Sistemas Embarcados

Trabalho 1 – Escalonador LLTF

Aluno: Bruno Bavaresco Zaffari

Professor: Sérgio Johann Filho

Configurações opcionais:

```
#include <ucx.h>
#define STOP_TIME 33
#define KILL_IF_DEADLINE_MISS 0
//#define DEBUG
/* application tasks */
```

Caso queira limitar o tempo TICKS ou desabilitar o “kill”.

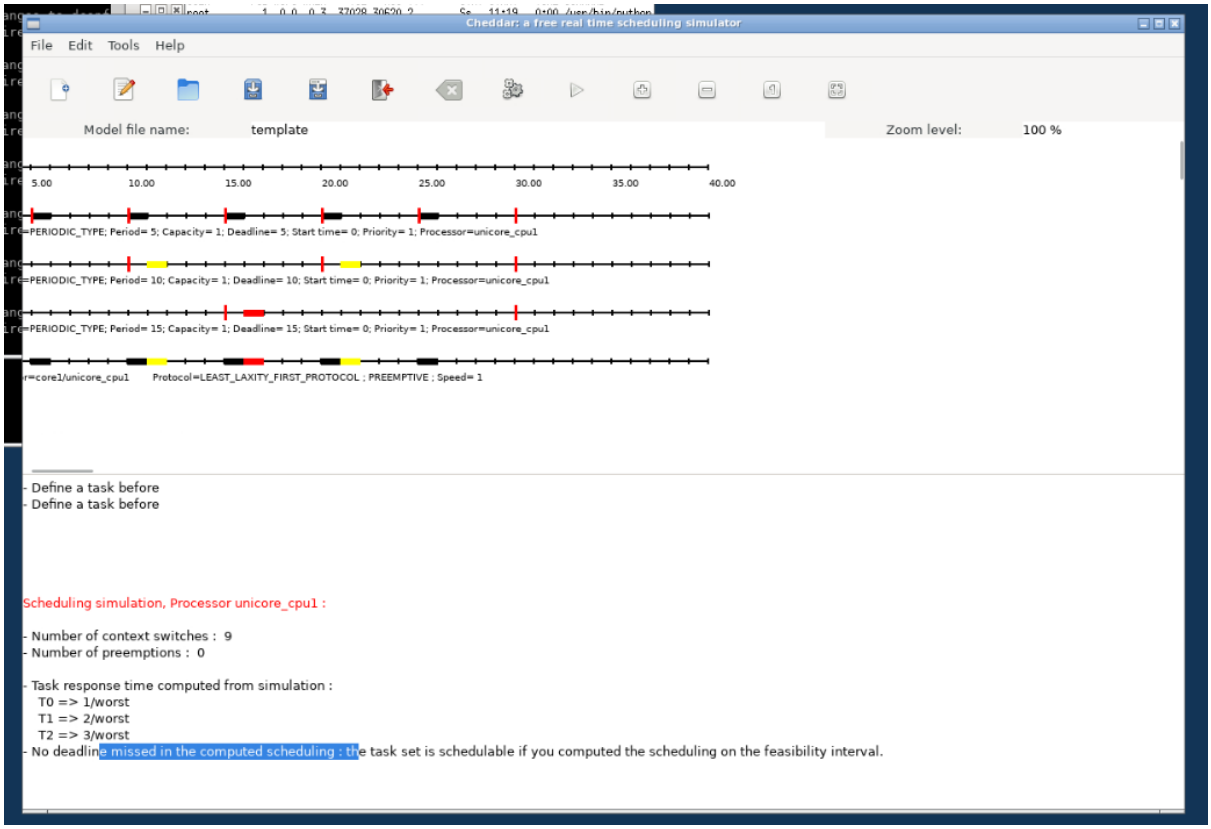
Definição das tarefas foi feita dessa forma,

```
s/app/rtsched.c
ain(void)
t */
struct lstf_parameter parameters[4] = {
    {.computation= 0x03, .period = 0x05, .deadline = 0x05, .slack = 0x02, .remaining = 0x03},
    {.computation= 0x05, .period = 0x0A, .deadline = 0x0A, .slack = 0x05, .remaining = 0x05},
    {.computation= 0x08, .period = 0x0f, .deadline = 0x0f, .slack = 0x07, .remaining = 0x08},
    {.computation= 0x01, .period = 0x0C, .deadline = 0x0C, .slack = 0x0B, .remaining = 0x01},
};

int ERR = -20;
/* tasks of the set */
kcb->rt_sched = lstf_sched;// dispatcher

// IDLE ========================================================
/* Init Task*/
// Idle - ID = 0
uint16_t idT_Idle = ucx_task_spawn(task_idle, DEFAULT_STACK_SIZE);
ucx_task_priority(idT_Idle, TASK_IDLE_PRIO);
//==============================================================

// NORMAL ======================================================
/* Init Task */
//0 - ID = 1
uint16_t idT0 = ucx_task_spawn(task0, DEFAULT_STACK_SIZE);
printf("Tarefa: 0 ID: %d\n",  (int)idT0);

//1 - ID = 2
uint16_t idT1 = ucx_task_spawn(task1, DEFAULT_STACK_SIZE);
printf("Tarefa: 1 ID: %d\n",  (int)idT1);

//2 - ID = 3
uint16_t idT2 = ucx_task_spawn(task2, DEFAULT_STACK_SIZE);
printf("Tarefa: 2 ID: %d\n",  (int)idT2);

//3 - ID = 4
uint16_t idT3 = ucx_task_spawn(task3, DEFAULT_STACK_SIZE);
printf("Tarefa: 3 ID: %d\n",  (int)idT3);
// ============================================================
```

