

SISTEMAS MICROPROCESSADOS I

Prof.: João Castelo



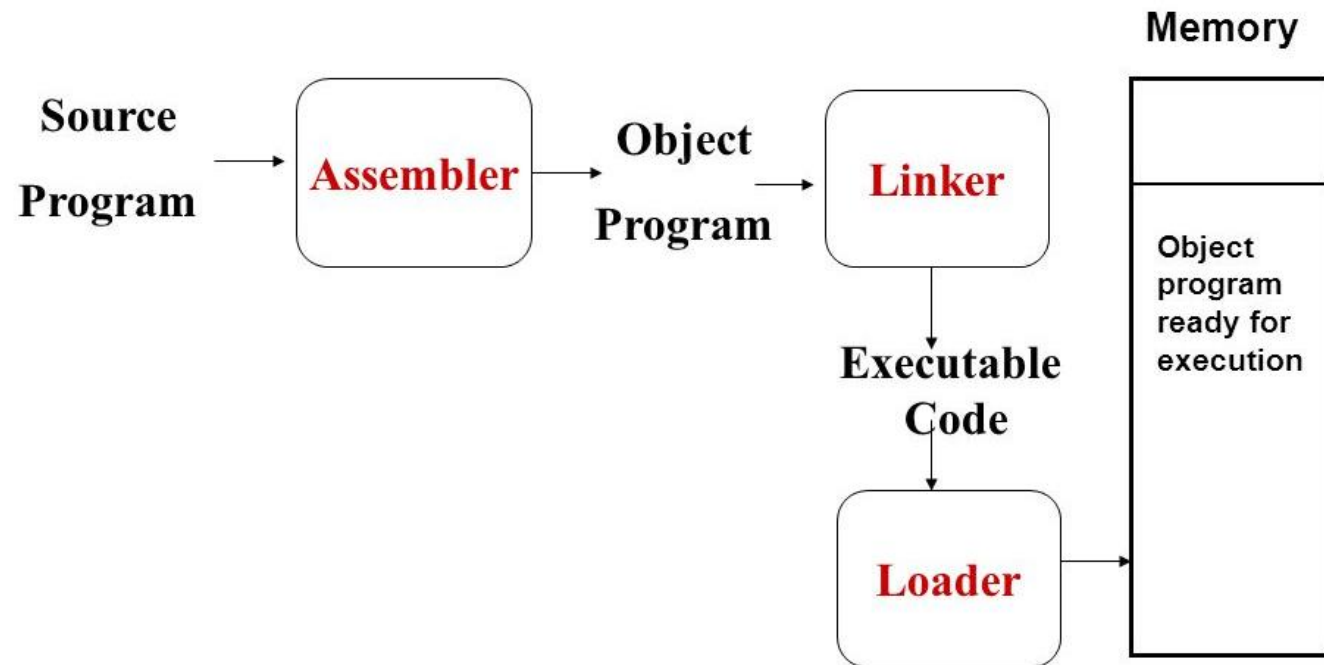
CENTRO UNIVERSITÁRIO
SENAI CIMATEC



Assembly

- Assembly é uma linguagem de baixo nível, chamada frequentemente de “linguagem de montagem”.
- É uma linguagem considerada difícil, principalmente porque o programador precisa conhecer a estrutura da máquina para usá-la.

Assembler



Assembler

- Um assembler traduz programas em linguagem assembly em código de máquina.
- A saída de um assembler é chamada de arquivo de objeto (object file), que contém uma combinação de instruções de máquina, bem como os dados necessários para colocar essas instruções na memória.

Linker

- Linker é um programa de computador que vincula e mescla vários arquivos-objeto para formar um arquivo executável.
- Todos esses arquivos podem ter sido compilados por montadores separados.
- A principal tarefa de um vinculador é pesquisar e localizar módulos / rotinas referenciados em um programa e determinar a localização da memória onde esses códigos serão carregados, fazendo com que a instrução do programa tenha referências absolutas.

Loader

- Loader é uma parte do sistema operacional e é responsável por carregar arquivos executáveis na memória e executá-los.
- Calcula o tamanho de um programa (instruções e dados) e cria espaço de memória para ele.
- Inicializa vários registros para iniciar a execução.

