**Aula 3**

**Alocação de memória**

Define ( reserva e define)

message DB 'WELCOME! '

OU

message DB 'W', 'E, 'L' , 'C', 'O' ,'M', 'E' , '!'

CÓDICO ASC QUEBRA DE LINHA 10 e 13

**Iniciação Múltiplas**

Times ---> permite múltiplas inicializaçõs do mesmo valor

marks TIMES 8 DW 0 ( reserva e define o valor para oito )

**Tabela de Símbolos**

Na tabela do slide, o offset representa quantidade de bytes DW --- Define word --- 2 bytes, sempre é o segundo.

Toda caractere conta 1 byte

***Diretiva EQU:***

Constante q n pode se alterar

**assign**

Posso alterar o valor

**%define**

redefinição e posso ter string também

**Transferência de Memória**

MOV

mov destination, sourse

Eu perco o valor no destination e peco no sourse

Nenhuma instrução , vai mover da memória p memória

Sempre terá que passar por um registrador

Parte mais importante da aula

Sintaxe : [] → recupera o valor que esta na posição da memória

sem [] é endereço da memória

response db 'y'

mov [response], 'n' ---> substituindo o conteúdo de response

os registradores também podem serem colocados em []...

mov [EBX],100 → significa o conteúdo daquela posição de memória vai receber 100

CODIGO OBRIGATÓRIO

mov eax,1 ; exit command to kernel

int 0x80 ; interrupt 80hex, call kernel