

Sistema Operacional HellfireOS (introdução)

HellfireOS

2

□ Características

- ☒ Sistema operacional de tempo real preemptivo
- ▮ Projetado para sistemas MPSoC
- ▮ Gerenciamento dinâmico de tarefas
- ▮ Chamadas de sistema
 - perdas de deadline
 - tempo de trocas de contexto
 - parâmetros de tarefas
 - uso de processador
 - memória

HellfireOS

3

- Diferentes políticas de escalonamento (RM, RR, EDF, DM)
- Primitivas de exclusão mútua, semáforos
- Proteção contra inversão de prioridades (priority inheritance)
- Alocação, liberação e gerenciamento dinâmico de memória
- LibC customizada
- Emulação de ponto flutuante
- Verificação de integridade
- Comunicação entre tarefas (memória compartilhada e trocas de mensagem)

HellfireOS

4

□ API do HellfireOS

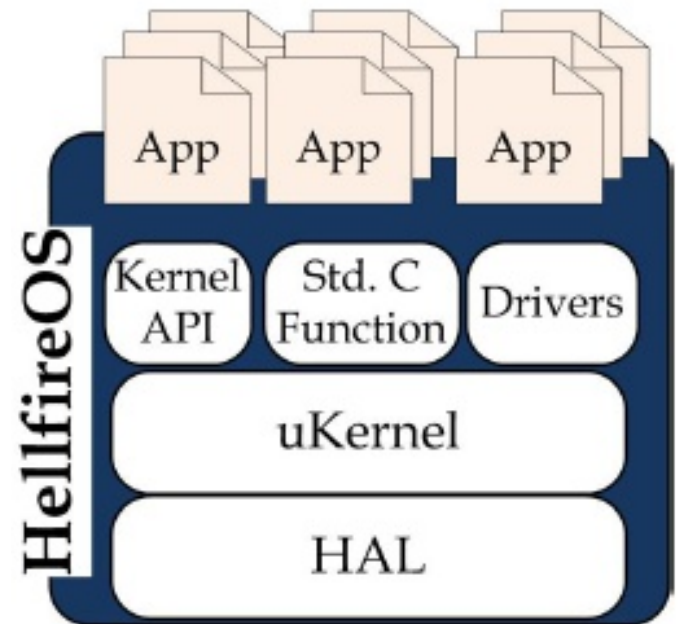
▮ Dividida em 5 grupos

- Manipulação de Tarefas
- Exclusão Mútua
- Manipulação de Memória
- Comunicação entre Processadores
- LibC

