O TAD FPrio

Este trabalho visa implementar o tipo abstrato de dado *Fila de Prioridades* (FPrio).

Uma fila de prioridades é uma fila de itens onde cada item é associado a uma prioridade (ou chave); os itens da fila são mantidos ordenados segundo prioridades crescentes.

Filas de prioridades são muito usadas quando há necessidade de manter dados ordenados segundo algum critério. Por exemplo, simulações geralmente usam uma fila de prioridades chamada *Lista de Eventos Futuros* (LEF) para manter os próximos eventos que irão ocorrer na simulação, na ordem crescente do tempo associado a cada evento. Nesse caso, a chave/prioridade de cada evento é o seu instante de ocorrência.

Algumas operações tipicamente definidas sobre uma fila de prioridades são:

- `cria (fila, capacidade)` : cria uma nova fila, informando o número máximo de itens que ela pode comportar; uma fila pode ser considerada infinita, se não tiver capacidade máxima definida.
- `destroi (fila)` : esvazia/destrói a fila.
- `insere (fila, item, prioridade)` : insere um novo item na fila, posicionado de acordo com o valor da prioridade informada.
- `item = retira_min (fila)` : retira o item da fila que tem a **menor** prioridade.
- `item = retira_max (fila)` : retira o item da fila que tem a **maior** prioridade.
- `valor_min (fila)` : informa o valor do item que tem a menor prioridade, sem alterar a fila.
- `valor_max (fila)` : informa o valor do item que tem a maior prioridade, sem alterar a fila.
- `chave_min (fila)` : informa o valor da menor prioridade, sem alterar a fila.
- `chave_max (fila)` : informa o valor da maior prioridade, sem alterar a fila.
- `tamanho (fila)` : informa o número atual de itens na fila.
- ...

Exemplo de funcionamento:

operação (item, prioridade)	**conteúdo da fila**
cria	
insere (A, 3)	**(A, 3)**
insere (B, 6)	(A, 3) **(B, 6)**
insere (C, 1)	**(C, 1)** (A, 3) (B, 6)
insere (D, 2)	(C, 1) **(D, 2)** (A, 3) (B, 6)
insere (E, 2)[1)]	(C, 1) (D, 2) **(E, 2)** (A, 3) (B, 6)
insere (F, 4)	(C, 1) (D, 2) (E, 2) (A, 3) **(F, 4)** (B, 6)
retira min	(D, 2) (E, 2) (A, 3) (F, 4) (B, 6)
retira min	(E, 2) (A, 3) (F, 4) (B, 6)
retira max	(E, 2) (A, 3) (F, 4)

