Contabilização de tarefas

Este projeto faz parte do PPOS v2.

Neste projeto, você irá adicionar mecanismos para contabilizar o uso do processador pelas tarefas em execução. Sua implementação deve produzir uma mensagem de saída com o seguinte formato, para cada tarefa que finaliza (incluindo o próprio *dispatcher*):

```
PPOS: task   5 (ping), 13833 ms run,  2772 ms cpu,   278 acts, exit code   0
```

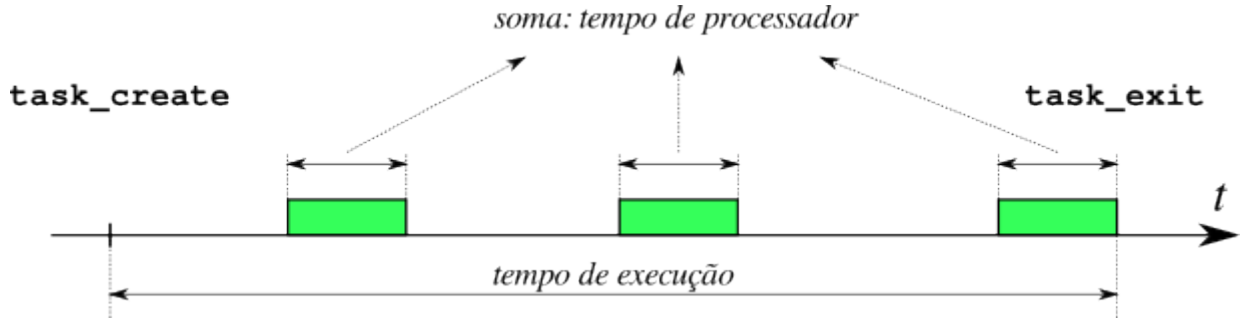
Valores impressos:

- 5: task id
- `ping`: nome da tarefa
- 13833: tempo de vida (execução)
- 2772: tempo de CPU (processamento)
- 278: ativações (quantas vezes recebeu a CPU)
- `0`: exit code (informado em `task_exit`)

Essa mensagem deve ser gerada pelo despachante, assim que cada tarefa encerrar.

Cálculo dos tempos

A figura abaixo ilustra a execução de uma determinada tarefa, de sua criação (`task_create`) ao seu encerramento (`task_exit`). As áreas em verde indicam o uso do processador. É fácil perceber como os valores de contabilização podem ser calculados:



Para a contabilização você precisará de uma **referência de tempo** (ou seja, um relógio). Para isso, pode ser definida uma variável global (em `time.c`) para contar *ticks* de relógio, incrementada a cada interrupção do temporizador (1 ms). Dessa forma, essa variável indicará o número de *ticks* decorridos desde a inicialização do sistema na função `ppos_init`, ou seja, funcionará como um relógio baseado em milissegundos.



Um dos objetivos deste projeto é entender como é implementada a contagem de tempo no kernel de um SO. Por isso, você deve construir seu próprio relógio, sem usar funções de consulta ao relógio do Linux como `gettimeofday` e outras.

Você deverá implementar a seguinte função em `time.c` para informar às tarefas o valor corrente do relógio:

```
unsigned int time()
```

Arquivos

Os seguintes arquivos são relevantes para este projeto:

- `kernel/time.c`: gestão de tempo
- `kernel/task.c`: gestão de tarefas
- `kernel/tcb.c`: define a estrutura do task control block
- `test/pingpong-contab.c`: teste sem prioridades; as tarefas devem concluir juntas, com tempos de vida e de CPU similares e número de ativações também similares.
- `test/pingpong-contab-prio.c`: teste com prioridades; as tarefas devem ter tempos de vida bem distintos, mas tempo de CPU e número de ativações similares (pois a carga computacional delas é a mesma).
- `test/pingpong-contab-stress.c`: teste para verificar se seu sistema se comporta bem com muitas tarefas (todas têm a mesma prioridade e mesma carga computacional).
- `test/pingpong-contab*.txt`: saídas esperadas (pequenos desvios são aceitáveis)

Outras informações

- Duração estimada: 4 horas.
- Dependências:
  - Tarefas cooperativas
  - Despachante de tarefas
  - Escalonador por prioridades (para o teste 2)
  - Preempção por Tempo

From:
https://wiki.inf.ufpr.br/maziero/ - **Prof. Carlos Maziero**

Permanent link:
https://wiki.inf.ufpr.br/maziero/doku.php?id=ppos-v2:relogio_e_contabilizacao

Last update: **2026/07/29 17:21**