HellfireOS - estrutura

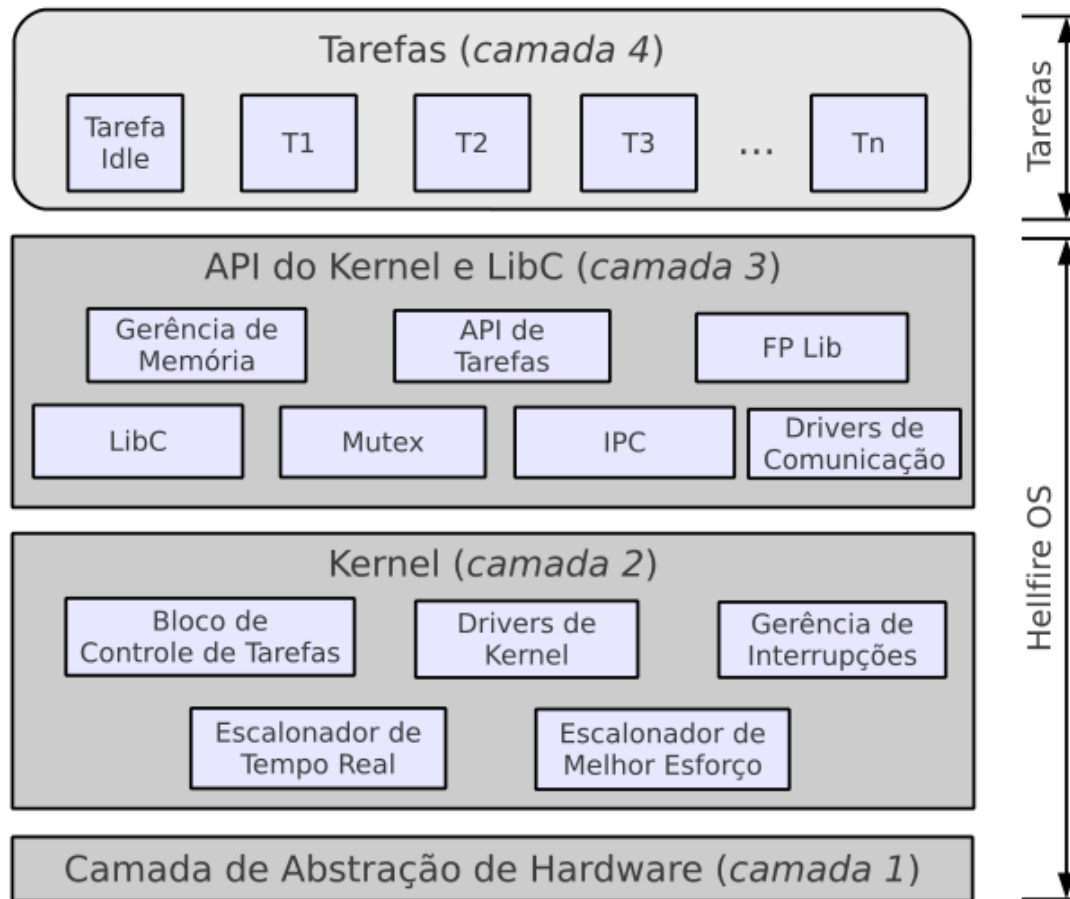
5

- Periféricos mapeados em memória
- Camada de abstração de hardware
 - ▮ Permite maior portabilidade



