**BLOCO DE CONSTRUÇÃO 1**

**Declarações, definições e protótipos**

#include "tx\_api.h”: Inclui um arquivo de cabeçalho chamado “**tx\_api.h**” que contém todos os equivalentes do sistema, estruturas de dados e protótipos de serviço.

#include <stdio.h>: Inclui um arquivo de cabeçalho padrão da linguagem “C/C++” chamado **“<stdio.h>”** que fornece funções relacionadas à entrada e saída padrão**,** permitindo assim imprimir mensagens na tela, ler dados, abrir e manipular arquivos.

#define STACK\_SIZE 1024: Define uma constante como tamanho da pilha na memória segura com o valor de 1024 bytes. A área de pilha do thread deve ser grande o suficiente para tratar seu aninhamento de chamada de função de pior caso possível e o uso de variável local.

#define BYTE\_POOL\_SIZE 9120: Define uma constante com tamanho de pool de bytes para 9120 bytes, não há limite para o número de pools de bytes de memória em um aplicativo. Os threads podem suspender em um pool até que a memória solicitada esteja disponível.

#define DISPLAY\_INTERVAL 5001: Define uma constante com intervalo de exibição para o tempo de 5001. É a variação do intervalo de tempo, até o próximo intervalo de tempo de uma determinada tarefa, para a exibição do experimento. Não há limite para o número de temporizadores em um aplicativo. No caso de vários temporizadores, respeita-se a ordem que foram ativados.

#define UPDATE\_INTERVAL 200: Define uma constante com intervalo de atualização para o tempo com valor de 200. Utilizada para controlar a frequência de atualizações de exibição.

Essa parte do cabeçalho definiu todas as bibliotecas e as constantes que serão utilizadas na aplicação. As bibliotecas padrão, já são pré-definidas na linguagem, ou pelo sistema operacional, bastando serem incluídas. As constantes foram definidas com um valor específico, onde cada vez que a constante for chamada, a mesma, será substituída por seu valor definido no cabeçalho.

**Defina os blocos de controle de objeto ThreadX**

TX\_THREAD Urgent\_thread; Define o bloco de controle de objeto da thread como “Urgent\_thread” ( THREAD DE URGÊNCIA)

TX\_THREAD Important\_thread; Define o bloco de controle de objeto da thread como “Important\_thread” ( THREAD DE IMPORTANCIA)

TX\_THREAD Routine\_thread; Define o bloco de controle de objeto da thread como “Routine\_thread” ( THREAD DE ROTINA)

TX\_MUTEX my\_mutex; Define o bloco de controle de objeto da thread MUTEX com o nome de “my\_mutex”. Um mutex, garante que sempre haja apenas um thread na seção crítica por vez.

TX\_BYTE\_POOL my\_byte\_pool; Define o bloco de controle de objeto da thread POOL DE BYTES DE MEMÓRIA com o nome de “my\_byte\_pool”. Para fornecer memória disponível das pilhas aos threads.

TX\_TIMER stats\_timer, update\_timer; Define o bloco de controle da thread TIMER (Temporizador da aplicação) com os nomes de “stats\_timer, update\_timer”.

Definido todos os blocos de controle que serão utilizados nas solicitações de prioridades da aplicação.

**Definir variáveis ​​para informações de desempenho do THREAD DE ROTINA.**

ULONG resumptions\_Routine; ULONG (“Unsigned Long” – dado longo sem sinal) é um tipo de dado de chamada de serviço, utilizado para definir “resumptions\_Routine”, um ponteiro para o destino do número de retomadas desse thread na aplicação de rotina.

ULONG suspensions\_Routine; ULONG (“Unsigned Long” – dado longo sem sinal) é um tipo de dado de chamada de serviço, utilizado para definir “suspensions\_Routine”, um ponteiro para o destino do número de suspensões desse thread nas ações de suspensão de rotina.

ULONG solicited\_preemptions\_Routine; ULONG (“Unsigned Long” – dado longo sem sinal) é um tipo de dado de chamada de serviço, utilizado para definir “solicited\_preemptions\_Routine”, uma solicitação de um ponteiro, para o destino do número de preempções, como resultado de uma chamada de serviço de API do ThreadX feita por esse thread de rotina.

