

SOFIA LIMA DOS SANTOS

LISTA - INTERRUPÇÃO 10H

(A parte prática desta lista está no arquivo “exercicio-int-10h” em <https://github.com/slimasofia/exercicios-assembly/tree/main/src>)

Quais são os serviços oferecidos por essa interrupção?

INT 10h, 0 - Definir modo de vídeo
INT 10h, 1 - Definir tipo de cursor
INT 10h, 2 - Definir posição do cursor
INT 10h, 3 - Ler posição do cursor
INT 10h, 4 - Ler posição da caneta de luz (Light Pen)
INT 10h, 5 - Selecionar página ativa de exibição
INT 10h, 6 - Rolar página ativa para cima
INT 10h, 7 - Rolar página ativa para baixo
INT 10h, 8 - Ler caractere e atributo na posição do cursor
INT 10h, 9 - Escrever caractere e atributo na posição do cursor
INT 10h, A - Escrever caractere no cursor atual
INT 10h, B - Definir paleta de cores
INT 10h, C - Escrever pixel gráfico na coordenada
INT 10h, D - Ler pixel gráfico na coordenada
INT 10h, E - Escrever texto no modo teletype
INT 10h, F - Obter estado atual do vídeo
INT 10h, 10 - Definir/obter registros de paleta (EGA/VGA)
INT 10h, 11 - Rotina do gerador de caracteres (EGA/VGA)
INT 10h, 12 - Configuração do subsistema de vídeo (EGA/VGA)
INT 10h, 13 - Escrever string (após 01/10/86)
INT 10h, 14 - Carregar fonte de caracteres LCD (convertível)
INT 10h, 15 - Retornar parâmetros físicos de exibição (convertível)
INT 10h, 1A - Combinação de exibição de vídeo (VGA)
INT 10h, 1B - Funcionalidade do BIOS de vídeo/Informação de estado (MCGA/VGA)
INT 10h, 1C - Salvar/Restaurar estado de vídeo (somente VGA)
INT 10h, FE - Obter buffer de regeneração de tela virtual DESQView/TopView
INT 10h, FF - Atualizar buffer de regeneração de tela virtual DESQView/TopView

Como os parâmetros são passados para a interrupção 10h:

São passados através dos registradores do processador.

AH: Contém o número do serviço a ser executado.

1. **AL, BH, BL, etc.:** Dependendo do serviço, outros registradores (como AL, BH, e BL) podem ser usados para passar parâmetros adicionais, como a cor do texto ou a posição do cursor.

Possíveis retornos:

Geralmente nenhum valor é retornado no registrador, as interrupções são usadas para realizar ações no hardware (exibir texto, mover o cursor, alterar o modo de vídeo), e o retorno pode ser um indicativo de sucesso ou falha geralmente através da alteração do estado de flags ou pelo efeito direto na tela.

Como funciona o mecanismo de interrupção no 8086:

O processador 8086 possui um mecanismo de interrupção controlado por uma tabela chamada Tabela de Vetores de Interrupção. Quando uma interrupção ocorre, o processador interrompe a execução do programa atual, salva seu estado (registro de ponteiro de pilha e o contexto atual) e começa a execução do manipulador de interrupção (um endereço específico fornecido na tabela de vetores de interrupção).

Registradores envolvidos no processo de interrupção:

Durante uma interrupção, os seguintes registradores são afetados:

- **Flags:** O processador salva o estado das flags de status para poder retornar ao estado anterior da execução.
- **CS (Code Segment) e IP (Instruction Pointer):** O processador salva o valor atual desses registradores para retomar a execução do código após o manuseio da interrupção.
- **Pilha:** O conteúdo dos registradores mencionados é empilhado, geralmente com os valores de CS e IP sendo armazenados na pilha, a fim de garantir que o processador retorne ao estado correto após o tratamento da interrupção.

Como a pilha é utilizada durante uma interrupção:

Quando ocorre uma interrupção, o 8086 realiza o seguinte procedimento:

1. A primeira coisa que o processador faz ao entrar em uma interrupção é empilhar o endereço de retorno, ou seja, o valor de **CS** (segmento de código) e **IP** (ponteiro de instrução).
2. O processador também salva o estado das **flags** (com a instrução **PUSHF**).
3. Após a execução do manipulador da interrupção, o processador desempilha os valores salvos da pilha (CS, IP e flags) e retoma a execução do programa que estava sendo executado antes da interrupção.