Vamos analisar o código...

```
1  #include "msp430.h"           ; #define controlled include file
2
3      NAME    main              ; module name
4
5      PUBLIC  main              ; make the main label visible
6                               ; outside this module
7
8      ORG     0FFFFh
9
10     RSEG    CSTACK             ; pre-declaration of segment
11     RSEG    CODE               ; place program in 'CODE' segment
12
13 main:
14                               ; main program
15     MOV.W   #WDTPW+WDTHOLD,&WDTCTL ; Stop watchdog timer
16
17
18
19     PUSH.W  #0006h             ;armazena o conteúdo indicado pelo operando fonte no topo da pilha
20     PUSH.W  #0004h
21     PUSH.W  #0002h
22     PUSH.W  #0000h
23
24     POP.W   R6                 ;retirar um valor armazenado na pilha
25     POP.W   R6
26     POP.W   R6
27     POP.W   R6
28
29     JMP     $                  ; jump to current location '$'
30                               ; (endless loop)
```


Include

- Esta é uma diretiva de pré-processador de estilo C que especifica um arquivo de cabeçalho a ser incluído na fonte.
- O arquivo de cabeçalho inclui todas as definições de macro, por exemplo, endereços de registro de função especial (WDTCTL) e bits de controle (WDTPW + WDT HOLD).

ORG

- Use a diretiva ORG para definir a localização do PC (Program Counter) do segmento atual para o valor de uma expressão definida.

RSEG

- RSEG é uma diretiva de montador de controle de segmento que controla como o código e os dados estão localizados na memória.
- RSEG é usado para marcar o início de um código ou segmento de dados relocável.
- CODE e DATA são tipos de segmento reconhecidos que são resolvidos pelo linker.
- O vinculador IAR XLINK pode reconhecer qualquer outro tipo de segmento (por exemplo, CSTACK para pilha de código).

Tarefas para inicialização

- Principais tarefas para inicializar o código em Assembly:
 - Inicializar o apontador da pilha (SP);
 - Desligar o WatchDog Timer (WDT);
 - Fazer a inicialização do vetor de reset - RESET_VECTOR.

WDT

Figure 16-2. WDTCTL Register

15	14	13	12	11	10	9	8
WDTPW							
7	6	5	4	3	2	1	0
WDTHOLD	WDTSSEL		WDTTMSSEL	WDTCNTCL	WDTIS		
rw-0	rw-0	rw-0	rw-0	r0(w)	rw-1	rw-0	rw-0

Table 16-2. WDTCTL Register Description

Bit	Field	Type	Reset	Description
15-8	WDTPW	RW	69h	Watchdog timer password. Always read as 069h. Must be written as 5Ah; if any other value is written, a PUC is generated.
7	WDTHOLD	RW	0h	Watchdog timer hold. This bit stops the watchdog timer. Setting WDTHOLD = 1 when the WDT is not in use conserves power. 0b = Watchdog timer is not stopped. 1b = Watchdog timer is stopped.
6-5	WDTSSEL	RW	0h	Watchdog timer clock source select 00b = SMCLK 01b = ACLK 10b = VLOCLK 11b = X_CLK; VLOCLK in devices that do not support X_CLK
4	WDTTMSSEL	RW	0h	Watchdog timer mode select 0b = Watchdog mode 1b = Interval timer mode
3	WDTCNTCL	RW	0h	Watchdog timer counter clear. Setting WDTCNTCL = 1 clears the count value to 0000h. WDTCNTCL is automatically reset. 0b = No action 1b = WDTCNT = 0000h
2-0	WDTIS	RW	4h	Watchdog timer interval select. These bits select the watchdog timer interval to set the WDTIFG flag and/or generate a PUC. 000b = Watchdog clock source $/ (2^{31})$ (18h:12m:16s at 32.768 kHz) 001b = Watchdog clock source $/ (2^{27})$ (01h:08m:16s at 32.768 kHz) 010b = Watchdog clock source $/ (2^{23})$ (00h:04m:16s at 32.768 kHz) 011b = Watchdog clock source $/ (2^{19})$ (00h:00m:16s at 32.768 kHz) 100b = Watchdog clock source $/ (2^{15})$ (1 s at 32.768 kHz) 101b = Watchdog clock source $/ (2^{13})$ (250 ms at 32.768 kHz) 110b = Watchdog clock source $/ (2^9)$ (15.625 ms at 32.768 kHz) 111b = Watchdog clock source $/ (2^6)$ (1.95 ms at 32.768 kHz)