

### Funções do Kernel – 2019/2

Este documento descreve as funções do kernel que deverão ser implementadas na segunda parte do trabalho com o processador Cesar e também indica como as funções devem ser chamadas pelos programas de aplicação.

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 16 (dezesesseis) bytes dessa área, que inicia no endereço H8000, devem conter os 8 (oito) 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, pelo código da região H0000 (fornecido pelo professor). Essa chamada é realizada com um JMP. Portanto, ao final de sua implementação dos procedimentos de reset, deve ser realizado um JMP _APP, de maneira a iniciar a execução do programa de aplicação.
H8002 e H8003	_kbhit	Devolve 0 no registrador R0 se nenhuma tecla foi pressionada e outro valor em caso contrário. Retorna ao programa chamador sem esperar pela digitação.
H8004 e H8005	_getchar	Obtém o código ASCII de um caractere digitado no teclado e devolve o mesmo no registrador R0. Não “eco” o caractere lido no visor. Enquanto não for digitado nenhum caractere, não retorna ao programa chamador.
H8006 e H8007	_putchar	Escreve no visor, na posição onde estiver o cursor, o caractere cujo código ASCII é informado nos 8 bits menos significativos do registrador R0. Os bits 8 a 15 do R0 são ignorados. O cursor “avança” para a posição seguinte no visor, exceto se já estiver na posição 35, caso em que o cursor deve “desaparecer”. Se o caractere a ser escrito for um “backspace” (código ASCII 8), o cursor deve ser recuado para a posição à sua esquerda, substituindo o caractere que estava naquela posição. Se o cursor já estiver na posição 0 do visor, o “backspace” deve ser ignorado.
H8008 e H8009	_putmsg	Escreve no visor, a partir da posição onde estiver o cursor, uma mensagem formada por uma sequência de caracteres ASCII terminada por um byte com valor zero (formato ASCIIZ). Escreve o cursor na posição do visor imediatamente após o último caractere da mensagem exibida (se a mensagem terminar na posição 35, o cursor “desaparece”). O byte terminador da mensagem (com valor 0) não deve ser exibido no visor. O endereço da memória onde inicia a mensagem deve ser fornecido no registrador R0. Se o espaço até o fim do visor for menor do que o tamanho da mensagem, interrompe a exibição da mesma na posição 35, fazendo o cursor “desaparecer”. Se a mensagem contiver um caractere “backspace”, o tratamento deve ser o mesmo descrito para a função PUTCHAR.
H800A e H800B	_setcursor	Altera a posição do cursor, representado pelo caractere “_”. Para isso, apaga o cursor da sua posição atual no visor e escreve o cursor na nova posição (informada no R0). Caso o valor em R0 seja maior do que 35, o cursor deverá “desaparecer”.
H800C e H800D	_start	Inicializa o timer com o valor informado no R0. O valor de tempo informado em R0 deve ser em milissegundos.
H800E e H800F	_ready	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.