Funções do Kernel – 2020/1

Este documento descreve as funções do kernel que deverão ser implementadas no trabalho com o processador Cesar, e também indica como as funções devem ser chamadas pelos programas de aplicação (fornecido pelo professor).

Essas funções permitem que o programa de aplicação possa utilizar os periféricos disponíveis no CESAR16i (teclado, visor e *timer*), sem a necessidade de conhecer o funcionamento do hardware e das portas de acesso a esses periféricos.

As funções a serem implementadas deverão ser colocadas na área de memória reservada para essa finalidade. Os primeiros bytes dessa área, que inicia no endereço H8000, deve conter os vetores que definem os endereços onde estão cada uma dessas funções. Cada vetor corresponde a uma função, e a ordem desses vetores deve ser obedecida, rigorosamente. Na tabela abaixo está apresentado o "Endereço do Vetor" (endereço onde deve ser colocado o endereço da função), o "Nome" da função e a "Descrição" da mesma:

Endereços do Vetor	Nome	Descrição
H8000 e H8001	_reset	Endereço da função que será chamada, na inicialização do programa. Essa chamada é realizada com um JMP. Portanto, ao final de sua implementação dos procedimentos de reset, deve ser realizado um JMP _APP (H0100), de maneira a iniciar a execução do programa de aplicação.
H8002 e H8003	_kbhit	Retorna em R0 o estado do teclado. Cada bit de R0 representa uma informação: • Bit 0: indica se há tecla; • Bit 1: indica se o estado SHIFT está ligado; • Bit 2: indica se o estado CTRL está ligado. • Os outros bits devem retornar, sempre, em zero. Essa função não é bloqueante. Ou seja, deve retornar imediatamente para o chamador, com a informação do teclado.
H0004 e H0005	_shift	Liga ou desliga o estado SHIFT, conforme o valor de R0: • Zero: desliga o estado SHIFT • Outro valor: liga o estado SHIFT
H0006 e H0007	_ctrl	Liga ou desliga o estado CTRL, conforme o valor de R0: • Zero: desliga o estado CTRL • Outro valor: liga o estado CTRL
H8008 e H8009	_getchar	Obtém o código ASCII de um caractere digitado no teclado e devolve o mesmo no registrador R0. Não "ecoa" o caractere lido no visor. Diferentemente da "_kbhit", permanece bloqueada, aguardando que algum caractere seja digitado. Retorna as letras conforme o estado SHIFT e CTRL: • Se ambos desligados, retorna sempre letras minúsculas (H61 – H7A); • Se CTRL ligado, converte letras em caracteres de controle correspondentes (valores ASCII entre H00 e H1A) • Se CTRL desligado e SHIFT ligado, converte letras para as correspondentes letras maiúsculas (valores ASCII entre H41 e H5A)
H800A e H800B	_putchar	Escreve um caractere no visor. Os parâmetros de entrada da função são os seguintes: • R0: ASCII do caractere a ser colocado no visor (entre H00 e H7E) • R1: Número do visor a ser usado (0 ou 1) • R2: Posição no visor selecionado (entre 0 e 17) Se a função for executada corretamente, retornar 0 (zero) em R0. Caso ocorra erro ou se os valores fornecidos na entrada sejam inválidos,

		a função deve ser ignorada e retornado valor diferente de 0 (zero) em R0.
H800C e H800D	_putmsg	Escreve uma mensagem no visor. Os parâmetros de entrada da função são os seguintes: • R0: Endereço do string a ser colocado no visor. O string deve ser terminado com "\0" (string "C"). • R1: Número do visor a ser usado (0 ou 1) • R2: Posição no visor selecionado (entre 0 e 17) Se a função for executada corretamente, retornar 0 (zero) em R0. Caso ocorra erro ou se os valores fornecidos na entrada sejam inválidos, a função deve ser ignorada e retornado valor diferente de 0 (zero) em R0. Caso o string seja maior do que o tamanho do visor (18 caracteres), o string deve ser truncado.
H800E e H800F	_start	Inicializa uma temporização. Inicializa o timer com o valor informado no R0. O valor de tempo informado em R0 deve ser em milissegundos. Se for chamada, novamente, antes do tempo atingir o seu final, um novo tempo será programado. Essa função deve ser usada em conjunto com a função _ready. Esse timer deve ter uma resolução de 10ms.
H8010 e H8011	_ready	Informa se o tempo programado pela função _start esgotou. Devolve 0 no registrador R0, caso o tempo programado com a função "_start" não tenha se esgotado. Retorna um valor diferente de 0, caso contrário.