

Pesquisar a fundo a interrupção 10h:

a) Quais são os serviços oferecidos por essa interrupção? Os serviços da interrupção 10h são acessados através do registrador AH:

Função (AH)	Descrição	Parâmetros	Retorno
00h	Definir modo de vídeo	AL = Modo de vídeo (00h = texto 40x25, 01h = texto 80x25, etc.)	-
01h	Definir forma do cursor	CH = Linha superior, CL = Linha inferior	-
02h	Posicionar cursor	BH = Página, DH = Linha, DL = Coluna	-
03h	Obter posição do cursor	BH = Página	DH = Linha, DL = Coluna
04h	Obter estado da luz indicadora	AL = 00h	AH = Estado da luz indicadora
05h	Selecionar página ativa	AL = Página	-
06h	Rolar a janela para cima	AL = Linhas a rolar, BH = Atributo de preenchimento	-
07h	Rolar a janela para baixo	AL = Linhas a rolar, BH = Atributo de preenchimento	-
08h	Ler caractere e atributo na posição do cursor	BH = Página	AL = Caractere, AH = Atributo
09h	Escrever caractere com atributo	AL = Caractere, BH = Página, BL = Atributo, CX = Número de repetições	-
0Ah	Escrever caractere na tela (sem avançar cursor)	AL = Caractere, BH = Página, BL = Atributo	-
0Bh	Definir paleta de cores	BH = Página, BL = Código da cor	-

0Ch	Ler pixel (modo gráfico)	BH = Página, CX = Coluna, DX = Linha	AL = Cor
0Dh	Ler modo de vídeo atual	-	AL = Modo de vídeo, AH = N° de colunas, BH = Página ativa
0Eh	Escrever caractere no modo texto	AL = Caractere, BH = Página	-
0Fh	Obter modo de vídeo atual	-	AL = Modo de vídeo, AH = N° de colunas

https://en.wikipedia.org/wiki/INT_10H

<https://www.facom.ufu.br/~gustavo/OC1/Assembly/Interrupcoes%20da%20BIOS.pdf>

b) Como os parâmetros são passados para a interrupção?

Cada registrador tem um papel específico e carrega informações necessárias para que a interrupção execute a função desejada. Dentre todos os registradores o AH é o mais importante, com ele conseguimos definir qual serviço vai ser executado, ele é o responsável por informar a bios que serviço você deseja utilizar, então toda vez que você deseja realizar alguma tarefa mova o serviço para AH. Já os demais registradores funcionam como auxiliares ou parâmetros para o serviço que está sendo solicitado, como armazenar valores e combinações, vou mostrar alguns exemplos:

AX: Podemos utilizá-lo em situações com operações aritméticas e lógicas, entrada e saída, interrupções de sistema e manipulação de string e movimentação de dados. Um exemplo simples seria passar na parte alta do registrador (AH) o serviço de vídeo para exibir algo na tela e na parte baixa (AL) passar o caractere na qual desejamos imprimir.

BX: Este registrador tem um propósito geral, ele não possui funções específicas, porém tem um papel em operações com gráficos e texto, ele é usado em vários serviços especialmente para definir o comportamento de visualização em algumas interrupções de BIOS. Exemplos onde podemos usar para definir uma página de vídeo ativa, atributos de vídeo como cor de fundo, que será exibida manipulação do vídeo em modo texto usado para especificar a página, escrever um caractere com cor e cor de fundo.

CX: O registrador CX é comumente utilizado como um contador em loops e repetições, ou seja ele controla a quantidade de vezes em que podemos fazer uma determinada atividade. Podemos utilizar ele para imprimir uma quantidade de vezes um determinado caractere na tela.

DX: Ele pode ser usado como em situações gerais, mas é muito utilizado quando se trata de armazenar dados adicionais ou em operações que envolvam coordenadas, tamanhos e posições específicas na tela. Exemplo seria definir a posição do cursor gráfico ou desenhar linhas ou formas.

<https://wiki.osdev.org/BIOS>
<https://grandidierite.github.io/bios-interrupts/>

c) Quais são os possíveis retornos?