Ao transformar uma tarefa em uma tarefa de tempo real — e após definir seus parâmetros — é fundamental descometar a linha onde a prioridade de tempo real foi atribuída, conforme ilustrado no exemplo abaixo:

```
// RT ==========================================================
/* Setup our custom scheduler and set RT priorities to three*/
//0 - ID = 1
ERR = ucx_task_rt_priority(idT0, &parameters[0]);
printf("ERR T0 %d\n", (int)ERR);
printf("Endereço do parameters[0] %d\n", (int)&parameters[0]);

//1 - ID = 2
ERR = ucx_task_rt_priority(idT1, &parameters[1]);
printf("ERR T1 %d\n", (int)ERR);
printf("Endereço do parameters[1] %d\n", (int)&parameters[1]);

//2 - ID = 3
//ERR = ucx_task_rt_priority(idT2, &parameters[2]);
//printf("ERR T2 %d\n", (int)ERR);
//printf("Endereço do parameters[2] %d\n", (int)&parameters[2]);

//3 - ID = 4
//ERR = ucx_task_rt_priority(idT3, &parameters[3]);
//printf("ERR T3 %d\n", (int)ERR);
//printf("Endereço do parameters[3] %d\n", (int)&parameters[3]);

// start UCX/OS, preemptive mode
return 1;
```

Teste Escalonável, Cheddar:



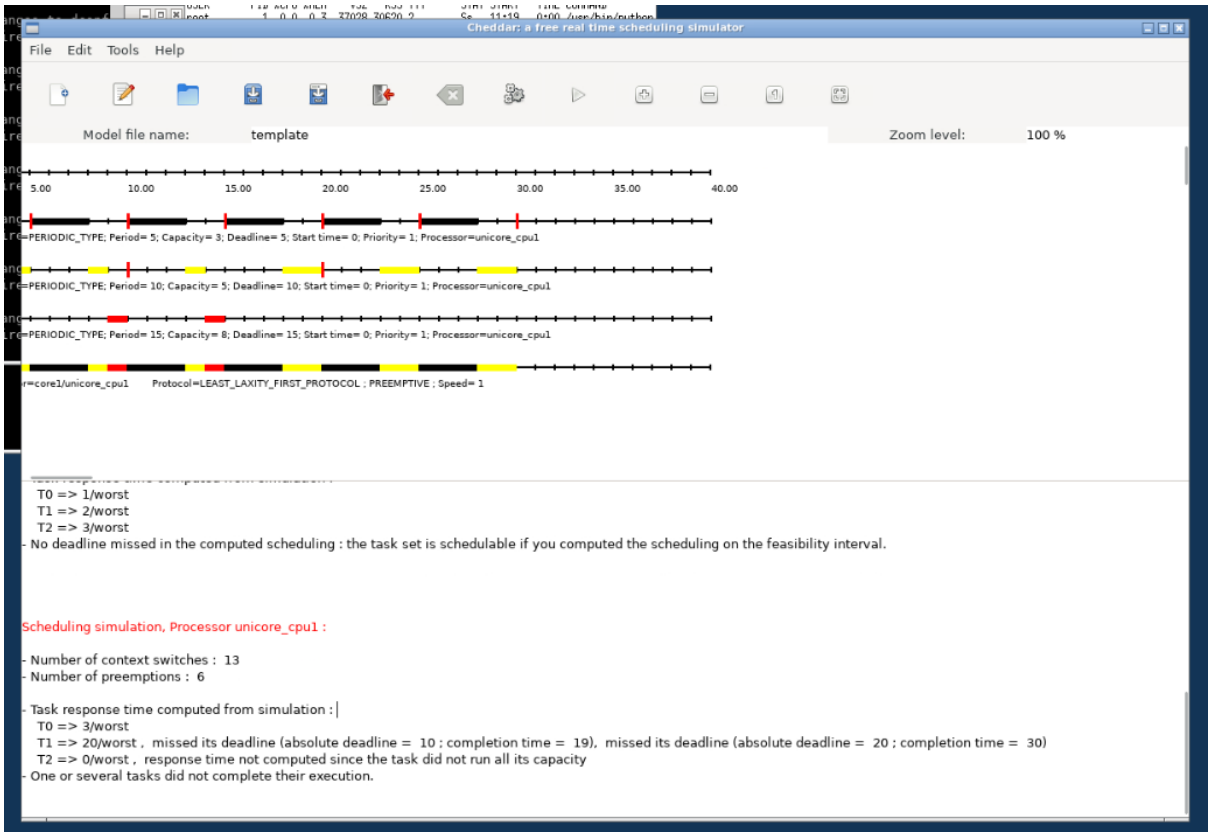
Teste Escalonável, escalonador de Tempo Real:

```
task->rt_prio 1073774554
ERR T2 0
Endereço do parameters[2] 1073774554
=TEMPO TICKS 00 = -RUNNING "taks0" <----------------------------
|TASK ID: 1 | Comp.: 1 | Per.: 5 | Dl.: 5 | Slack: 4 | Remaining: 0 |
=TEMPO TICKS 01 = -RUNNING "taks1" <----------------------------
|TASK ID: 2 | Comp.: 1 | Per.: 10 | Dl.: 10 | Slack: 8 | Remaining: 0 |
=TEMPO TICKS 02 = -RUNNING "task2" <----------------------------
|TASK ID: 3 | Comp.: 2 | Per.: 15 | Dl.: 15 | Slack: 11 | Remaining: 1 |
=TEMPO TICKS 03 = -RUNNING "task2" <----------------------------
|TASK ID: 3 | Comp.: 2 | Per.: 15 | Dl.: 15 | Slack: 11 | Remaining: 0 |
=TEMPO TICKS 04 = -RUNNING "task_idle" <----------------------------
TASK ID: 0
=TEMPO TICKS 05 = -RUNNING "taks0" <----------------------------
|TASK ID: 1 | Comp.: 1 | Per.: 5 | Dl.: 5 | Slack: 4 | Remaining: 0 |
=TEMPO TICKS 06 = -RUNNING "task_idle" <----------------------------
TASK ID: 0
=TEMPO TICKS 07 = -RUNNING "task_idle" <----------------------------
TASK ID: 0
=TEMPO TICKS 08 = -RUNNING "task_idle" <----------------------------
TASK ID: 0
=TEMPO TICKS 09 = -RUNNING "task_idle" <----------------------------
TASK ID: 0
=TEMPO TICKS 10 = -RUNNING "taks0" <----------------------------
|TASK ID: 1 | Comp.: 1 | Per.: 5 | Dl.: 5 | Slack: 4 | Remaining: 0 |
=TEMPO TICKS 11 = -RUNNING "taks1" <----------------------------
|TASK ID: 2 | Comp.: 1 | Per.: 10 | Dl.: 10 | Slack: 8 | Remaining: 0 |
```

```
=TEMPO TICKS 11 = -RUNNING "taks1" <----------------------------