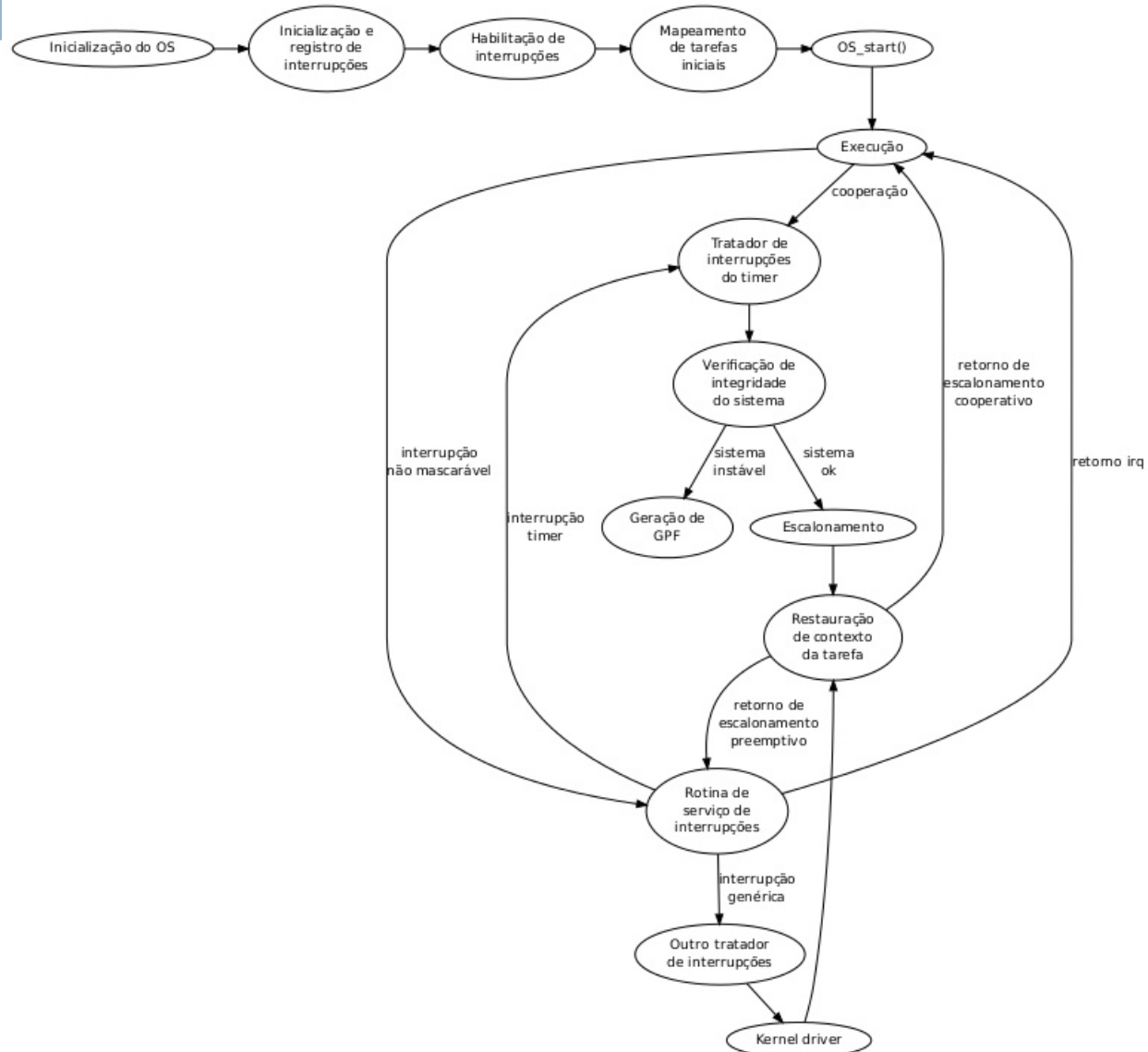
HellfireOS - estrutura

6



Fluxo de execução

7



HellfireOS - tarefas

8

□ Tarefas (TASKS)

