

Trabalho de Programação 2 Processador CESAR16i

1. Descrição Geral

Nesse trabalho você deverá implementar o kernel e as funções de sua API. Isso envolve a inicialização do kernel e suas variáveis, a inicialização do sistema de interrupções, os tratadores de interrupção do teclado e do timer e algumas funções de “SISTEMA”.

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, sugerimos, será 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 (endereços H8000 até HFFBF), logo após ter sido feita a carga de seu kernel, para testar sua implementação.

O kernel deverá ser construído de maneira a possuir suporte às interrupções de teclado e timer e também para as funções de sistema, conforme descritas no arquivo “FuncoesDoKernel.pdf” que acompanha essa especificação.

Ao terminar a inicialização do kernel e das interrupções, o kernel deverá executar um desvio (JMP) para o endereço onde está o programa de aplicação, para que esta possa iniciar sua operação. No arquivo de referência (KERNEL_REF.CED) estão definidos os valores de vários símbolos. Um desses símbolos é o “_APP”. Esse símbolo está definido com o endereço onde estará carregada a aplicação. A instrução a ser executada pelo seu kernel é a seguinte:

```
JMP    _APP
```

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

Sua implementação deverá obedecer a seguinte divisão do espaço de endereçamento do CESAR:

Faixa de Endereços	Descrição
(H0000 a H7FFF)	Essa área é de uso exclusivo do kernel. Nessa área você deverá colocar todo o código de seu kernel, que será composto por: <ul style="list-style-type: none">• Inicialização do kernel (primeiras instruções a serem executadas pelo CESAR)• Inicialização de variáveis internas do kernel• Inicialização do sistema de interrupções• Implementação dos tratadores de interrupção• Implementação das funções de sistema (ver arquivo “FuncoesDoKernel.pdf” que acompanha essa especificação)• Definição da tabela de vetores (a partir do endereço H0100 até H01FF)• Definição da área de pilha (stack)
(H0100 a H01FF)	Área onde serão colocados os vetores das funções do kernel.
(H8000 a HFF7F)	Área para o programa de aplicação, que será fornecido pelo professor, para possibilitar o teste de seu kernel.
(HFF80 a HFFFF)	Área dos periféricos.

3. Detalhes da Tabela de Vetores

Nessa área estão os endereços que apontam para o início das funções de sistema, que você deverá implementar. Cada vetor, por se tratar de um endereço, ocupa dois bytes. O primeiro é o vetor[0], o segundo o vetor[1], e assim por diante.

O programa de aplicação que estiver carregado na memória (endereços H8000 até HFF7F) poderá chamar as funções de sistema através desses vetores. Assim, pode-se usar diferentes kernels com o mesmo programa de aplicação.

Todas as funções de sistema devem encerrar sua operação usando uma instrução “RTS R7”.

4. Correção e Entregáveis

A correção do trabalho será feita através de um programa de teste especialmente desenvolvido para esta finalidade.

Para realizar os testes, seu kernel será carregado na memória e, em seguida, será carregado o programa de aplicação de teste na área H8000 até HFF7F. Para isso, será usada a “carga parcial” disponível no simulador do CESAR16i. Então, será iniciada a execução e a verificação do comportamento correto do kernel.

Você deverá entregar um arquivo fonte (arquivo .CED) com a sua implementação do kernel, escrito em linguagem simbólica do CESAR16i, usando o montador Daedalus. O código do programa fonte deverá conter comentários descritivos da implementação.

O 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.