|TASK ID: 2 | Comp.: 1 | Per.: 10 | Dl.: 10 | Slack: 8 | Remaining: 0 |
=TEMPO TICKS 12 = -RUNNING "task_idle" <----------------------------
TASK ID: 0
=TEMPO TICKS 13 = -RUNNING "task_idle" <----------------------------
TASK ID: 0
=TEMPO TICKS 14 = -RUNNING "task_idle" <----------------------------
TASK ID: 0
=TEMPO TICKS 15 = -RUNNING "taks0" <----------------------------
|TASK ID: 1 | Comp.: 1 | Per.: 5 | Dl.: 5 | Slack: 4 | Remaining: 0 |
=TEMPO TICKS 16 = -RUNNING "task2" <----------------------------
|TASK ID: 3 | Comp.: 2 | Per.: 15 | Dl.: 15 | Slack: 12 | Remaining: 1 |
=TEMPO TICKS 17 = -RUNNING "task2" <----------------------------
|TASK ID: 3 | Comp.: 2 | Per.: 15 | Dl.: 15 | Slack: 12 | Remaining: 0 |
=TEMPO TICKS 18 = -RUNNING "task_idle" <----------------------------
TASK ID: 0
=TEMPO TICKS 19 = -RUNNING "task_idle" <----------------------------
TASK ID: 0
=TEMPO TICKS 20 = -RUNNING "taks0" <----------------------------
|TASK ID: 1 | Comp.: 1 | Per.: 5 | Dl.: 5 | Slack: 4 | Remaining: 0 |
=TEMPO TICKS 21 = -RUNNING "taks1" <----------------------------
|TASK ID: 2 | Comp.: 1 | Per.: 10 | Dl.: 10 | Slack: 8 | Remaining: 0 |
=TEMPO TICKS 22 = -RUNNING "task_idle" <----------------------------
TASK ID: 0
=TEMPO TICKS 23 = -RUNNING "task_idle" <----------------------------
TASK ID: 0
=TEMPO TICKS 24 = -RUNNING "task_idle" <----------------------------
TASK ID: 0
=TEMPO TICKS 25 = -RUNNING "taks0" <----------------------------
|TASK ID: 1 | Comp.: 1 | Per.: 5 | Dl.: 5 | Slack: 4 | Remaining: 0 |
=TEMPO TICKS 26 = -RUNNING "task_idle" <----------------------------
TASK ID: 0
```

```
}

int32_t app_main(void)
{
    /* uint8_t */
    struct lstf_parameter parameters[4] = {
        {.computation= 0x01, .period = 0x05, .deadline = 0x05, .slack = 0x04, .remaining = 0x01},
        {.computation= 0x01, .period = 0x0A, .deadline = 0x0A, .slack = 0x09, .remaining = 0x01},
        {.computation= 0x02, .period = 0x0f, .deadline = 0x0f, .slack = 0x0E, .remaining = 0x02},
        //{.computation= 0x01, .period = 0x0C, .deadline = 0x0C, .slack = 0x0B, .remaining = 0x01},
    };

    int ERR = -20;
    /* tasks of the set */
    kcb->rt_sched = lstf_sched;// dispatcher

    // IDLE ============================================================
    /* Init Task*/
    // Idle - ID = 0
    uint16_t idT_Idle = ucx_task_spawn(task_idle, DEFAULT_STACK_SIZE);
    ucx_task_priority(idT_Idle, TASK_IDLE_PRIO);
    //==================================================================

    // NORMAL ==========================================================
    /* Init Task */
```

