

Sistemas Embarcados - Trabalho Prático I

Lucas Fialho e Felipe Peiter

May 2, 2018

kernel.h

Na struct de `tcb_entry` foi adicionado um contador de jobs aperiódicos:

(uint32_t apjobs;)

Foi também criado um ponteiro para a fila de tarefas aperiódicas:

*(struct queue *krl_ap_queue;)*

main.c

Aqui foi inicializada as estruturas criadas em conjuntos com as estruturas do *kernel.h*.

panic.c

Foram criadas erros de panic personalizados para as tarefas aperiódicas.

task.c

Foi feita divisão das tarefas para os preenchimentos de suas queues específicas, de acordo com as condições especificadas na descrição do trabalho.

scheduler.c

Foi modificado para incluir o servidor desenvolvido, no qual foi implementado utilizando um task RT que é a aplicação *pollingServer.c*.

pollingServer.c

A função *ap_queue_next* avança o elemento da queue, assim como a função *run_queue_next* e *rt_queue_next* do arquivo *scheduler.c*

A função *ap_sched* irá processar o primeiro elemento da queue e validar se possui outros elementos. Caso ainda possui jobs aperiódicos ele irá decrementar o contador de tarefas restantes e salvar o contexto do kernel, verificando se o mesmo foi alterado. Caso não foi alterado irá processar o primeiro elemento da queue. Assim como irá escalonar as tarefas da fila.

A função *task* irá executar tarefas aperiódicas, essa função foi copiada de *sched_test2.c*

A função *randomAP* irá gerar tasks de maneira aleatória entre 50ms a 500ms utilizando a função de delay fornecida.

O *app_main* irá inicializar o Polling Server e disparar as tasks descritas anteriormente.