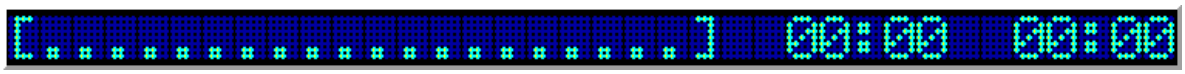
Kernel

Este documento descreve as estruturas a serem gerenciadas pelo Kernel assim como as funções da API de acesso ao Kernel, que deverão ser implementadas no trabalho com o Computador CESAR16i.

Gerência do Visor

O espaço do visor deverá ser gerenciado pelo kernel de forma a dividi-lo em três partes: (i) texto, (ii) relógio 0 e (iii) relógio 1.

- A parte "texto" corresponde às posições 0 (zero) até 21 (22 caracteres);
- A parte "relógio 0" corresponde às posições 22 até 28 (7 caracteres);
- A parte "relógio 1" corresponde às posições 29 até 35 (7 caracteres).



Na parte "texto" do visor devem ser colocados os caracteres recebidos no kernel através da função "_putchar". Os caracteres devem ser colocados na posição indicada pelo cursor. A operação do cursos está descrita nas funções "_getchar" e "_putchar".

Nas partes "relógio 0" e "relógio 1" devem ser apresentados dois relógios, um em cada área do visor. Cada relógio deve ter minutos e segundos, apenas. A ocupação no visor de cada área do relógio, formada por 7 posições, deve obedecer a seguinte formatação:

O kernel deverá controlar a visibilidade de cada relógio. Sempre que for digitado um caractere "," ou "<", o "relógio 0" deve ser tornado visível ou invisível, alternadamente; sempre que for digitado um caractere "." ou ">", o "relógio 1" deve ser tornado visível ou invisível, alternadamente.

Ao ser inicializado, os relógios devem estar invisíveis, parados (sem avançar) e zerados.

Funções do Kernel

A seguir, são descritas as funções da API e a forma como devem ser chamadas pelos programas de aplicação, incluindo os parâmetros de entrada e valores resultantes de saída.

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 o kernel.

Iniciando no endereço H0100, você deve definir a Tabela de Vetores do Kernel. Cada vetor (ponteiro, com 2 bytes) dessa tabela deve conter o endereço, no kernel, onde inicia a implementação da função correspondente ao vetor.

Observar que a ordem desses vetores deve ser, rigorosamente, obedecida. Os vetores e suas funções correspondentes está indicado abaixo:

Vetor	Função
[0]	_kbhit
[1]	_getchar
[2]	_putchar
[3]	_get_clock_status
[4]	_get_clock_time
[5]	_turnon_clock
[6]	_clr_clock

Abaixo você encontra a descrição das funções a serem implementadas.

1. Função: "_kbhit"

Função através da qual pode-se solicitar ao kernel a informação sobre a existência de tecla digitada. A função deve retornar com a informação da existência de tecla, sem aguardar pela digitação de qualquer tecla.

- *Parâmetros de entrada*: nenhum.
- *Parâmetro de saída*: registrador R0, com a informação da existência de tecla.

A função retorna no registrador R0 a informação se existe tecla ou não.

- Se há tecla, o valor em R0 será zero;
- Se não há tecla, o valor em R0 será um valor qualquer diferente de zero.

2. Função: "_getchar"

Função através da qual pode-se solicitar ao kernel que aguarde pela digitação de uma tecla. A função deve retornar o código ASCII da tecla digitada. Portanto, a função deve aguardar pela digitação de uma tecla.

- *Parâmetros de entrada*: nenhum.
- *Parâmetro de saída*: registrador R0, com a tecla digitada.

A função só retorna (só termina) quando o usuário digitar alguma tecla. O código ASCII da tecla digitada deve ser retornado no registrador R0.

Sempre que a função "_getchar" for chamada e estiver bloqueada aguardando por uma tecla, a posição do cursor (ver detalhes na função "_putchar" abaixo) deve ser apresentado no visor através do símbolo "_" (*underscore*). Esse símbolo deve ser alternado com o caractere que estiver sendo apresentado nessa posição. Essa alternância deve ter uma periodicidade tal que cada símbolo permaneça no visor por 500ms.

3. Função: "_putchar"

Envia um caractere ASCII para o visor. Esse caractere pode ser um caractere visível (que tenha representação simbólica) ou um caractere de controle.

- *Parâmetros de entrada*: registrador R5, com o caractere a ser colocado no visor.
- *Parâmetro de saída*: registrador R0, com o código de erro de retorno.

O código no registrador R5 pode representar caracteres visíveis ou caracteres de controle.

