

# CHAVE DE CONEXÃO

1

## PONTO 2.1

TIME-TRIGGERED-SYSTEMS E EVENT-TRIGGERED

SYSTEMS

DESCREVER SISTEMAS DE TEMPO REAL BASEADO EM TEMPO E EM EVENTOS. MOSTRAR VANTAGENS E DESVANTAGENS. MOSTRAR UMA IMPLEMENTAÇÃO EM C DE UMA BIBLIOTECA PARA IMPLEMENTAÇÃO DE UM SISTEMA MULTITAREFA BASEADO EM TEMPO E EM COMO DE UM EXEMPLO.

IMPLEMENTAR O MESMO EXEMPLO USANDO A API INDOCAIONADA POR UM KERNEL DE TEMPO REAL

## SOLUÇÃO

BASEADO EM TEMPO



BASEADO EM VANTAGEM PERIÓDICA  
NECESSITA APENAS UM TEMPORIZADOR  
TAREFAS SÃO RUN-TO-COMPLETION

BASEADO EM EVENTOS

REAGE AOS EVENTOS QUE PROVOCAM  
INTERRUPÇÕES

NECESSITA DE MECANISMO DE INTERRUPTO  
NO PROCESSADOR

GERALMENTE É USADO COM UM

CONTROLO DE INTERRUPTOS QUE

PERMITE VÁRIOS NÍVEIS DE

PRIORIDADE





2

## VANTAGENS DO TTS

ESCALONAMENTO MAIS SIMPLES  
DEMANDAS APENAS UM PILHA,  
EXIGINDO BEM MENOS RAM

CONCEITUALMENTE MAIS SIMPLES  
NÃO EXISTE PROBLEMA DE "CORNICAS"

NÃO OBTÊM MECANISMOS DE  
EXCLUSÃO MÚTUA

RESPONDIMENTO DE SEQUENCIAMENTO

## DESVANTAGENS

FÁCIL DE MODELAR QUE  
TEMPO DE EXECUÇÃO DE TAREFAS  
NÃO É O PERÍODO DE  
ESCALONAMENTO

MENOS FLEXÍVEL  
PODE SER MAIS LENTO (RESPOSTA)

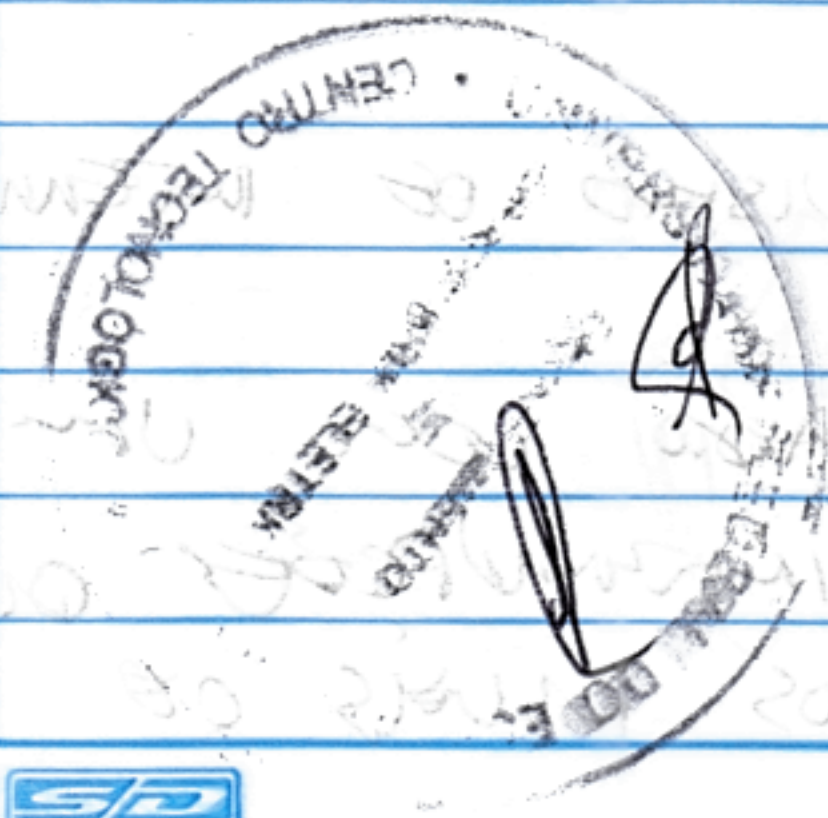
## VANTAGENS DO ETS

MAIS FLEXÍVEL MAS  
MAIS COMPLEXO  
ESCALONÁVEL

## DESVANTAGENS

NECESSITA UMA PILHA PARA  
CADA TAREFA QUE DEVE TER  
TAMANHO MÁXIMO

OCCORRER "CORNICAS" QUANDO  
TIDAMENTE DEMANDAS DE  
MECANISMOS DE SINCRONIZAÇÃO  
IMPLEMENTADOS BEM MAIS COMPLEXO  
MAIOR (RAM E RAM)





