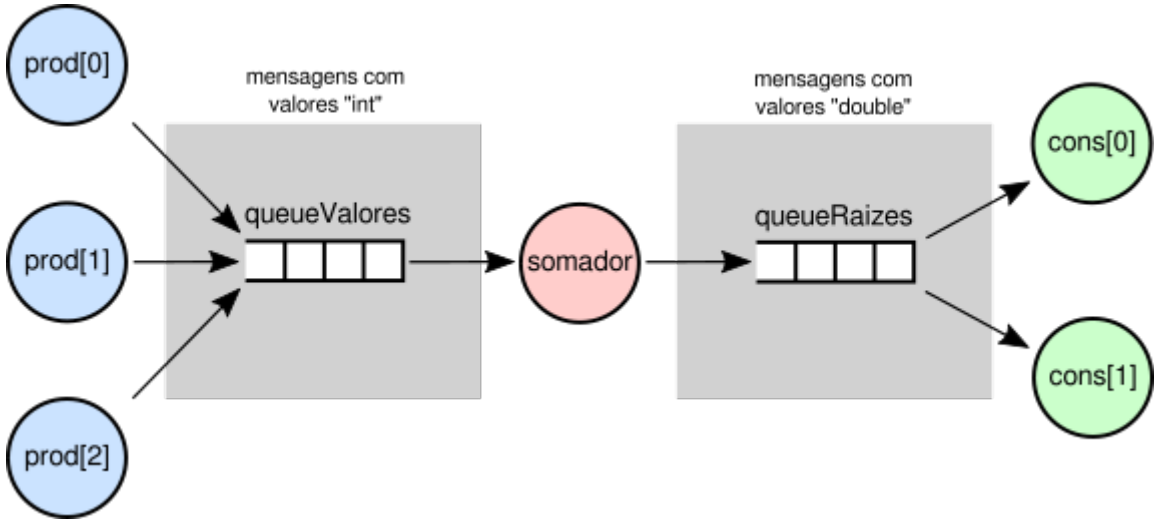
Informa o número de mensagens presentes na fila indicada por `queue`. Retorna 0 ou +N em caso de sucesso e -1 em caso de erro.

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.

Seu código deverá funcionar com este programa de teste e deverá gerar um resultado similar a este. A figura abaixo representa graficamente esse programa de teste:



Outras informações

- Duração estimada: 6 horas.
- Dependências:
  - Gestão de Tarefas
  - Dispatcher
  - Preempção por Tempo
  - Tarefas suspensas
  - Tarefas dormindo
  - Semáforos

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

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

Last update: **2023/05/03 17:37**