Os **caracteres visíveis** são aqueles cujos códigos iniciam em H20 (SPACE) e vão até H7A ("z"). Ao receber um desses caracteres, o kernel deve coloca-lo no visor, na posição indicada pelo cursor.

O cursor, cujo controle é do kernel, indica a posição onde deve ser escrito o caractere recebido através da função "_putchar". Esse cursor pode receber valores entre 0 (zero) e 21, que corresponde ao início e o término do cursor, respectivamente.

Após escrever um caractere na posição indicada pelo cursor, o kernel deverá incrementar a posição do cursor, preparando-o para a próxima escrita no visor. Caso a escrita tenha ocorrido na última posição do visor, o cursor não deverá ser incrementado, sendo mantido nessa posição final.

A tabela ASCII tem 32 **caracteres de controle**. Entretanto, nessa implementação do kernel será usado apena um deles. Os códigos dos caracteres de controle restantes devem ser ignorados. Os caracteres de controle a serem processados são os seguintes:

- CR (H0D) – *Carriage Return*: move o cursor para a primeira posição mais a esquerda do visor.
- BS (H08) – *Back Space*: move o cursor uma posição para a esquerda. Caso o cursor esteja no início do visor, esse caractere de controle deve ser ignorado.

Na inicialização do kernel, o visor deve ser limpo (apagado) e o cursor deve ser posicionado no início do visor.

A função retorna no registrador R0 um código de erro.

- Se não houve erro, o valor em R0 será zero;
- Se houve algum erro ou informação inválida de entrada, o valor em R0 será um valor qualquer diferente de zero.

4. Função: "_get_clock_status"

Essa função fornece as informações relativas à situação atual de cada relógio, no que diz respeito à sua visibilidade e se está andando.

- *Parâmetros de entrada*: registrador R5, com o identificador do relógio a ser lido. Esse valor pode ser 0 (zero) ou 1 (um). Outros valores devem ser ignorados.
- *Parâmetro de saída*: registrador R0, com a indicação do estado do relógio (visibilidade e se está parado).

No retorno da função, o registrador R0 conterá as informações do estado do relógio em seus bits 0 e bit 1.

O bit 0, quando ligado (valor "1"), indica que o relógio está visível; o bit 1, quando ligado (valor "1"), indica que o relógio está andando.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	Andando	Visível

Se houve algum erro ou informação inválida de entrada, o valor em R0 será o valor HFFFF (65535).

5. Função: "_get_clock_time"

Essa função fornece o valor atual de cada relógio. Essa informação é fornecida como um número inteiro de segundos. Como cada relógio tem minutos e segundos, o valor fornecido deverá ser o resultado da aplicação da seguinte fórula (60*MIN+SEG).

- *Parâmetros de entrada*: registrador R5, com o identificador do relógio a ser lido. Esse valor pode ser 0 (zero) ou 1 (um). Outros valores devem ser ignorados.
- *Parâmetro de saída*: registrador R0, com o número de segundos atualmente presentes no relógio.

No retorno da função, o registrador R0 conterá o número de segundos correspondentes à informação atual no relógio. Essa informação corresponde aquela presente no relógio, independentemente de seu estado (visível/invisível ou parado/andando).

Se houve algum erro ou informação inválida de entrada, o valor em R0 será o valor HFFFF (65535).

6. Função: "_turnon_clock"

Essa função é usada para indicar ao kernel se o relógio deve "andar" ou "parar".

- *Parâmetros de entrada*:
  - registrador R5, com o identificador do relógio a ser ativado/desativado. Esse valor pode ser 0 (zero) ou 1 (um). Outros valores devem ser ignorados;
  - registrador R4, com a indicação do estado a ser colocado o relógio. Esse valor pode ser 0 (zero), caso o relógio deva parar, ou um valor diferente de 0 (zero), caso o relógio deva andar.
- *Parâmetro de saída*: registrador R0, com o código de erro de retorno.

A função retorna no registrador R0 um código de erro.

- Se não houve erro, o valor em R0 será zero;
- Se houve algum erro ou informação inválida de entrada, o valor em R0 será um valor qualquer diferente de zero.

7. Função: "_clr_clock"

Essa função é usada para indicar ao kernel que o relógio deve ser zerado. Ou seja, minutos e horas armazenado no relógio devem ser zerados.

- *Parâmetros de entrada*: registrador R5, com o identificador do relógio a ser zerado. Esse valor pode ser 0 (zero) ou 1 (um). Outros valores devem ser ignorados;
- *Parâmetro de saída*: registrador R0, com o código de erro de retorno.

A função retorna no registrador R0 um código de erro.

- Se não houve erro, o valor em R0 será zero;

- Se houve algum erro ou informação inválida de entrada, o valor em R0 será um valor qualquer diferente de zero.