- ▮ Unidade de escalonamento
- ▮ Semelhantes a threads
- ▮ Possuem o comportamento definido em um bloco de código
- ▮ São adicionadas ao sistema durante a inicialização ou execução, com parâmetros que definem seus requisitos de tempo real, tamanho da pilha, atributos ...
- ▮ Estruturas de controle e pilha alocadas dinamicamente pelo kernel

HellfireOS - tarefas

9

□ Tarefas (TASKS)

```
void Task(void){
    /* (variáveis alocadas na pilha) */
    unsigned int i,j;

    /* (código de inicialização) */
    j = 10;

    while(1){
        /* (código da tarefa) */
        for(i=0 ; i<j ; i++){
            printf("\nHello World! %d", i);
        }
    }
}
```

HellfireOS - tarefas

10

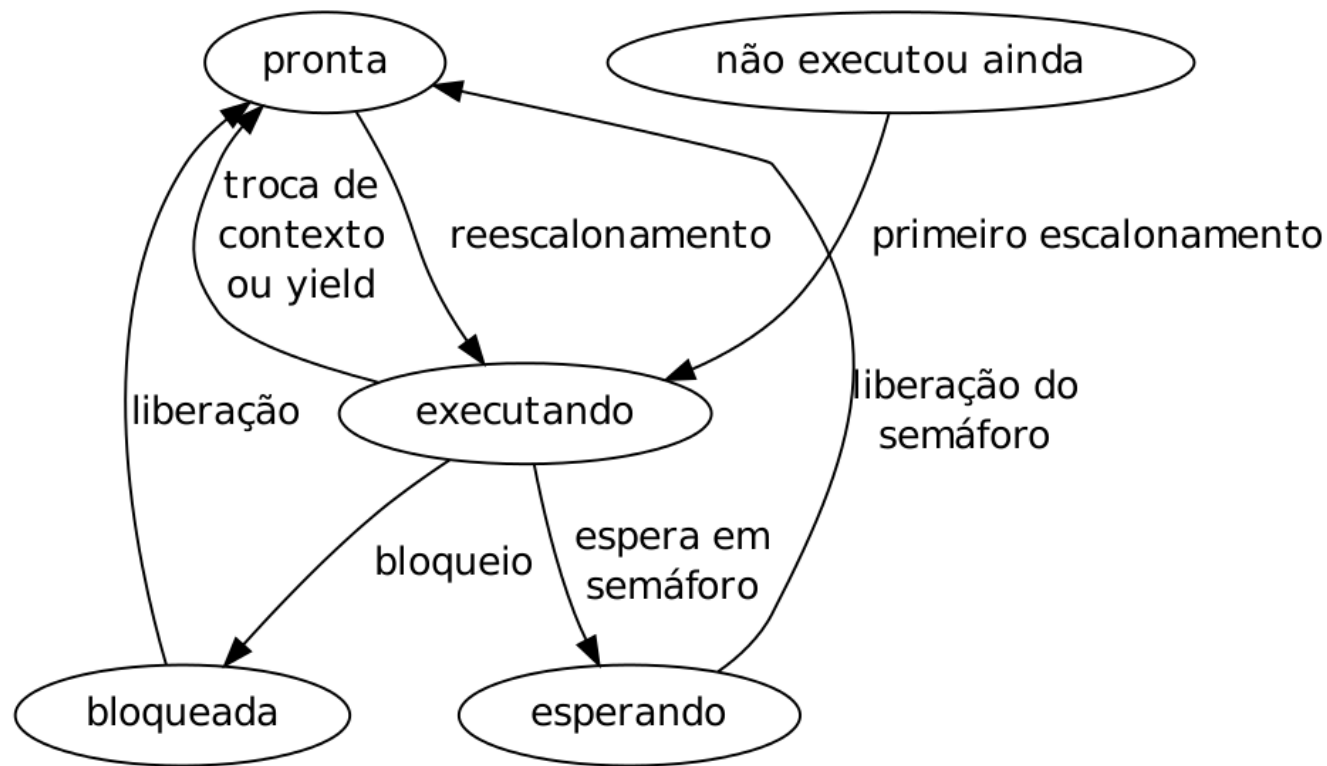


Figura 5.5 – Estados das tarefas

HellfireOS – escalonamento

11

Tabela 5.1 – Valores para tempos de *tick*

Frequência de operação (MHz)	15	16	17	18	19	20	21
25	1.31ms	2.62ms	5.24ms	10.48ms	20.97ms	41.94ms	83.88ms
33	0.99ms	1.98ms	3.97ms	7.94ms	15.88ms	31.77ms	63.55ms
50	0.65ms	1.31ms	2.62ms	5.24ms	10.48ms	20.97ms	41.94ms
66	0.49ms	0.99ms	1.98ms	3.97ms	7.94ms	15.88ms	31.77ms
100	0.32ms	0.65ms	1.31ms	2.62ms	5.24ms	10.48ms	20.97ms

Tabela 5.2 – Número de trocas de contexto

Frequência de operação (MHz)	15	16	17	18	19	20	21
25	763.36	381.68	190.84	95.42	47.69	23.84	11.92
33	1010.1	505.05	251.89	125.94	62.97	31.48	15.74
50	1538.46	763.36	381.68	190.84	95.42	47.69	23.84
66	2040.82	1010.1	505.05	251.89	125.94	62.97	31.48
100	3125	1538.46	763.36	381.68	190.84	95.42	47.69

$$period = \frac{2^a}{freq}$$

HellfireOS - escalonamento

12

- Escalonamento em 2 níveis
 - Nível 1: tempo real
 - Nível 2: melhor esforço
- Inicialmente, é invocado o escalonador de tempo real (normalmente RM). Caso a tarefa a ser escalonada seja a IDLE, é chamado o escalonador de melhor esforço.
 - Tarefa IDLE compete com outras de melhor esforço em um escalonador circular

HellfireOS - escalonamento

13

- Tarefas de tempo real $\rightarrow t = \{p, c, d\}$
- $t1 = \{4,1,4\}$, $t2 = \{6,2,6\}$, $t3 = \{8,1,8\}$
- $t0, t4, t5, t6$ e $t7$ - melhor esforço

