

Trabalho do CESAR (2019/2)

Parte 2

Descrição Geral

- Sua tarefa nessa parte do trabalho é desenvolver uma parte do Kernel, responsável pelo controle dos periféricos.
- Dessa forma, você deverá desenvolver funções para controlar
 - Visor
 - Teclado
 - Timer
- Além disso, você deve implementar a inicialização do processador (procedimentos de *reset*), que envolvem
 - Inicialização dos periféricos
 - Escrita dos tratadores de interrupção
 - Inicialização de quaisquer outras variáveis necessárias para as funcionalidades que você está desenvolvendo

Funções

- Função de inicialização
 - _reset
- Funções para teclado
 - _kbhit
 - _getchar
- Funções para visor
 - _putchar
 - _putmsg
 - _setcursor
- Funções para timer
 - _start
 - _ready
- A descrição de cada função pode ser encontrada no arquivo “FuncoesDoKernel.pdf”

Espaço de Endereçamento

- Para sua implementação, é importante entender como a memória está dividida
- Para fins desse trabalho, você pode imaginar a memória dividida em duas partes
 - Parte fornecida pelo professor
 - Corresponde aos endereços H0000 até H7FFF
 - Essa parte está no arquivo “app_prof_sem_kernel.mem”
 - Parte a ser desenvolvida
 - Corresponde aos endereços H8000 até HFFFF
 - Essa parte pode ser desenvolvida a partir do arquivo “kernel_ref.ced”

O quê desenvolver?

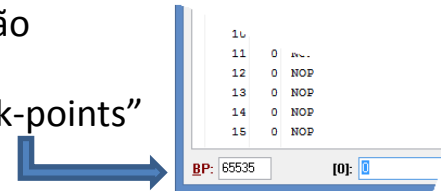
- A parte que lhe cabe desenvolver pode ser dividida em três partes
 - Área com os vetores de entrada (H8000 até H800F)
 - Área de código, dados e pilha (H8010 até HFF7F)
 - Área de periféricos (HFF80 até HFFFF)
- Os vetores de entrada são 8 (oito)
 - A forma de defini-los pode ser vista no arquivo “kernel_ref.ced” (ver “org_KRN_INT”)
 - Existe um vetor para cada função a ser implementada
- A área de periféricos
 - Nessa áreas estão os endereços dos periféricos, conforme estudados em aula
- Área de código, dados e pilha
 - Aqui você deverá colocar
 - A implementação de suas funções
 - A implementação dos tratadores de interrupção
 - As variáveis necessárias para a operação de seu kernel
 - A área reservada para a pilha do sistema (que será usada pelo kernel e pela aplicação)

Sobre o “kernel_ref.ced”

- Esse arquivo pode ser usado para desenvolver o seu kernel
- Ele contém, em ordem
 - Equates de símbolos de áreas de memória e de endereços de portas para periféricos
 - Definição da área de vetores (em `_KRN_INT`)
 - Função “`_reset`”
 - Nessa função você encontrará uma lista de atividades a serem implementadas, conforme seu projeto de kernel
 - Essa função deve encerrar com um “`JMP APP`”
 - Protótipos das outras funções
 - Cada função tem uma descrição do que ela deve fazer
 - Essas funções devem encerrar com um “`RTS R7`”

Como testar?

- Depois de montar e corrigir todos os erros, você deverá rodar seu kernel junto com o programa de aplicação fornecido pelo professor.
- Para isso, você deverá:
 1. Montar seu kernel com o Daedalus
 2. Carregar o arquivo “app_prof_sem_kernel.mem” no simulador do CESAR
 - Não esquecer de desligar o “atualizar registradores”
 3. Fazer uma carga parcial com o seu kernel
 - Os endereços a serem usados são 32768 a 65535
 4. Rodar o programa
 - O programa de aplicação é uma implementação da primeira parte do trabalho do CESAR
- Caso seja necessário depurar seu kernel
 - Lembre-se que a interrupção só é chamada quando em execução
 - O passo-a-passo não permite a chamada da interrupção
 - Para fazer parar dentro da interrupção, é necessário usar “break-points”



Arquivos Fornecidos

- TrabalhoCesar2019_2_Parte2.pdf
 - Descrição do trabalho
- FuncoesDoKernel.pdf
 - Descrição das funções do kernel
- app_prof_sem_kernel.mem
 - Programa de aplicação, fornecido pelo professor
- kernel_ref.ced
 - Fonte de referência, para iniciar o desenvolvimento
- Apresentacao.pdf
 - Esse conjunto de slides