Teste Não-escalonável, Cheddar:



Teste Não-escalonável, escalonador Tempo Real:

```
------------------------------------------------------------------
=TEMPO TICKS 00 =
--------------------------------------------------
CONFERNDO:||Slack-> 02|| (Dl.= 05 -Rem.= 03 - T.R.= 00) |(Comp. 03 )|(T.R.= 00 = 0 mod 5)|
--------------------------------------------------
CONFERNDO:||Slack-> 05|| (Dl.= 10 -Rem.= 05 - T.R.= 00) |(Comp. 05 )|(T.R.= 00 = 0 mod 10)|
--------------------------------------------------
CONFERNDO:||Slack-> 07|| (Dl.= 15 -Rem.= 08 - T.R.= 00) |(Comp. 08 )|(T.R.= 00 = 0 mod 15)|

------------------------------------------------------------------
-RUNNING "taks0" <--------------------------------------------------
|TASK ID: 1 | Comp.: 3 | Per.: 5 | Dl.: 5 | Slack: 2 | Remaining: 2 |

------------------------------------------------------------------
=TEMPO TICKS 01 =
--------------------------------------------------
CONFERNDO:||Slack-> 02|| (Dl.= 05 -Rem.= 02 - T.R.= 01) |(Comp. 03 )|(T.R.= 01 = 1 mod 5)|
--------------------------------------------------
CONFERNDO:||Slack-> 04|| (Dl.= 10 -Rem.= 05 - T.R.= 01) |(Comp. 05 )|(T.R.= 01 = 1 mod 10)|
--------------------------------------------------
NFERNDO:||Slack-> 06|| (Dl.= 15 -Rem.= 08 - T.R.= 01) |(Comp. 08 )|(T.R.= 01 = 1 mod 15)|

------------------------------------------------------------------
-RUNNING "taks0" <--------------------------------------------------
|TASK ID: 1 | Comp.: 3 | Per.: 5 | Dl.: 5 | Slack: 2 | Remaining: 1 |

------------------------------------------------------------------
=TEMPO TICKS 02 =
--------------------------------------------------
CONFERNDO:||Slack-> 02|| (Dl.= 05 -Rem.= 01 - T.R.= 02) |(Comp. 03 )|(T.R.= 02 = 2 mod 5)|
--------------------------------------------------
CONFERNDO:||Slack-> 03|| (Dl.= 10 -Rem.= 05 - T.R.= 02) |(Comp. 05 )|(T.R.= 02 = 2 mod 10)|
--------------------------------------------------
CONFERNDO:||Slack-> 05|| (Dl.= 15 -Rem.= 08 - T.R.= 02) |(Comp. 08 )|(T.R.= 02 = 2 mod 15)|

------------------------------------------------------------------
-RUNNING "taks0" <--------------------------------------------------
|TASK ID: 1 | Comp.: 3 | Per.: 5 | Dl.: 5 | Slack: 2 | Remaining: 0 |

------------------------------------------------------------------
=TEMPO TICKS 03 =
--------------------------------------------------
----->"FINISHED" TASK_ID 1
CONFERNDO:||Slack-> 02|| (Dl.= 05 -Rem.= 00 - T.R.= 03) |(Comp. 03 )|(T.R.= 03 = 3 mod 5)|
--------------------------------------------------
CONFERNDO:||Slack-> 02|| (Dl.= 10 -Rem.= 05 - T.R.= 03) |(Comp. 05 )|(T.R.= 03 = 3 mod 10)|
--------------------------------------------------
CONFERNDO:||Slack-> 04|| (Dl.= 15 -Rem.= 08 - T.R.= 03) |(Comp. 08 )|(T.R.= 03 = 3 mod 15)|

------------------------------------------------------------------
-RUNNING "taks1" <--------------------------------------------------
|TASK ID: 2 | Comp.: 5 | Per.: 10 | Dl.: 10 | Slack: 2 | Remaining: 4 |
```

```
------------------------------------------------------------------
=TEMPO TICKS 05 =
--------------------------------------------------
----->"FINISHED" TASK_ID 1
----->"RESET" TASK_ID 1
CONFERNDO:||Slack-> 02|| (Dl.= 05 -Rem.= 03 - T.R.= 00) |(Comp. 03 )|(T.R.= 00 = 5 mod 5)|
--------------------------------------------------
CONFERNDO:||Slack-> 02|| (Dl.= 10 -Rem.= 03 - T.R.= 05) |(Comp. 05 )|(T.R.= 05 = 5 mod 10)|
--------------------------------------------------
CONFERNDO:||Slack-> 02|| (Dl.= 15 -Rem.= 08 - T.R.= 05) |(Comp. 08 )|(T.R.= 05 = 5 mod 15)|

------------------------------------------------------------------
-RUNNING "taks0" <--------------------------------------------------
|TASK ID: 1 | Comp.: 3 | Per.: 5 | Dl.: 5 | Slack: 2 | Remaining: 2 |

------------------------------------------------------------------
=TEMPO TICKS 06 =
--------------------------------------------------
NFERNDO:||Slack-> 02|| (Dl.= 05 -Rem.= 02 - T.R.= 01) |(Comp. 03 )|(T.R.= 01 = 6 mod 5)|
--------------------------------------------------
NFERNDO:||Slack-> 01|| (Dl.= 10 -Rem.= 03 - T.R.= 06) |(Comp. 05 )|(T.R.= 06 = 6 mod 10)|
--------------------------------------------------
CONFERNDO:||Slack-> 01|| (Dl.= 15 -Rem.= 08 - T.R.= 06) |(Comp. 08 )|(T.R.= 06 = 6 mod 15)|

------------------------------------------------------------------
-RUNNING "taks1" <--------------------------------------------------
|TASK ID: 2 | Comp.: 5 | Per.: 10 | Dl.: 10 | Slack: 1 | Remaining: 2 |

------------------------------------------------------------------
=TEMPO TICKS 07 =
--------------------------------------------------
CONFERNDO:||Slack-> 01|| (Dl.= 05 -Rem.= 02 - T.R.= 02) |(Comp. 03 )|(T.R.= 02 = 7 mod 5)|
--------------------------------------------------
CONFERNDO:||Slack-> 01|| (Dl.= 10 -Rem.= 02 - T.R.= 07) |(Comp. 05 )|(T.R.= 07 = 7 mod 10)|
--------------------------------------------------
CONFERNDO:||Slack-> 00|| (Dl.= 15 -Rem.= 08 - T.R.= 07) |(Comp. 08 )|(T.R.= 07 = 7 mod 15)|

------------------------------------------------------------------
-RUNNING "task2" <--------------------------------------------------
|TASK ID: 3 | Comp.: 8 | Per.: 15 | Dl.: 15 | Slack: 0 | Remaining: 7 |

------------------------------------------------------------------
=TEMPO TICKS 08 =
--------------------------------------------------
CONFERNDO:||Slack-> 00|| (Dl.= 05 -Rem.= 02 - T.R.= 03) |(Comp. 03 )|(T.R.= 03 = 8 mod 5)|
--------------------------------------------------
CONFERNDO:||Slack-> 00|| (Dl.= 10 -Rem.= 02 - T.R.= 08) |(Comp. 05 )|(T.R.= 08 = 8 mod 10)|
--------------------------------------------------
CONFERNDO:||Slack-> 00|| (Dl.= 15 -Rem.= 07 - T.R.= 08) |(Comp. 08 )|(T.R.= 08 = 8 mod 15)|

------------------------------------------------------------------
-RUNNING "taks0" <--------------------------------------------------
|TASK ID: 1 | Comp.: 3 | Per.: 5 | Dl.: 5 | Slack: 0 | Remaining: 1 |
```

```c
int32_t app_main(void)
{
    /* uint8_t */
    struct lstf_parameter parameters[4] = {
        {.computation= 0x03, .period = 0x05, .deadline = 0x05, .slack = 0x02, .remaining = 0x03},
        {.computation= 0x05, .period = 0x0A, .deadline = 0x0A, .slack = 0x05, .remaining = 0x05},
        {.computation= 0x08, .period = 0x0f, .deadline = 0x0f, .slack = 0x07, .remaining = 0x08},
        //{.computation= 0x01, .period = 0x0C, .deadline = 0x0C, .slack = 0x0B, .remaining = 0x01},
    };

    int ERR = -20;
    /* tasks of the set */
    kcb->rt_sched = lstf_sched;// dispatcher

    // IDLE ============================================================
    /* Init Task*/
    // Idle - ID = 0
    uint16_t idT_Idle = ucx_task_spawn(task_idle, DEFAULT_STACK_SIZE);
    ucx_task_priority(idT_Idle, TASK_IDLE_PRIO);
    //==================================================================

    // NORMAL ==========================================================
    /* Init Task */
    //0 - ID = 1
    uint16_t idT0 = ucx_task_spawn(task0, DEFAULT_STACK_SIZE);
    printf("Tarefa: 0 ID: %d\n",  (int)idT0);

    //1 - ID = 2
    uint16_t idT1 = ucx_task_spawn(task1, DEFAULT_STACK_SIZE);
    printf("Tarefa: 1 ID: %d\n",  (int)idT1);

    //2 - ID = 3
    uint16_t idT2 = ucx_task_spawn(task2, DEFAULT_STACK_SIZE);
    printf("Tarefa: 2 ID: %d\n",  (int)idT2);

    //3 - ID = 4
    //uint16_t idT3 = ucx_task_spawn(task3, DEFAULT_STACK_SIZE);
    //printf("Tarefa: 3 ID: %d\n",  (int)idT3);
    // ================================================================
```