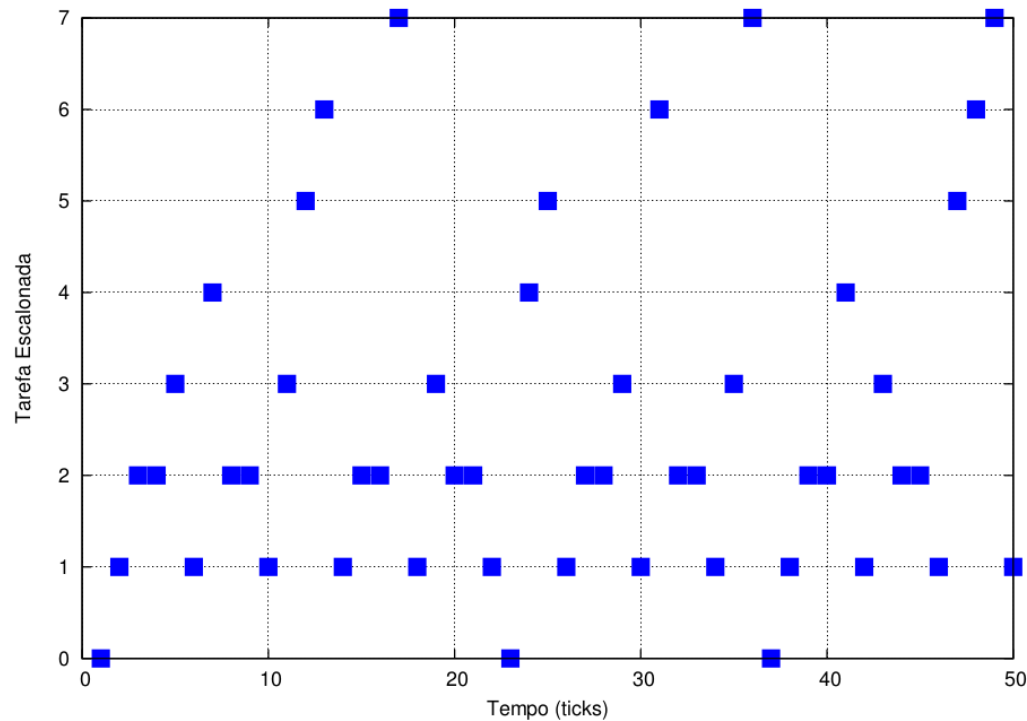
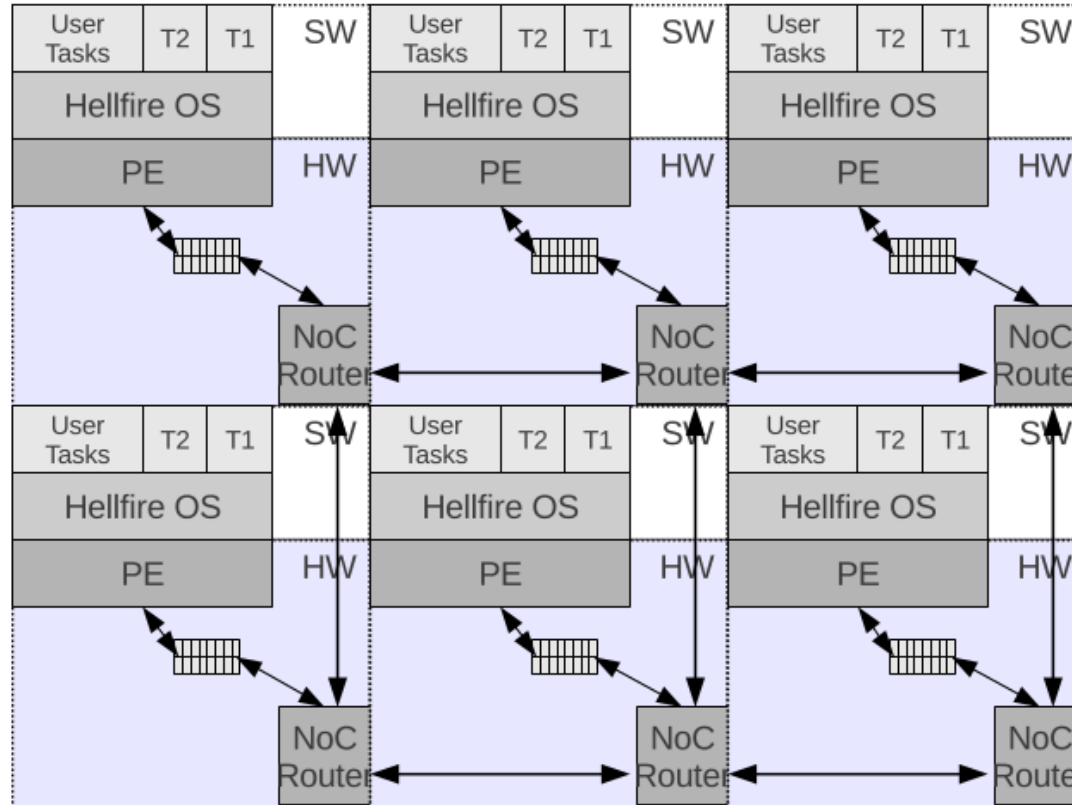


