

Trabalho de Programação – Processador CESAR16i 2023/1

1. Descrição Geral

Nesse trabalho você deverá implementar o kernel e as funções da API (*Application Programming Interface*), usando o processador CESAR16i. 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”.

Após desenvolvido seu kernel, você deverá entregar **APENAS** o arquivo fonte com sua implementação (arquivo .CED).

Para o desenvolvimento do trabalho será colocado à disposição um arquivo fonte que, sugerimos, deve ser usado como base para o desenvolvimento do kernel. O nome deste arquivo é “KERNEL_REF.CED”.

Também serão disponibilizados arquivos com a implementação de aplicações, para facilitar o teste de sua implementação. Esses arquivos deverão ser carregados no simulador usando a carga parcial para os endereços H8000 até HFFBF. Essa carga parcial deve ser feita logo após seu kernel ter sido carregado.

O kernel deverá ser construído de maneira a suportar as interrupções de teclado e timer, assim como as funções de sistema descritas no arquivo “FuncoesDoKernel.pdf” que acompanha esta especificação.

Ao realizar a programação do seu kernel, e logo após encerrados todos os procedimentos de inicialização, sua implementação deverá executar um desvio (JMP) para o endereço onde está o programa de aplicação, de maneira a iniciar sua execução. Sugere-se que seja executada pelo kernel a seguinte instrução, onde “_APP” é um símbolo definido no arquivo “KERNEL_REF.CED” e corresponde ao início da aplicação:

JMP _APP

No arquivo de referência (KERNEL_REF.CED), além do símbolo “_APP”, estão definidos outros símbolos que correspondem à endereços da memória úteis ao desenvolvimento do kernel.

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)	Kernel Essa área é de uso exclusivo de seu kernel. Nessa área você deverá colocar TODO o código de seu kernel, incluindo variáveis locais. Seu kernel 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)	Tabela de vetores Área reservada para os vetores das funções do kernel. (ver detalhes a seguir).
(H8000 a HFF7F)	Programa de aplicação Área reservada para o programa de aplicação, que será fornecido pelo professor, para possibilitar o teste do kernel.
(HFF80 a HFFFF)	Periféricos 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 elemento desse vetor é o “vetor[0]”, o segundo é o “vetor[1]”, e assim por diante.

O programa de aplicação que estiver carregado na memória poderá chamar as funções de sistema através desses vetores. Assim, pode-se usar diferentes implementações do kernel com o mesmo programa de aplicação.

Finalmente, todas as funções de sistema devem encerrar sua operação usando uma instrução “RTS R7”. O motivo disso é que a chamada dessas funções será feita através de uma chamada de subrotina (JSR R7,endereço).

4. Correção e Entregas

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

Para realizar os testes, seu kernel será carregado na memória e, em seguida, carregado o programa de aplicação de teste. Para isso, será usada a “carga parcial” disponível no simulador do CESAR16i. Depois dessas duas cargas a execução será iniciada a partir do endereço H0000, e será verificado o comportamento correto do kernel.

Você deverá entregar um arquivo fonte (arquivo .CED) com a sua implementação do kernel, escrito em linguagem simbólica do CESAR16i, o qual será montado com 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. Trabalhos que tenham erro de montagem 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**).

Recomenda-se, também, que sejam usados os programas de exemplo disponibilizados durante as aulas para a construção de seu kernel.

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.