\*PREEMPÇÃO: Refere-se à capacidade do sistema operacional de interromper a execução de uma tarefa em execução e permitir que outra tarefa com uma prioridade mais alta seja executada. Isso é importante em sistemas de tempo real, onde tarefas críticas podem precisar ser executadas imediatamente, mesmo que outra tarefa esteja em andamento.

Definido todas as variáveis que serão utilizadas para buscar informações de desempenho das prioridades de rotina, com a utilização de ponteiros.

**Definir variáveis ​​para informações de desempenho das thread de THREAD DE URGÊNCIA.**

ULONG resumptions\_Urgent; ULONG (“Unsigned Long” – dado longo sem sinal) é um tipo de dado de chamada de serviço, utilizado para definir “resumptions\_Urgent”, um ponteiro para o destino do número de retomadas desse thread na aplicação de urgência.

ULONG suspensions\_Urgent; ULONG (“Unsigned Long” – dado longo sem sinal) é um tipo de dado de chamada de serviço, utilizado para definir “suspensions\_Urgent”, um ponteiro para o destino do número de suspensões desse thread nas ações de suspensão de urgência.

Definido todas as variáveis que serão utilizadas para buscar informações de desempenho das prioridades de urgência, com a utilização de ponteiros.

**Definir variáveis ​​para uso com informações de desempenho MUTEX.**

ULONG mutex\_puts; ULONG (“Unsigned Long” – dado longo sem sinal) é um tipo de dado de chamada de serviço, utilizado para definir “mutex\_puts”, um ponteiro para o destino do número de solicitações put executadas nesse mutex.

\*TX\_MUTEX\_PUTS: A operação put libera um mutex previamente obtido. Para um thread liberar um mutex, o número de operações put deve ser igual ao número de operações get anteriores.

ULONG mutex\_gets; ULONG (“Unsigned Long” – dado longo sem sinal) é um tipo de dado de chamada de serviço, utilizado para definir “mutex\_gets”, um ponteiro para o destino do número de solicitações de obtenção executadas nesse mutex.

\*TX\_MUTEX\_GET: A operação get obtém um mutex que não pertence a outro thread.

\*MUTEX: É basicamente um semáforo binário, o que significa que apenas um thread pode ter um mutex por vez. Além disso, em um mutex de propriedade, o mesmo thread pode executar uma operação get com mutex bem-sucedida várias vezes, 4.294.967.295 vezes para ser exato.

Definido todas as variáveis que serão utilizadas para buscar informações de desempenho dos mutexes, com a utilização de ponteiros.

**Definir variável para análise de tempo**

ULONG current\_time; ULONG (“Unsigned Long” – dado longo sem sinal) é um tipo de dado de chamada de serviço, utilizado para definir “current\_time”, uma variável utilizada para sincronização de tarefas, controlando os eventos da aplicação com base no tempo fornecido pelo sistema.

**Definição dos protótipos:**

void Urgent\_thread\_entry(ULONG thread\_input); Cria uma função que não retorna um valor de entrada, recebendo o nome de “Urgent\_thread\_entry” (entrada da thread de urgência) e recebe como parâmetros da função a variável “thread\_input” (entrada da thread) que é do tipo ULONG.

void Important\_thread\_entry(ULONG thread\_input); Cria uma função que não retorna um valor de entrada, recebendo o nome de “Important\_thread\_entry” (entrada da thread de importância) e recebe como parâmetros da função a variável “thread\_input” (entrada da thread) que é do tipo ULONG.

void Routine\_thread\_entry(ULONG thread\_input); Cria uma função que não retorna um valor de entrada, recebendo o nome de “Routine\_thread\_entry” (entrada da thread de rotina) e recebe como parâmetros da função variável “thread\_input” (entrada da thread) que é do tipo ULONG.

void print\_stats(ULONG), print\_update(ULONG); Função criada para declarar 2 argumentos, sendo uma para impressão de informações e a outra função para impressão de atualizações.

Cria uma função que não retorna um valor de entrada, recebendo o nome de “print\_stats”, sendo utilizada para impressão de informações, com base no valor passado no argumento “ULONG”.

Cria uma função que não retorna um valor de entrada, recebendo o nome de “print\_update”, sendo utilizada para impressão de atualizações, com base no valor passado no argumento “ULONG”.

Definido todas as funções utilizadas no decorrer da execução da aplicação, definidas para a criaçao de informações e atualizações das threads de prioridades, variável para análise de tempo e impressão dos dados.