Figura 5.8 – Escalonamento de tarefas em dois níveis

Alexandra Aguiar, Sérgio
Johann, Felipe Magalhães,
Fabiano Hessel

HellfireOS - hardware

14



HellfireOS - hardware

15

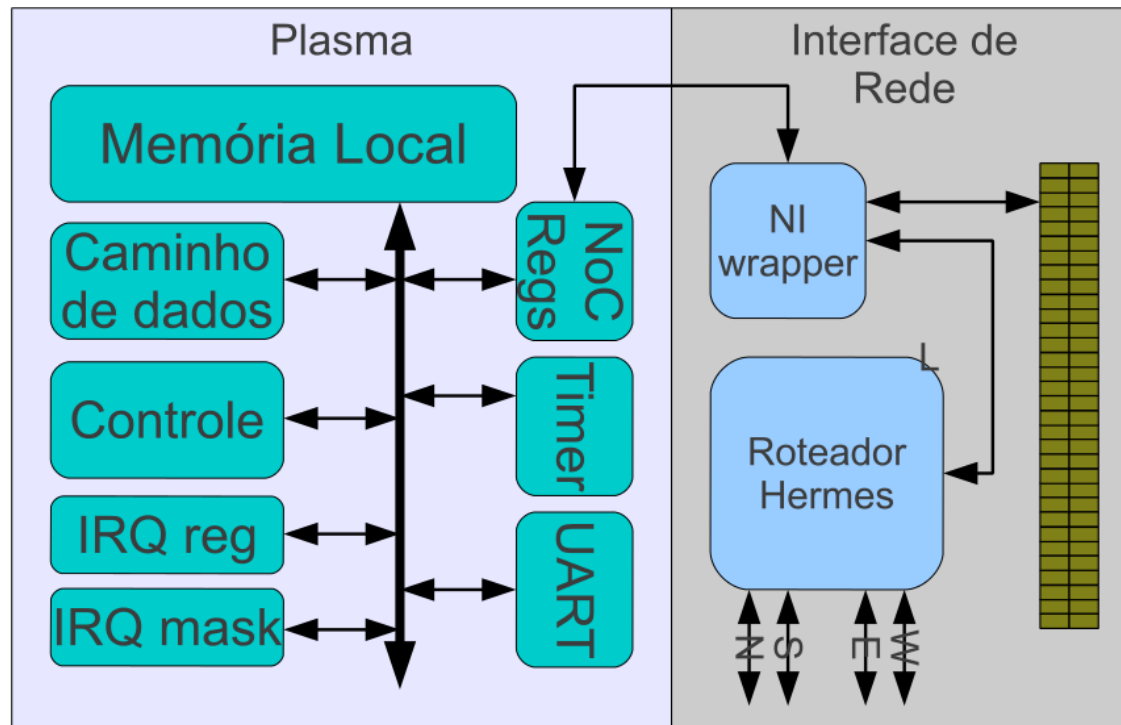


Figura 5.21 – Componentes de um nodo

HellfireOS - sincronização

16

- Comunicação entre tarefas
 - ▮ Modelo pode ser abstraído pelo OS
 - ▮ Localmente por memória compartilhada
 - Mutexes, semáforos, mailboxes
 - ▮ Remotamente por trocas de mensagem
 - Modelo produtor / consumidor
 - Filas individuais para cada tarefa
 - Primitivas send() e receive()

HellfireOS - sincronização

17

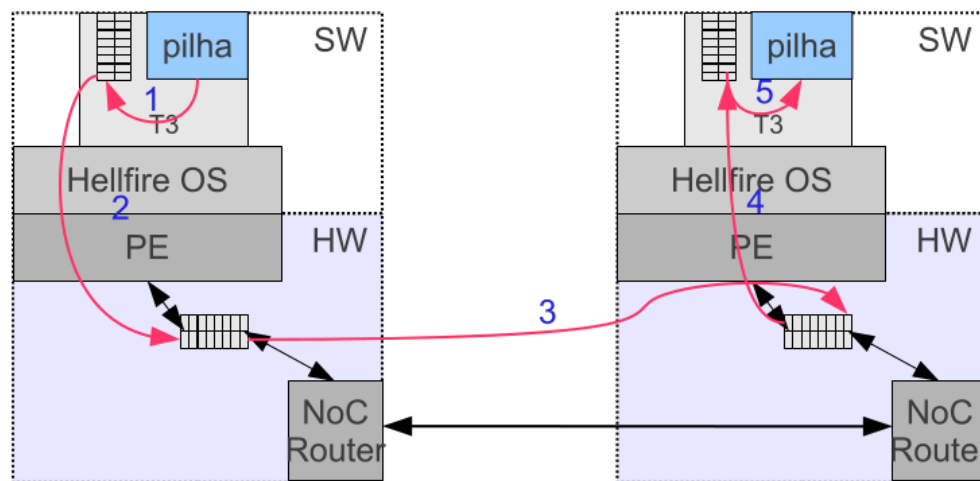


Figura 5.9 – Comunicação entre tarefas, filas de *software* e *hardware*

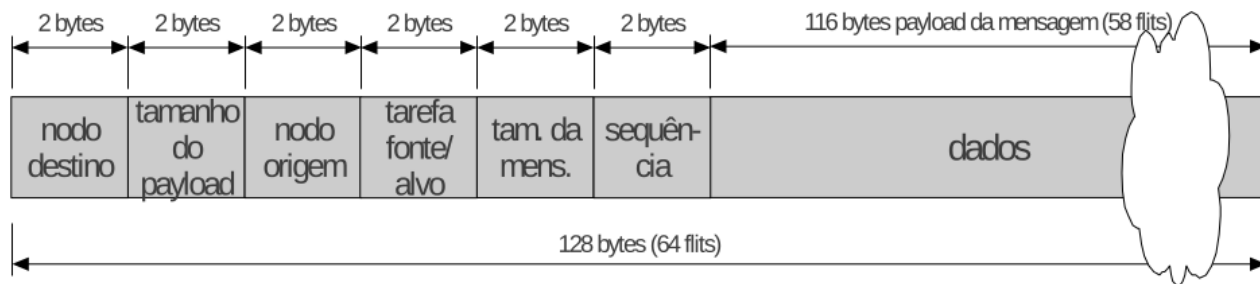


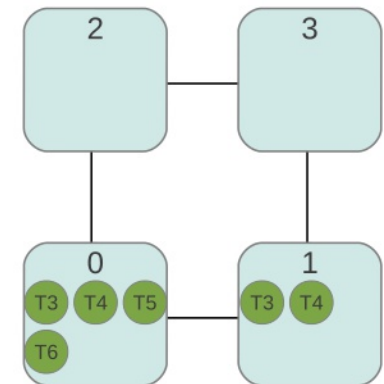
Figura 5.10 – Formato do pacote de dados