(3)

## IMPLEMENTAÇÃO TTS

```

typedef struct {
    void (*func)(void)
    uint periodo;
    uint contador;
    uint del run
} TCB TCB_t,
TCB_t TCB[10];

```

DISPATCH EXECUTADO NO main

```

void dispatch(void) { // verifica posição ocupada
    for (int i = 0; i < 10; i++) {
        if (TCB[i].contador == 0) {
            TCB[i].func();
            TCB[i].contador = TCB[i].periodo;
        } else {
            TCB[i].contador--;
        }
    }
}

```

UPDATE EXECUTADO NA interrupção PERIODICA (SysTick\_Handler)

```

void update(void) {

```

```

    for (int i = 0; i < 10; i++) {
        TCB[i].contador--;
        // evitar valor < 0
    }
}

```



(A)

(B)

## EXEMPLO

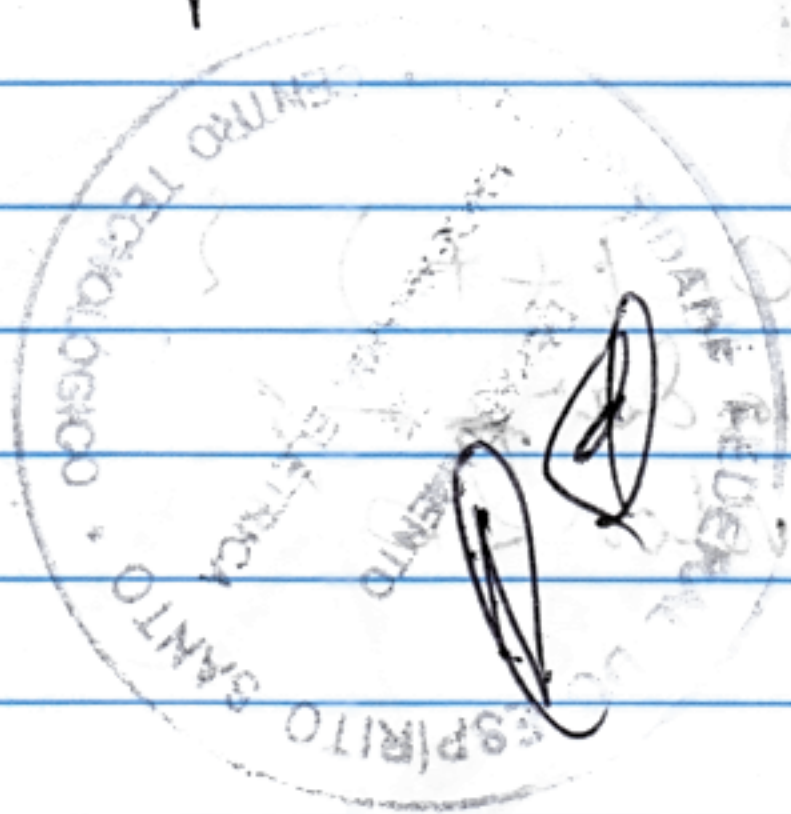
// 2 LEDs com freq. diferentes

```
void main() {  
    Init();  
    AddTask(0, FuncA, 10, 1);  
    AddTask(1, FuncB, 17, 1);  
    while(1) {  
        Dispatch();  
    }  
}
```

período  
chamada

```
void FuncA() {  
    ToggleLED(1);  
}  
void FuncB() {  
    ToggleLED(2);  
}
```

// poderiam ser 100, etc.





USANDO ETS

```

OS Stack StackA[1000];
OS Stack StackB[1000];
void main()

```

```

OSInit();

```

```

OSAdd(FuncA, 10, StackA, 1000);
OSAdd(FuncB, 13, StackB, 1000);

```

```

OSStart();

```

```

}

```

```

void TaskA(void)

```

```

while(1)

```

```

    LED_Toggle(1);
    Delay(10);

```

```

}

```

```

}

```

```

void TaskB(void)

```

```

while(1)

```

```

    LED_Toggle(2);
    Delay(13);

```

```

}

```

```

}

```







UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

**PROTOCOLO DE ASSINATURA**



O documento acima foi assinado digitalmente com senha eletrônica através do Protocolo Web, conforme Portaria UFES nº 1.269 de 30/08/2018, por  
MARIA DA CONCEICAO CARNEIRO DA SILVA - SIAPE 1996445  
Departamento de Engenharia Elétrica - DEE/CT  
Em 05/05/2025 às 17:13

Para verificar as assinaturas e visualizar o documento original acesse o link: <https://api-lepisma.prod.uks.ufes.br/arquivos-assinados/1122224?tipoArquivo=O>