Relatório dos Componentes de Escalonamento

Pedro Rosanes

14 de dezembro de 2010

1 Análise do Escalonador Padrão

O escalonador padrão adota uma política FIFO. Ele provê as interfaces *Scheduler* e *TaskBasic*. As tarefas se conectam ao escalonador através da *TaskBasic*. Ao compilar um programa em NesC, todas tarefas básicas viram uma interface desse tipo. Porém, para se diferenciarem é criado um parâmetro na interface [1].

A fila implementada **não** funciona como um vetor circular. Existe um 'ponteiro' para a cabeça e um para o rabo, além de um vetor de tamanho 255 (O que limita a quantidade máxima de tarefas). A cabeça contém o identificador do primeiro na fila. A célula cujo index corresponde ao *id* do primeiro contém o *id* do segundo, e assim em diante. Chega-se ao fim da fila quando a célula contém um identificador de vázio.

2 Análise do Escalonador *Earliest Deadline First*

Este escalonador aceita tarefas com deadline e elege as que menor *deadline* para executar. A interface usada para criar esse tipo de tarefas é *TaskDeadline*. O *deadline* é passado por parâmetro pela função *postTask*. As *TaskBasic* também são aceitas como recomendado pelo TEP 106[2].

Em contraste, o escalonador não segue outra recomendação. Ele não elimina a possibilidade de *starvation* pois as tarefas básicas só são atendidas quando não há nenhuma com *deadline* para ser atendida. A fila de prioridades é implementada da mesma forma que a do escalonador padrão1, a única mudança está na inserção. Para inserir, a fila é percorrida do começo até o fim, procurando-se o local exato de inserção. O problema é que essa operação custa $\bigcirc(n)$. Isso poderia ser evitado implementando uma *heap* que tem complexidade de inserção de $\bigcirc(\log n)$.

Tive problemas com o componente *Counter32khzC*, pois ele não existe para o *micaz*. Para poder compilar o escalonador tive de tirá-lo. Ele era usado para calcular a hora atual, e somar ao deadline. Sem esse componente, o escalonador virá um escalonador de prioridades (mínimo). Porém, o escalonador não está rodando o *taskLoop*, apesar de inicializar e verificar se existe tarefas na etapa de inicialização[3].

[1] Para mais informações sobre interfaces parametrizadas olhar o livro TinyOS Programming[1, s. 8.3 e 9].

Referências

[1] P. Levis e D. Gay. TinyOS Programming, capítulo 11, 2009.

[2] P. Levis e C. Sharp. TEP 106: Schedulers and Tasks. http://www.tinyos.net/tinyos-2.x/doc/html/tep106.html

[3] Boot Sequence, TinyOS Tutorial. http://docs.tinyos.net/index.php/Boot_Sequence