

Trabalho de Programação 2

Processador CESAR16i – Parte 2 – Kernel

1. Descrição Geral

Nessa segunda parte do trabalho você deverá implementar o kernel e suas funções. Isso envolve a inicialização do kernel e suas variáveis, com a inicialização do sistema de interrupções e os tratadores de interrupção do teclado e do timer.

Da mesma forma que na primeira parte do trabalho, você deverá entregar **apenas** o programa fonte do seu kernel (arquivo .CED). Para o desenvolvimento desta parte do trabalho, será colocado à disposição um arquivo fonte que poderá ser usado como base para o desenvolvimento (KERNEL_REF.CED).

Também será disponibilizado um arquivo com a implementação de uma aplicação (APP_PROF.MEM). Esse arquivo deverá ser carregado no simulador, usando a carga parcial, logo após ter sido feita a carga de seu kernel, para testar sua implementação do kernel.

O kernel deverá ser construído de maneira a possuir suporte às interrupções de teclado e timer e também para as funções descritas, conforme arquivo “FuncoesDoKernel.pdf” que acompanha essa especificação.

Ao terminar a inicialização do kernel e as interrupções, o kernel deverá executar um saldo (JMP) para o endereço onde está o programa de aplicação, para que a aplicação possa iniciar sua operação. No arquivo de referência fornecido o símbolo “_APP” é usado para indicar o endereço onde estará a aplicação. A instrução a ser executada é a seguinte:

```
JMP    _APP
```

2. Divisão do espaço de endereçamento (alocação de memória)

Para a implementação do trabalho o espaço de endereçamento do CESAR será dividido da seguinte forma:

Faixa de Endereços	Descrição
0 a 255 (H0000 a H00FF)	Essa área é de uso exclusivo do kernel. Aqui você deve colocar as primeiras instruções a serem executadas pelo kernel na inicialização. Aqui também está o ponto de entrada das funções a serem implementadas, no endereço 128 (H0080), identificado pelo rótulo _SISTEMA.
256 32767 (H0100 a H7FFF)	Área para o programa de aplicação que está sendo fornecido (APP_PROF.MEM) para testar o kernel. Ele será carregado usando uma carga parcial do arquivo para os endereços 256 32767 (H0100 a H7FFF).
32768 a 65535 (H8000 a HFFFF)	Essa área é de uso exclusivo do kernel. Aqui você deve colocar todo o seu código de inicialização do sistema de interrupções, tratamento das interrupções e as implementações das funções.

3. Detalhes da área 0 a 255 (H0000 até H00FF)

Essa área é reservada para o kernel porque, ao iniciar a execução de qualquer programa, a primeira instrução buscada é aquela que se encontra no endereço 0. Dessa forma, o kernel inicia sua operação no endereço 0 e, caso não haja espaço suficiente nesses primeiros 256 bytes para a sua implementação, deve-se recorrer a funções que estarão codificadas na região 32768 a 65535 (H8000 até HFFFF).

Nessa área também se deve colocar o ponto de entrada das funções a serem implementadas, conforme descrito no arquivo “FuncoesDoKernel”. Conforme descrito, o endereço 128 (H0080) (identificado pelo rótulo _SISTEMA) é o endereço a ser chamado para iniciar a execução de qualquer uma das funções. Para identificar a função a ser executada, o programa de aplicação deve colocar o seu código correspondente no registrador R5. Portanto, para que a aplicação possa chamar uma dessas funções, deverão ser utilizadas as seguintes duas linhas de comando:

```
MOV    #CódigoDaFuncao,R5  
JSR    R7,_SISTEMA
```

Notar que, também nesse caso, caso não seja possível colocar todo o código de tratamento dessas funções nos 256 bytes dessa área da memória, pode-se utilizar a área compreendida entre 32768 e 65535 (H8000 e HFFFF).

4. Correção e Entregáveis

A correção desta primeira parte do trabalho será feita usando o programa de teste especialmente desenvolvido para esta finalidade, que será carregado no simulador através de uma carga parcial nos endereços 256 a 32767 (H0100 a H7FFF) da memória após a carga de sua implementação do kernel.

Deve ser entregue um arquivo fonte (arquivo .CED) com a solução correspondente, escrito em linguagem simbólica do CESAR16i usando o montador Daedalus. O código do programa fonte deverá conter comentários descritivos da implementação.

Esta parte do trabalho deverá ser entregue até a data prevista indicada no sistema Moodle. Não serão aceitos trabalhos entregues além do prazo estabelecido. Trabalhos não entregues até a data prevista receberão nota zero.

5. Observações

Recomenda-se a troca de ideias entre os alunos. Entretanto, a identificação de cópias de trabalhos acarretará na aplicação do Código Disciplinar Discente e a tomada das medidas cabíveis para essa situação (**tanto o trabalho original quanto os copiados receberão nota zero**).

O professor da disciplina reserva-se o direito, caso necessário, de solicitar uma demonstração do programa, onde o aluno será arguido sobre o trabalho como um todo. Nesse caso, a nota final do trabalho levará em consideração o resultado da demonstração.