HellfireOS – mapeamento

18

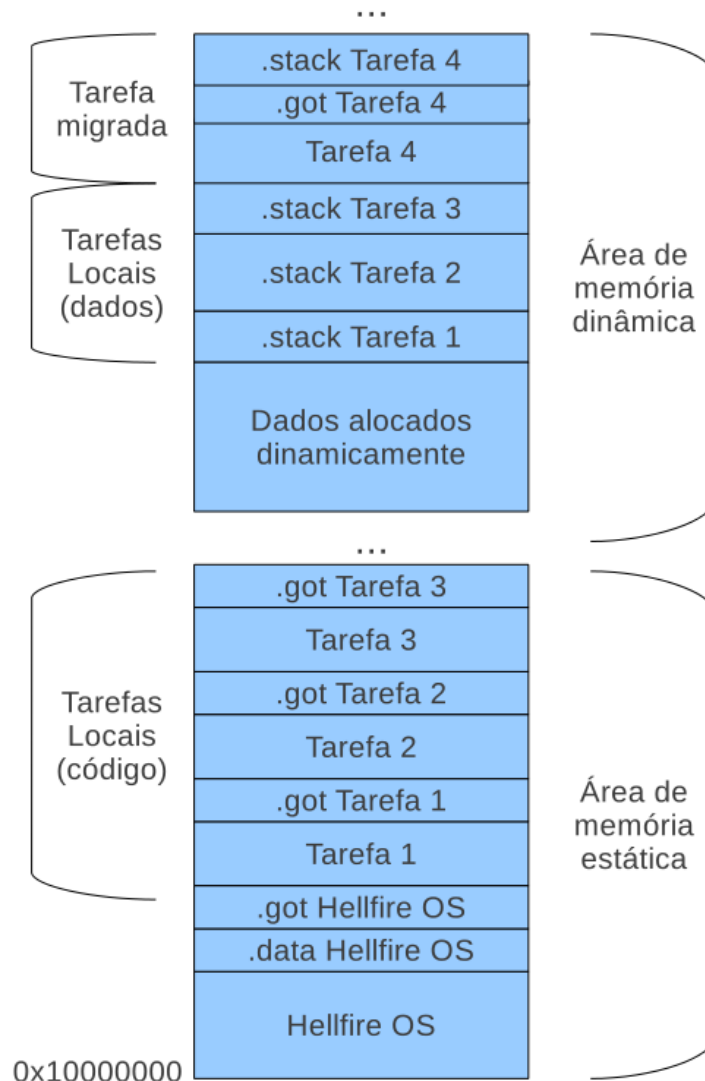
```
void ApplicationMain(void){
    // tarefa Ti = <ti, nCPU_IDti, pi, ei, di>
    // tarefas T0, T1 e T2 são mapeadas durante a inicialização
    #if CPU_ID==0
        // tarefa T3 = <t3, n0t3, 50, 1, 50>
        OS_AddPeriodicTask(task1, 50,1,50, "task one", 1024, 1, TASK_CAN_MIGRATE);
        // tarefa T4 = <t4, n0t4, 5, 1, 5>
        OS_AddPeriodicTask(task2, 5,1,5, "task two", 1500, 1, TASK_CAN_MIGRATE);
        // tarefa T5 = <t5, n0t5, 5, 1, 5>
        OS_AddPeriodicTask(task3, 5,1,5, "task three", 1050, 1, TASK_CANNOT_MIGRATE);
        // tarefa T6 = <t6, n0t6>
        OS_AddTask(task4, "task four", 1024, 1, TASK_CANNOT_MIGRATE);
    #endif
    #if CPU_ID==1
        // tarefa T3 = <t3, n1t3, 5, 1, 5>
        OS_AddPeriodicTask(task1, 5,1,5, "task one", 2048, 1, TASK_CAN_MIGRATE);
        // tarefa T4 = <t4, n1t4, 5, 1, 5>
        OS_AddPeriodicTask(task2, 5,1,5, "task two", 2048, 1, TASK_CAN_MIGRATE);
    #endif

    OS_Start();
}
```



HellfireOS - mapeamento

19



Alexandra Aguiar, Sérgio
Lorenzini, Felipe Magalhães,
Rodrigo Hessel

HellfireOS – desempenho

20

Tabela 6.1 – Tempo e *Overhead* das trocas de contexto em função do número de tarefas