Nota: Apesar de ter divergido no “**TEMPO TICKS 6**”. A princípio tem-se o debug pra confirmar e o escalonador feito com UCX-OS escalonou certamente a com menor Slack, a princípio estaria certo. Sendo um parâmetro divergente no Cheddar q tenha dado a troca de Task0 para Tesk1. Ambas acusaram corretamente que era inviável o escalonamento das tasks. Caso fosse mudado o marco $ **KILL_IF_DEADLINE_MISS**, para outro número, a não ser zero, ela teria parado a execução no primeiro DEADLINE_MISS.

Tarefas Aperiódicas e Periódicas:

Tarefas Periódicas: **“task0” (P:5; D:5; C:1)** -> ID: 01 e “task1” (P:10; D:10; C:1) -> ID: 02;

Tarefas Aperiódicas: “task2” -> ID: 03 e “task3” -> ID: 04;

NOTA: Apesar de ter a “tarefa_idle” nunca será executada, nesse caso;

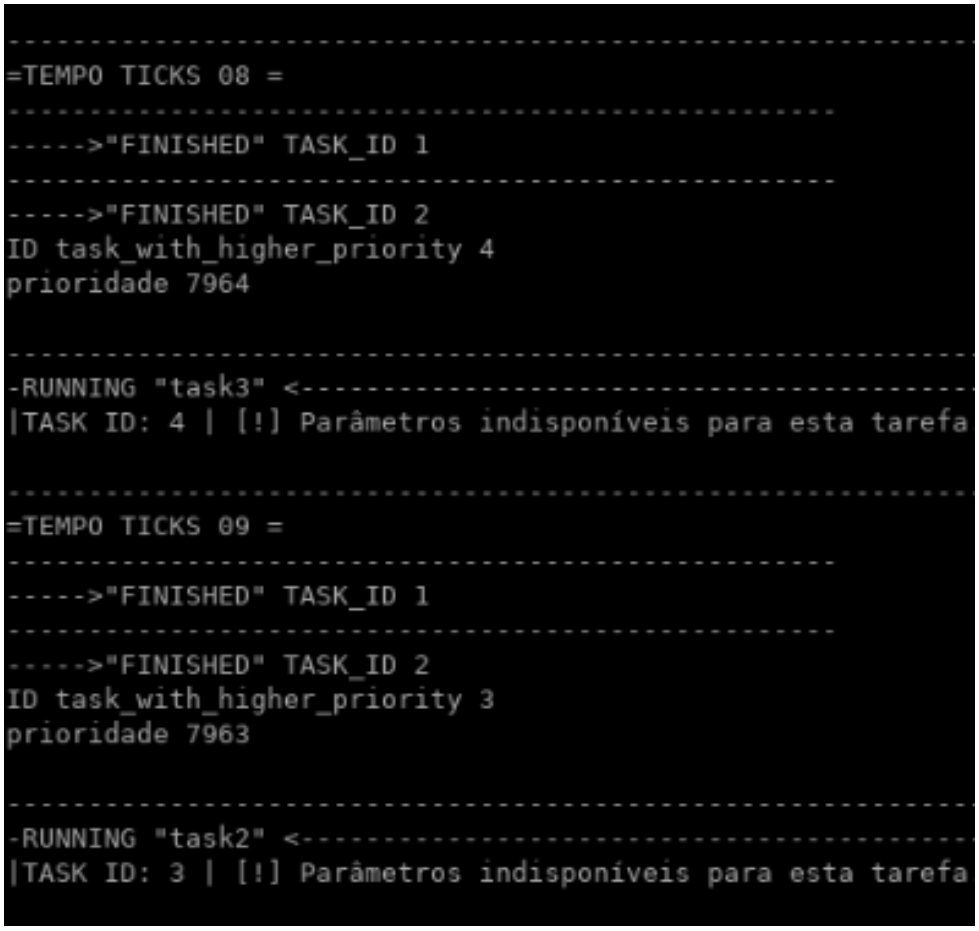
```c
int32_t app_main(void)
{
    /* uint8_t */
    struct lstf_parameter parameters[4] = {
        {.computation= 0x01, .period = 0x05, .deadline = 0x05, .slack = 0x04, .remaining = 0x01},
        {.computation= 0x01, .period = 0x0A, .deadline = 0x0A, .slack = 0x09, .remaining = 0x01},
        //{.computation= 0x08, .period = 0x0f, .deadline = 0x0f, .slack = 0x07, .remaining = 0x08},
        //{.computation= 0x01, .period = 0x0C, .deadline = 0x0C, .slack = 0x0B, .remaining = 0x01},
    };
```

```
// RT ===================================================================
/* Setup our custom scheduler and set RT priorities to three*/
//0 - ID = 1
ERR = ucx_task_rt_priority(idT0, &parameters[0]);
printf("ERR T0 %d\n", (int)ERR);
printf("Endereço do parameters[0] %d\n", (int)&parameters[0]);

//1 - ID = 2
ERR = ucx_task_rt_priority(idT1, &parameters[1]);
printf("ERR T1 %d\n", (int)ERR);
printf("Endereço do parameters[1] %d\n", (int)&parameters[1]);

//2 - ID = 3
//ERR = ucx_task_rt_priority(idT2, &parameters[2]);
//printf("ERR T2 %d\n", (int)ERR);
//printf("Endereço do parameters[2] %d\n", (int)&parameters[2]);

//3 - ID = 4
//ERR = ucx_task_rt_priority(idT3, &parameters[3]);
//printf("ERR T3 %d\n", (int)ERR);
//printf("Endereço do parameters[3] %d\n", (int)&parameters[3]);

// start UCX/OS, preemptive mode
return 1;
```