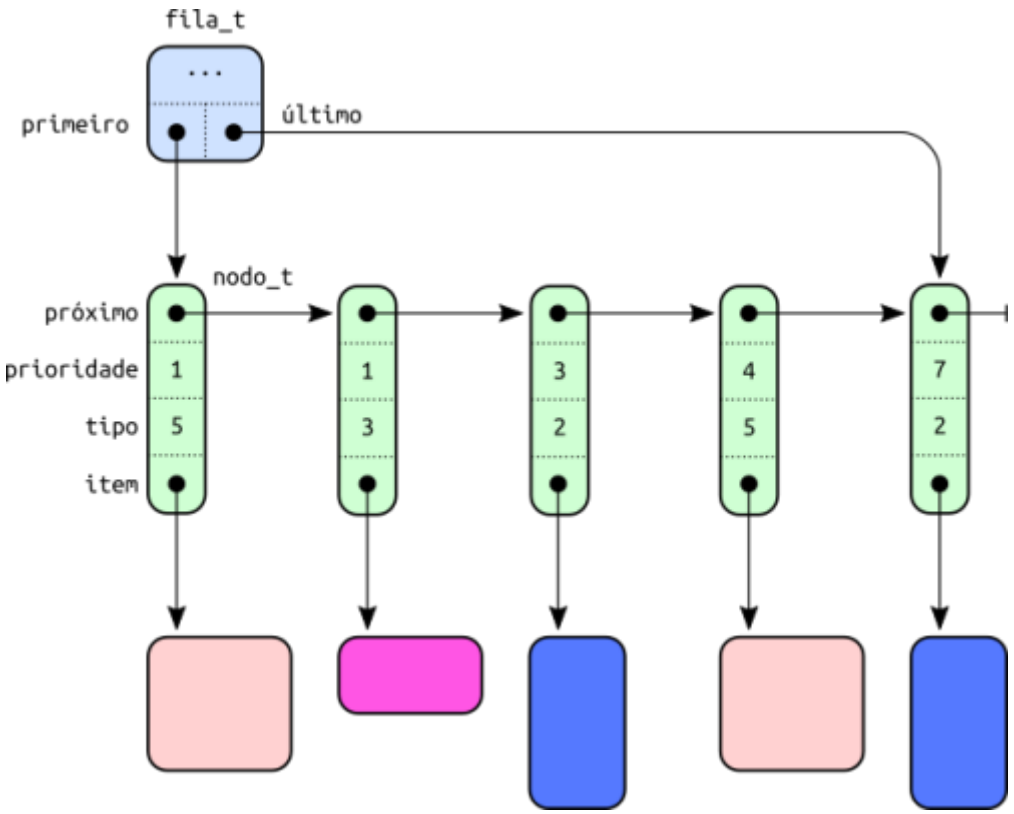
Fila genérica

A fila de prioridades a ser implementada neste projeto deve ser **genérica**, ou seja, ela pode armazenar qualquer tipo de informação (inteiros, structs, outras filas, etc).

A solução para esse problema consiste em:

- a fila não deve armazenar dados, mas somente ponteiros para eles
- os itens da fila devem ser ponteiros genéricos (`void *`) que apontam para conteúdos
- a fila deve armazenar uma informação sobre o tipo do dado apontado em cada item

Uma fila genérica é portanto implementada como uma lista encadeada de ponteiros com tipos associados, assim:



Atividade

Você deve implementar um TAD *Fila de Prioridades* genérico usando uma lista encadeada simples, com as seguintes características:

- A cada item da fila é associada uma **prioridade** (valor inteiro)
- A fila deve ser mantida em ordem crescente de prioridades
- Itens com a mesma prioridade devem respeitar a política FIFO
- Cada item na fila é um ponteiro genérico (`void *`) que aponta para um conteúdo
- A cada item é associado um **tipo** (inteiro), que permite à aplicação saber o que está armazenado ali

Os arquivos necessários para desenvolver este trabalho estão neste arquivo. Eles são:

- `fprio.h`: arquivo de cabeçalho com os protótipos das funções (**não deve ser alterado**).
- `fprio.c`: arquivo que implementa as operações sobre a fila (“esqueleto” a completar).
- `teste-fprio.c`: código que testa a biblioteca (**não deve ser alterado**).
- `teste-fprio.txt`: saída do código que testa a biblioteca.
- `makefile`: arquivo do utilitário “make” para compilar seu código.

Entregáveis

Entregue um único arquivo `tp5.tgz` contendo todos os arquivos do projeto.

Critérios de avaliação:

- Funcionar corretamente

- Não ter problemas de memória (testar com Valgrind)
- Respeitar o conceito de TAD
- Usar funções para evitar repetição de blocos de código
- Respeitar estilo (endentação, espaçamento, nomes de variáveis)
- Usar comentários (nem demais, nem de menos)
- … (outros que se fizerem necessários)

[1)]

Se dois itens têm a mesma prioridade, devem ser inseridos e retirados em ordem FIFO.

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

Permanent link:
https://wiki.inf.ufpr.br/maziero/doku.php?id=prog1:tad_fprio

Last update: **2024/10/31 19:38**