

Este é o cache do Google de http://dainf.ct.utfpr.edu.br/~maziero/doku.php/so:biblioteca_de_filas. Ele é um instantâneo da página com a aparência que ela tinha em 24 mar. 2015 02:43:24 GMT. A [página atual](#) pode ter sido alterada nesse meio tempo. [Saiba mais](#)
Dica: para localizar rapidamente o termo de pesquisa nesta página, pressione **Ctrl+F** ou **⌘-F** (Mac) e use a barra de localização.

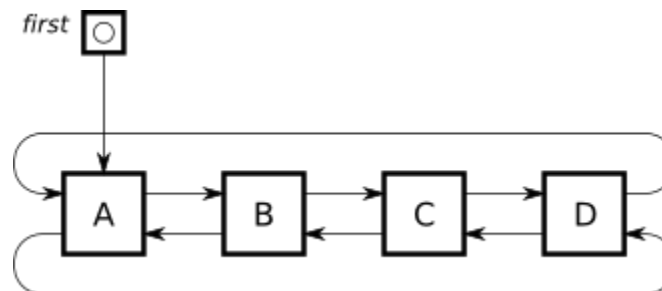
[Versão somente texto](#)

```
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;
```

Prof. Carlos A. Maziero

Biblioteca de Filas

O sistema operacional gerencia muitas filas: de processos prontos, suspensos, dormindo, esperando em semáforos, etc. A estrutura de dados mais adequada para implementar essas filas é uma **lista circular duplamente encadeada**, como indicada na figura abaixo:



Este projeto consiste em construir uma pequena biblioteca que ofereça operações básicas de inserção e remoção em uma lista circular duplamente encadeada totalmente escrita em ANSI C, usando estruturas e ponteiros. A fila é genérica e pode ser usada para organizar vários tipos de dados (**atualizada em maio/2013**).

Esta biblioteca será utilizada em vários outros projetos, portanto capriche na implementação!

Interface

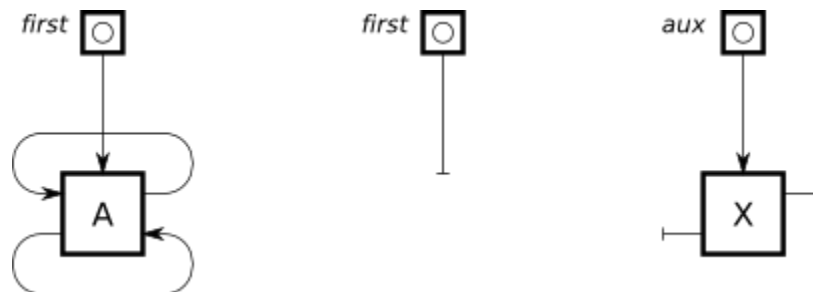
A biblioteca a ser construída deverá respeitar rigorosamente a interface definida no arquivo `queue.h` (que não deve ser modificado). Ela deverá ser totalmente escrita em C ANSI, em um arquivo único chamado `queue.c`, e deverá funcionar corretamente com o programa de teste `testafila.c`. **Somente o arquivo `queue.c` será entregue ao professor.**

Exemplos

Os exemplos abaixo permitem compreender o significado preciso das estruturas e operações a implementar. É importante observar que as operações feitas pela biblioteca consistem somente de manipulações de ponteiros; **não devem ser feitas alocações/liberações de memória dentro da biblioteca.**

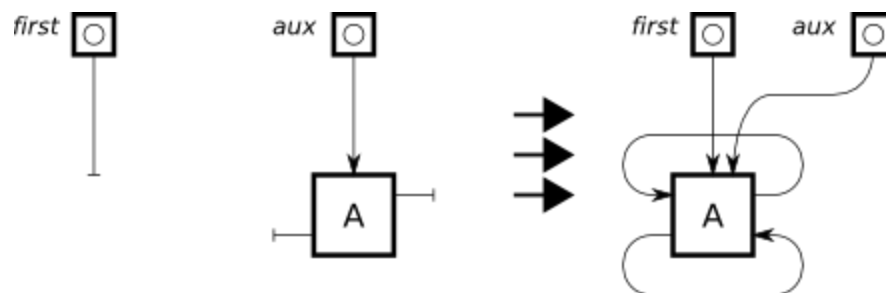
Exemplo 1

Uma fila com um único elemento, uma fila vazia e um elemento isolado (elemento fora de uma fila):



Exemplo 2

Inserção de um elemento em uma fila vazia:



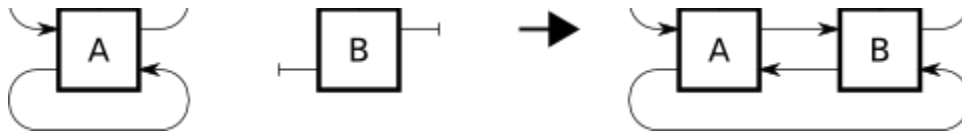
Observe que:

- o elemento a inserir deve estar isolado, ou seja, não deve pertencer a nenhuma outra fila;
- o elemento a inserir já existe, ou seja, não há necessidade de alocar memória para ele (malloc).

Exemplo 3

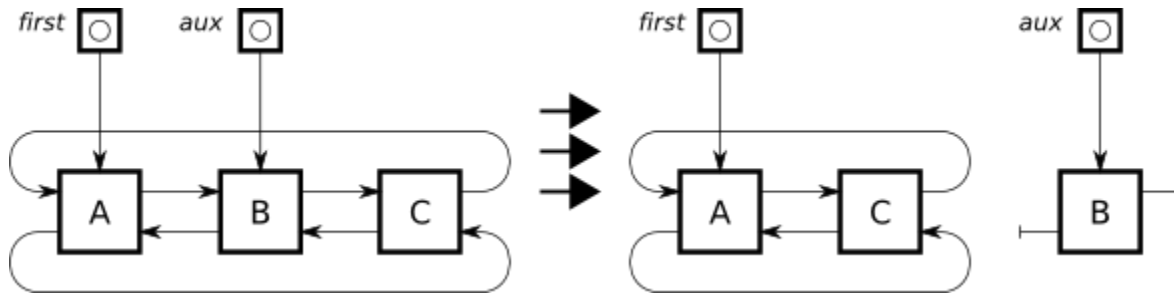
Inserção de um elemento no fim de uma fila não-vazia:





Exemplo 4

Remoção de um elemento da fila, indicado pelo ponteiro aux. Observe que a remoção apenas retira o elemento da fila, sem o destruir, alterar seu conteúdo ou liberar sua memória.



Outras informações

- Duração estimada: 6 horas.
- Dependências:
 - Conhecimento de linguagem C.