Teste funcional das tarefas Aperiódicas e Periódicas:

```
------------------------------------------------------------------------
=TEMPO TICKS 00 =
------------------------------------------------
------------------------------------------------

------------------------------------------------------------------------
-RUNNING "taks0" <------------------------------------------------------
|TASK ID: 1 | Comp.: 1 | Per.: 5 | Dl.: 5 | Slack: 4 | Remaining: 0 |

------------------------------------------------------------------------
=TEMPO TICKS 01 =
------------------------------------------------
----->"FINISHED" TASK_ID 1
------------------------------------------------

------------------------------------------------------------------------
-RUNNING "taks1" <------------------------------------------------------
|TASK ID: 2 | Comp.: 1 | Per.: 10 | Dl.: 10 | Slack: 8 | Remaining: 0 |

------------------------------------------------------------------------
=TEMPO TICKS 02 =
------------------------------------------------
----->"FINISHED" TASK_ID 1
------------------------------------------------
----->"FINISHED" TASK_ID 2
ID task_with_higher_priority 3
prioridade 7966

------------------------------------------------------------------------
-RUNNING "task2" <------------------------------------------------------
|TASK ID: 3 | [!] Parâmetros indisponíveis para esta tarefa.

------------------------------------------------------------------------
=TEMPO TICKS 03 =
------------------------------------------------
----->"FINISHED" TASK_ID 1
------------------------------------------------
----->"FINISHED" TASK_ID 2
ID task_with_higher_priority 4
prioridade 7966

------------------------------------------------------------------------
-RUNNING "task3" <------------------------------------------------------
|TASK ID: 4 | [!] Parâmetros indisponíveis para esta tarefa.
```

```
------------------------------------------------------------------------
=TEMPO TICKS 04 =
------------------------------------------------
----->"FINISHED" TASK_ID 1
------------------------------------------------
----->"FINISHED" TASK_ID 2
ID task_with_higher_priority 3
prioridade 7965

------------------------------------------------------------------------
-RUNNING "task2" <------------------------------------------------------
|TASK ID: 3 | [!] Parâmetros indisponíveis para esta tarefa.

------------------------------------------------------------------------
=TEMPO TICKS 05 =
------------------------------------------------
----->"FINISHED" TASK_ID 1
----->"RESET" TASK_ID 1
------------------------------------------------
----->"FINISHED" TASK_ID 2

------------------------------------------------------------------------
-RUNNING "taks0" <------------------------------------------------------
|TASK ID: 1 | Comp.: 1 | Per.: 5 | Dl.: 5 | Slack: 4 | Remaining: 0 |

------------------------------------------------------------------------
=TEMPO TICKS 06 =
------------------------------------------------
----->"FINISHED" TASK_ID 1
------------------------------------------------
----->"FINISHED" TASK_ID 2
ID task_with_higher_priority 4
prioridade 7965

------------------------------------------------------------------------
-RUNNING "task3" <------------------------------------------------------
|TASK ID: 4 | [!] Parâmetros indisponíveis para esta tarefa.

------------------------------------------------------------------------
=TEMPO TICKS 07 =
------------------------------------------------
----->"FINISHED" TASK_ID 1
------------------------------------------------
----->"FINISHED" TASK_ID 2
ID task_with_higher_priority 3
prioridade 7964

------------------------------------------------------------------------
-RUNNING "task2" <------------------------------------------------------
|TASK ID: 3 | [!] Parâmetros indisponíveis para esta tarefa.
```



=TEMPO TICKS 08 =

----->"FINISHED" TASK_ID 1

----->"FINISHED" TASK_ID 2
ID task_with_higher_priority 4
prioridade 7964

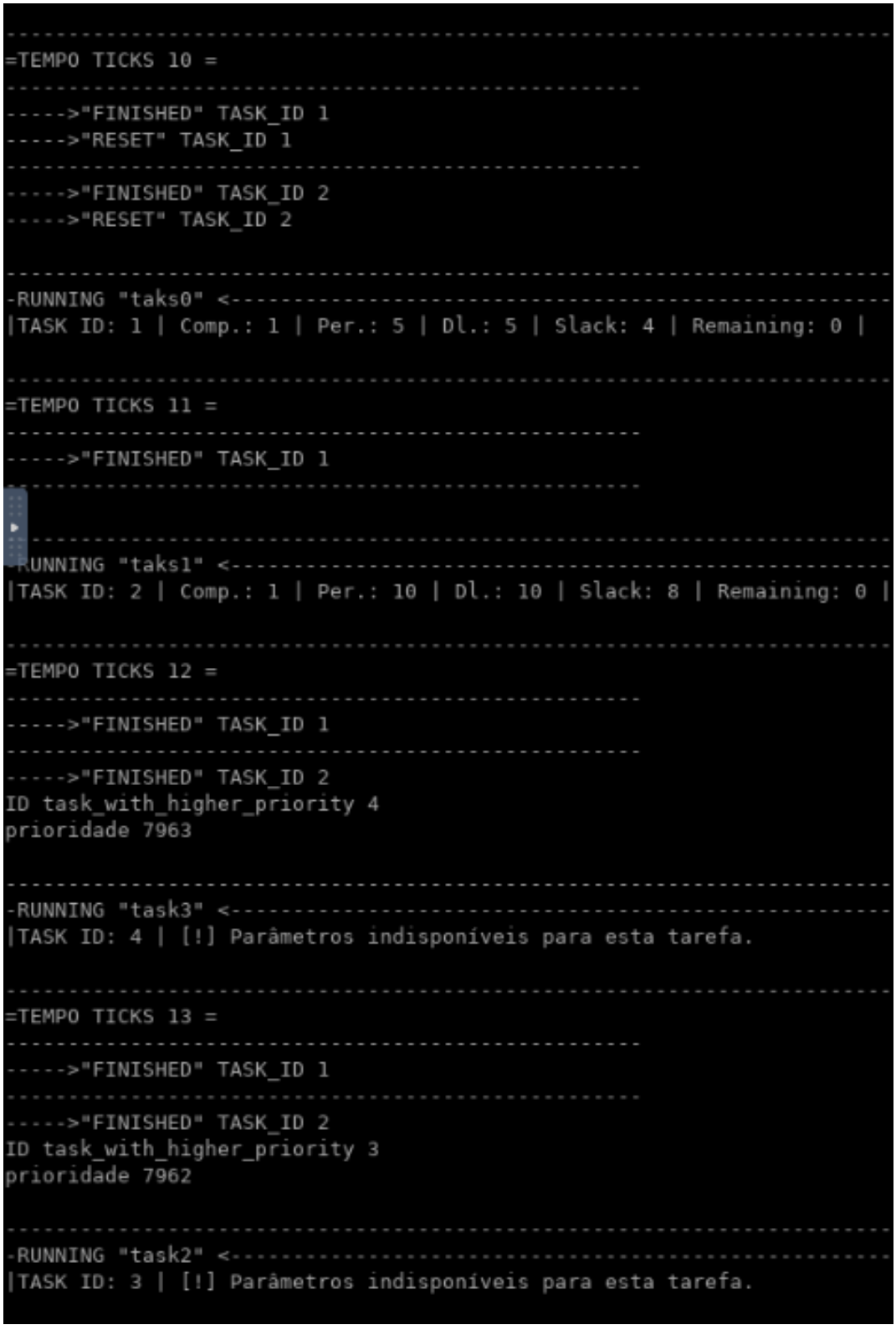
-RUNNING "task3" <----------
|TASK ID: 4 | [!] Parâmetros indisponíveis para esta tarefa