Algumas funções da interrupção 10h retornam informações nos registradores, como a posição do cursor ou o modo de vídeo atual. Coloquei os retornos na tabela acima.

Estudar a arquitetura do processador 8086:

a) Como funciona o mecanismo de interrupção no 8086?

No processador 8086, o mecanismo de interrupção é essencial e serve para lidar com situações inesperadas na qual permite parar a execução do programa para tratar de situações simples como entrada ou saída de dados e até mesmo erros, de uma forma geral ele interage com o hardware solicita a interrupção do programa, seja por hardware ou software o processador segue os passos:

Armazena e salva o conteúdo: Após a interrupção, o processador adiciona na pilha o conteúdo e informações como o valor do registrador (Instruction Pointer) que informa qual a próxima instrução a ser executada, também salva o valor do registrador (Code segment) para mostrar o segmento atual do código, e por último o registrador de flags que contém as informações do programa em no momento, jogando as informações na pilha conseguimos voltar de onde o programa parou.

Identifique o endereço da rotina de interrupção: As interrupções possuem um identificador único na qual é conhecido como vetor de interrupção esse, ele é usado para encontrar e localizar o endereço do serviço correspondente. Então ele armazena isso em uma tabela de vetores de interrupção

Desvia para a rotina de serviço: Isso ocorre quando o processador inicia a interrupção e essa rotina é uma instrução especial pois é ela quem trata dos eventos que causam interrupções, como o serviço de leitura do teclado ou envio de dados e até mesmo a correção de erros.

Finaliza a rotina de interrupção: Quando a função terminar, use a instrução IRET (retorno de interrupção). Retorna o IP, CS e Flags originais da lista, permitindo que o programa principal continue a execução de onde parou.

<https://hintron.github.io/8086-toolchain/stable/labs/InterruptMechanism.html>

<https://www.ic.unicamp.br/~celio/mc404-2005/interrupts-8086.html>

[http://inf.ufes.br/~zegonc/material/Sistemas_Operacionais/Mecanismo%20de%20Interrupcao%20\(new\).pdf](http://inf.ufes.br/~zegonc/material/Sistemas_Operacionais/Mecanismo%20de%20Interrupcao%20(new).pdf)

<https://examradar.com/8086-microprocessor-interrupt/>

b) Quais registradores estão envolvidos em uma interrupção?

Já havia dado uma introduzida no tópico anterior das informações agora vou falar um pouco mais :

IP (ponteiro de instrução): salva o endereço da próxima instrução a ser executada. Quando ocorre uma interrupção, seu valor é armazenado na pilha para que o programa possa ser reiniciado imediatamente após a execução.

CS (Código de Código): Indica o segmento de armazenamento de código atual. Seu valor também afeta a manutenção do projeto.

SP (Stack Pointer): Aponta para o topo da pilha e é usado para armazenar e restaurar IP, CS e Flags durante interrupções.

Flag : Contém informações sobre o status do programa (como restrições habilitadas ou desabilitadas). O valor do registrador de flags é retido para restaurar o sistema após uma interrupção.

AX, BX, CX, DX: Registradores gerais utilizados para programas de interrupção, utilizados para passar parâmetros, armazenar resultados ou armazenar objetos temporariamente.

C) Como a pilha é utilizada durante uma interrupção?

No processador 8086, a pilha é necessária para preservar o estado do programa em situações de interrupção. O procedimento é segmentado em três fases fundamentais:

Acumulação automática: Quando uma interrupção acontece, os valores dos registradores IP, CS e Flags são armazenados de forma automática na pilha. Isso assegura que o processador tenha a capacidade de prosseguir com a execução do programa onde foi interrompido.

Execução do programa de serviço: Ao executar o programa de interrupção, a pilha pode ser empregada para armazenar informações temporárias ou para registrar o estado de outros registradores gerais, tais como AX, BX, CX. Isso possibilita que a rotina opere sem alterar o contexto central do programa.

Recuperação com IRET: Após a execução da rotina, a instrução IRET é usada para remover os valores de Flags, cs e ip, trazendo de volta ao estado original.