Frequência		25MHz		100MHz		500MHz	
Número de Tarefas	Ciclos	Tempo	Overhead	Tempo	Overhead	Tempo	Overhead
5	1013	40us	0.38%	10us	0.09%	2us	0.01%
10	1563	62us	0.59%	16us	0.15%	3us	0.03%
15	2113	84us	0.80%	21us	0.20%	4us	0.04%
20	2663	106us	1.01%	27us	0.25%	5us	0.05%
25	3215	128us	1.22%	32us	0.30%	6us	0.06%
30	3765	150us	1.43%	37us	0.36%	7us	0.07%
35	4315	172us	1.64%	43us	0.41%	9us	0.08%
40	4865	194us	1.85%	49us	0.46%	10us	0.09%
45	5415	216us	2.06%	54us	0.51%	11us	0.10%
50	5965	238us	2.27%	60us	0.57%	12us	0.11%
55	6513	260us	2.48%	65us	0.62%	13us	0.12%
60	7063	282us	2.69%	71us	0.67%	14us	0.13%
65	7613	304us	2.90%	76us	0.72%	15us	0.14%
70	8165	326us	3.11%	82us	0.78%	16us	0.15%

HellfireOS – desempenho

21

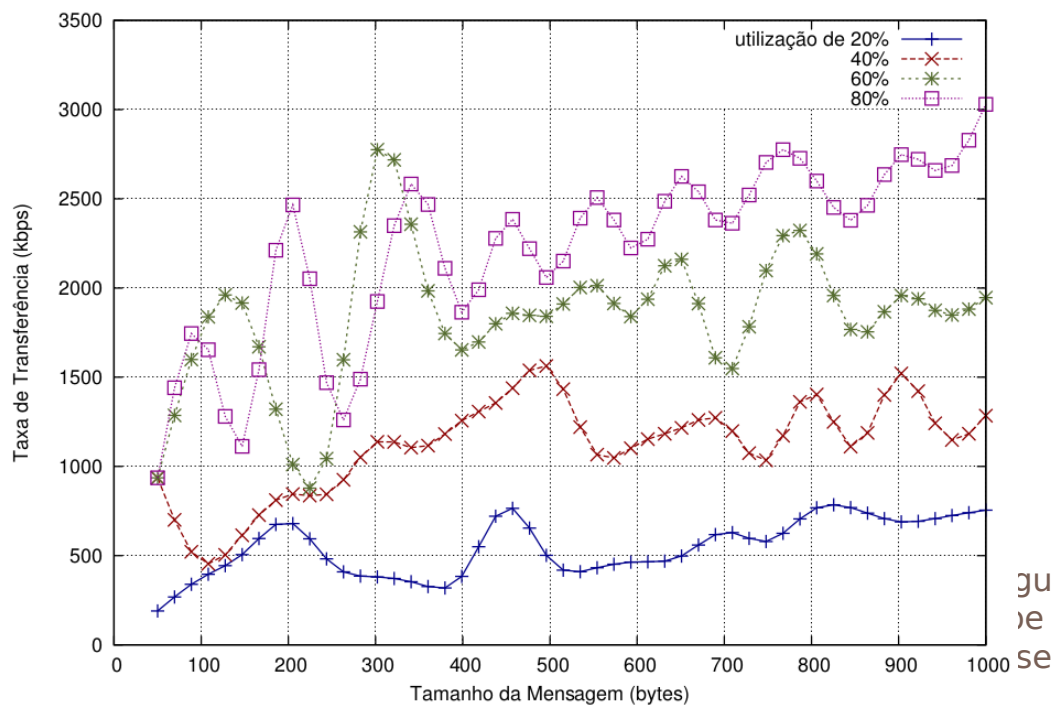
Tabela 6.2 – *Overhead* de primitivas ou eventos

<i>Syscall</i> / Evento	Ciclos	Tempo
Troca de contexto (10 tarefas)	1563	62us
ISR da rede (64 <i>flits</i>)	1079	43us
OS_AddPeriodicTask()	2842	114us
OS_BlockTask()	88	<4us
OS_ResumeTask()	84	<4us
OS_Fork()	49107	1964us
OS_KillTask()	3087	123us
OS_ChangeTaskParameters()	122	5us
OS_CurrentTaskId()	9	<1us
OS_Malloc() (4kB)	754	30us
OS_Free() (4kB)	727	29us
OS_Calloc() (4kB)	21302	852us
OS_Realloc() (4kB para 8kB)	10760	430us

HellfireOS – desempenho

22

- Comunicação entre tarefas em nodos distintos (MPSoC @ 25 MHz – roteadores da NoC e processadores)
- Desempenho das primitivas de comunicação em nível de aplicação como função de utilização de CPU (tarefas RT)
- Novo driver: melhoria de 5x frente estes resultados (10 a 20 MBit/s)



guiar, Sérgio
de Magalhães,
sel