=TEMPO TICKS 09 =

----->"FINISHED" TASK_ID 1

----->"FINISHED" TASK_ID 2
ID task_with_higher_priority 3
prioridade 7963

-RUNNING "task2" <----------
|TASK ID: 3 | [!] Parâmetros indisponíveis para esta tarefa



Escalonador LLTF:

Para passagem do tempo correta foi entrada crítica para entrar no DISPATCHER:

```
void krnl_dispatcher(void)
{
    kcb->ticks++;
    CRITICAL_ENTER();
    _dispatch();
    CRITICAL_LEAVE();
}       You, 22 hours ago • final …
```

Para funcionar corretamente precisa adicionar no main uma tarefa IDLE:

```
// IDLE ============================================================
/* Init Task*/
// Idle - ID = 0
uint16_t idT_Idle = ucx_task_spawn(task_idle, DEFAULT_STACK_SIZE);
ucx_task_priority(idT_Idle, TASK_IDLE_PRIO);
//==================================================================
```

<u>Escalonador em si:</u>

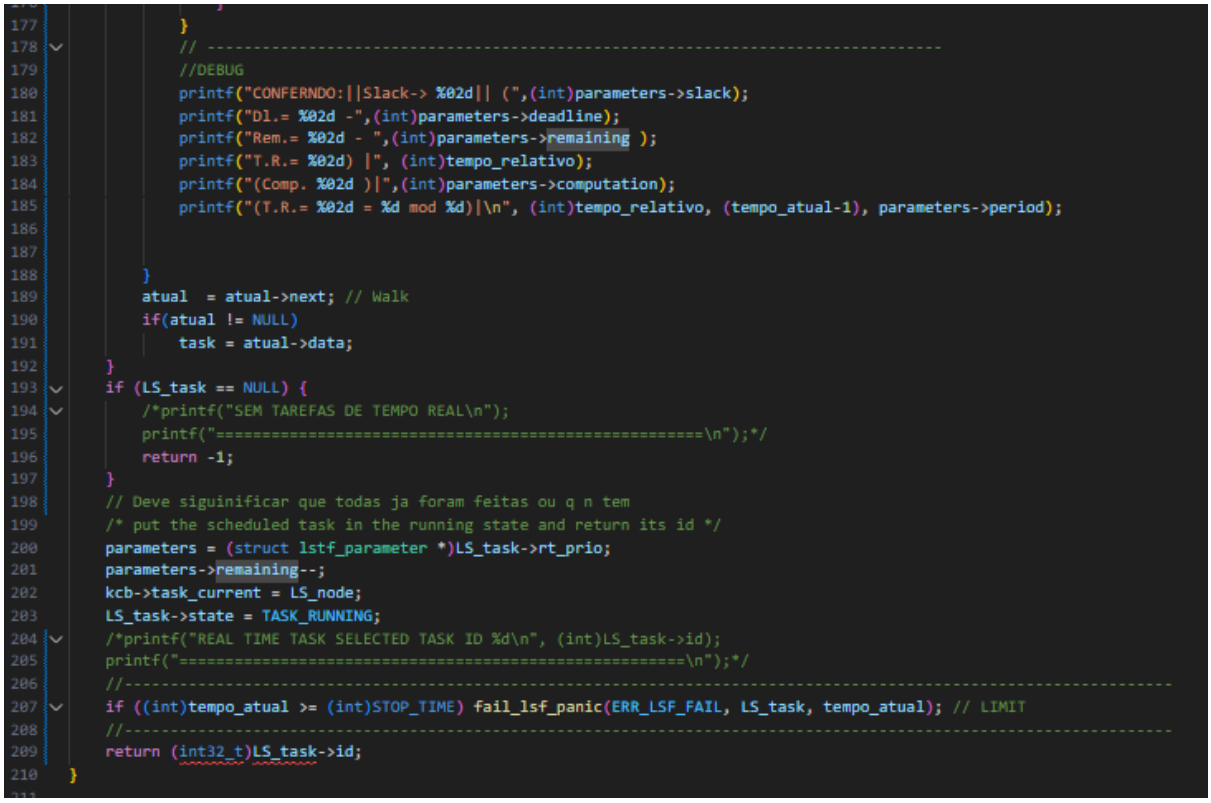
Variáveis:

```
int32_t lstf_sched(void){
    struct node_s *LS_node = kcb->tasks->head->next;
    struct node_s *atual = kcb->tasks->head->next; //inicio
    struct lstf_parameter *parameters = NULL;       You, 24 hours ago • quase …
    struct tcb_s *LS_task = atual->data;
    struct tcb_s *task= atual->data;
    int tempo_atual, tempo_relativo = 0;
    int ls = 255;   // UINT_MAX;
    int temp = 0;
    tempo_atual = (int)ticks_h();
    //-----------------------------------------------
    struct tcb_s *t = kcb->task_current->data;
    if (t->state == TASK_RUNNING) t->state = TASK_READY;
    //-----------------------------------------------
    printf("\n--------------------------------------------------------------------------\n");
    printf("=TEMPO TICKS %02d = \n", tempo_atual -1);
    LS_node = NULL;
    LS_task = NULL;
    while (atual) {
        if(task->rt_prio != 0 && task->priority !=TASK_IDLE_PRIO && task->state == TASK_READY ){
```

