

```
typedef struct task_t
{
    ... struct task_t *prev, *next; ... //
    ... int id; ... //
    ... ucontext_t context; ... //
    ... void *stack; ... //
    ... struct task_t *parent; ... //
    ... enum status_t status; ... //
    ... // ... (outros campos serão adicionados) ...
} task_t;
```

**Carlos Maziero**

## Filas de mensagens

Uma fila de mensagens é uma estrutura usada pelas tarefas para comunicação de mensagens de tamanho fixo. O acesso à fila é bloqueante, ou seja, uma tarefa que tenta colocar uma mensagem em uma fila cheia terá de esperar até que surjam vagas; uma tarefa que deseja receber mensagens de uma fila vazia terá de esperar até que uma mensagem seja enviada para a fila.

As funções a implementar são descritas a seguir.

### Cria uma fila de mensagens

```
int mqueue_create (mqueue_t *queue, int max, int size)
```

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

### 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 através de funções C como `bcopy` ou `memcpy`. Retorna 0 em caso de sucesso e -1 em caso de erro.

### 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, que deve ser copiada através de funções C como `bcopy` ou `memcpy`. Retorna 0 em caso de sucesso e -1 em caso de erro.

### 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 se comportam como produtoras e consumidoras de mensagens. A fila de mensagens em si é uma região crítica que deve ser protegida de eventuais condições de disputa. Para isso, você deve utilizar os semáforos desenvolvidos no módulo anterior em sua implementação.

Seu código deverá funcionar com este programa de teste e deverá gerar um resultado similar a este.

## Outras informações

- Duração estimada: 6 horas.
- Dependências:
  - Gestão de Tarefas
  - Dispatcher
  - Preempção por Tempo
  - Tarefa Main
  - Operador Join
  - Sleeping
  - Semáforos

---

so/filas\_de\_mensagens.txt · Última modificação: 2015/03/27 17:19 por maziero