

Laboratório 0: Filas

Entrega até sábado, 9/3, às 23h

Neste laboratório, iremos realizar a implementação de uma fila, utilizando as estratégias de lista encadeada e array. Você pode verificar a implementação de uma fila com lista em **filalistasp.c**, porém essa implementação não possui um ponteiro para o último nó da fila. Dessa forma, para realizar uma inserção, todos os valores da fila precisam ser percorridos.

Para testar o programa, utilize o arquivo de teste **teste.c** e a interface **fila.h**. Também baixe os arquivos **processo.c** e **processo.h**. Caso esteja utilizando o compilador gcc, pode gerar o executável através da seguinte linha de comando:

```
gcc -o teste teste.c filalistasp.c processo.c
```

ps: o uso do argumento -o define o nome do executável gerado

O programa em questão “simula” (com muitas aspas) o uso de uma fila para um conjunto de processos. Em um laço de repetição, um dos processos é retirado para execução por um certo tempo. Se o tempo utilizado não for o suficiente, o processo é reinserido na fila.

Para cada uma das questões abaixo, baixe o arquivo correto e implemente as funções com o comentário “// COMPLETAR”.

1. (3.0) Termine a implementação de uma fila utilizando lista encadeada do módulo **filalista.c**. Nessa implementação, considere que a estrutura possui 2 ponteiros: um para o primeiro nó e um para o último nó da lista. Para testar, troque o arquivo **filalistasp.c** pelo arquivo **filalista.c**, pois esse arquivo implementa as mesmas funções declaradas na interface **fila.h**.

```
gcc -o teste teste.c filalista.c processo.c
```

2. (7.0) Termine a implementação do módulo **filavetor.c**, que implementa uma fila circular utilizando um vetor. Inicialmente, não se preocupe com a questão de redimensionamento, pois o teste foi implementado considerando uma fila com, no máximo, 8 valores. Para testar, use o módulo **filavetor.c** ao gerar o executável:

```
gcc -o teste teste.c filavetor.c processo.c
```

3. (1.0 Bônus) Considerando a implementação utilizando um vetor, como poderíamos implementar o redimensionamento da fila (apenas quando for necessário aumentar o tamanho do vetor)?

4. (1.0 Bônus) Faça a implementação de uma calculadora pós-fixa utilizando uma pilha. Você pode reutilizar a implementação da aula anterior, incluindo ela no módulo da calculadora. A interface de uma calculadora pós-fixa pode ser feita da seguinte forma:

```
typedef struct calc Calc;
```

```
Calc* calc_cria ();
```

```
void calc_operando (Calc* c, float v);
```

```
float calc_operador (Calc* c, char op); // retorna o valor da operação com operador 'op'
```

```
void calc_libera (Calc* c);
```

A calculadora pós-fixa deve funcionar da seguinte forma: se o método *calc_operando* for chamado, apenas adiciona o valor na pilha. Se *calc_operador* for chamado, deve-se retirar os 2 valores do topo da pilha, realizar a operação definida pelo operador, e inserir na pilha o resultado dessa operação. Você pode testar sua implementação com o caso da equação **(1-2)*(4+5)**, que em notação pós-fixa é correspondente a **1 2 - 4 5 + ***.

Crie um novo arquivo teste que insere uma lista de operandos e operadores na ordem pós-fixa para testar sua implementação. Caso um operador seja adicionado e a pilha for vazia, você pode considerar o valor *0.0f*.

Faça upload dos seus arquivos **filalista.c** e **filavetor.c** no EAD até dia 9 de março, às 23h. Caso tenha feito a questão 4, envie o arquivo de **teste** que você implementou e os arquivos de **interface** e **módulo** da **pilha** e **calculadora**. Indique na área de texto, que funções você implementou, e se o resultado verificado para cada uma está correto ou não. Lembre-se de fazer a entrega mesmo que não tenha chegado ao final do exercício.