Parte do loop que controla o Acréscimo ou decréscimo do Slack e, se a tarefa é ou não escalonável.

```
while (atual) {
    if(task->rt_prio != 0 && task->priority !=TASK_IDLE_PRIO && task->state == TASK_READY ){
        parameters = (struct lstf_parameter *)task->rt_prio;
        tempo_relativo = (tempo_atual-1) % parameters->period;//(tempo_atual-1) pois no segundo 0 tem o IDLE
        // ------------------------------------------------------------------------------
        //DEBUG
        printf("-------------------------------------------------\n");
        printf("|Calculating Slack of TASK_ID %02d|\n", (int)task->id);
        if(parameters->remaining != 0 ){
            if (tempo_relativo== 0 && KILL_IF_DEADLINE_MISS ==0){   // RESTART parameters->remaining
                parameters->remaining = parameters->computation;
                temp = (int)parameters->deadline - (int)parameters->computation;
                printf("----->!!!!!!!!!!!!!!!DEADLINEMISS!!!!!!!!!!!!!!! ID %d\n", (int)task->id);
                printf("----->\"RESET\" TASK_ID %d\n", (int)task->id);
                parameters->slack =  (uint8_t)temp;
            }

            temp = (int)parameters->deadline - (int)parameters->remaining - (int)tempo_relativo;
            if (temp >=0 ){
                parameters->slack = (uint8_t)temp; //ATUALIZA SLACK e CONFERE MENOR SLACK VALIDO
                // ------------------------------------------------------------------------------------------------
                if (ls > parameters->slack) {
                    LS_task = task;
                    LS_node = atual;
                    ls = parameters->slack;
                }
                // ------------------------------------------------------------------------------------------------
            }

            else { // Caso slack negativo e o remaining positivo siguinifica que ocorreu um deadlinemiss!
                printf("----->!!!!!!!!!!!!!!!DEADLINEMISS!!!!!!!!!!!!!!! ID %d\n", (int)task->id);

                if (KILL_IF_DEADLINE_MISS) fail_lsf_panic(ERR_LSF_FAIL, LS_task, tempo_atual);
                }
        }
        // ------------------------------------------------------------------------------------------
        if(parameters->remaining == 0){
            printf("----->\"FINISHED\" TASK_ID %d\n", (int)task->id);// Finished inside deadline/period
            tempo_relativo = (tempo_atual-1) % parameters->period;
            if((tempo_relativo)== 0 ){  // RESTART parameters->remaining
                parameters->remaining = parameters->computation;
                temp = (int)parameters->deadline - (int)parameters->computation;
                printf("----->\"RESET\" TASK_ID %d\n", (int)task->id);
                parameters->slack =  (uint8_t)temp;
                // ------------------------------------------------------------------------------------------------
                if (ls > parameters->slack) { //ATUALIZA SLACK e CONFERE MENOR SLACK VALIDO
                    LS_task = task;
                    LS_node = atual;
                    ls = parameters->slack;
                }
                // ------------------------------------------------------------------------------------------------
            }
        }
        // ------------------------------------------------------------------------------
```

Parte final do loop que tem prints de debug e próximo elemento:



```
            }
            // ------------------------------------------------------------------------
            //DEBUG
            printf("CONFERNDO:||Slack-> %02d|| (",(int)parameters->slack);
            printf("Dl.= %02d -",(int)parameters->deadline);
            printf("Rem.= %02d - ",(int)parameters->remaining );
            printf("T.R.= %02d) |", (int)tempo_relativo);
            printf("(Comp. %02d )|",(int)parameters->computation);
            printf("(T.R.= %02d = %d mod %d)|\n", (int)tempo_relativo, (tempo_atual-1), parameters->period);


        }
        atual  = atual->next; // Walk
        if(atual != NULL)
            task = atual->data;
    }
    if (LS_task == NULL) {
        /*printf("SEM TAREFAS DE TEMPO REAL\n");
        printf("====================================================\n");*/
        return -1;
    }
    // Deve siguinificar que todas ja foram feitas ou q n tem
    /* put the scheduled task in the running state and return its id */
    parameters = (struct lstf_parameter *)LS_task->rt_prio;
    parameters->remaining--;
    kcb->task_current = LS_node;
    LS_task->state = TASK_RUNNING;
    /*printf("REAL TIME TASK SELECTED TASK ID %d\n", (int)LS_task->id);
    printf("====================================================\n");*/
    //-------------------------------------------------------------------------------------------------------------
    if ((int)tempo_atual >= (int)STOP_TIME) fail_lsf_panic(ERR_LSF_FAIL, LS_task, tempo_atual); // LIMIT
    //-------------------------------------------------------------------------------------------------------------
    return (int32_t)LS_task->id;
}
```

Formato das tarefas:

```
void task2(void)
{
    struct lstf_parameter *params;
    while (1) {
        params = ucx_task_ifno();
        printf("\n--------------------------------------------------------------------------\n");
        printf("-RUNNING \"task2\" <-----------------------------------------------------\n");
        printf("|TASK ID: %u | ", ucx_task_id());
        if (params != NULL) {
            printf("Comp.: %u | ", params->computation);
            printf("Per.: %u | ", params->period);
            printf("Dl.: %u | ", params->deadline);
            printf("Slack: %u | ", params->slack);
            printf("Remaining: %u |\n", params->remaining);
        }
        else {
            printf("[!] Parâmetros indisponíveis para esta tarefa.\n");
        }

        ucx_task_wfi();
    }
}
```

Nota: Os arquivos que mais foram alterados foram o kernel.h, ucx.c, e rtsched.c