Arquitetura e Organização de Computadores

Prática de Programação em Assembly

Givanaldo Rocha de Souza
http://docente.ifrn.edu.br/givanaldorocha
givanaldo.rocha@ifrn.edu.br

Modelo de Programação

- **EAX** – Acumulador em operações aritméticas, lógicas, E/S etc.
- **EBX** – Base para instruções com vetores de dados.
- **ECX** – Contador em operações iterativas e repetitivas.
- **EDX** – Dados utilizados nas operações de multiplicação e divisão.
- **ESI** – Endereço fonte em instruções de manipulação de vetores.
- **EDI** – Endereço destino em instruções de manipulação de vetores.

Modelo de Programação

- Registros de Segmentos
  - CS – Indica o segmento de código;
  - DS – Indica o segmento de dados;
  - ES – Indica o segmento extra de dados;
  - SS – Indica o segmento para a pilha;
- IP – ponteiro de instruções;
- SP – ponteiro da pilha;
- BP – ponteiro para posições de memória;
- FLAGS – bits de status e controle;

Modos de Endereçamento de Dados

- Registro – MOV AL, BL
- Imediato – MOV AL, 28H
- Direto – MOV [1234], AL
- Indireto – MOV [BX],CL
- Base+Index – MV [BX+SI],BP
- Relativo – MOV CL,[BX+4]
- Relativo Base + Index – MOV ARRAY[BX+SI], DX

Conjunto de Instruções – Resumo

- MOV dest, src – MOV AL, BL
- INT num – INT 21H
- ADD dest, valor – ADD AL, 10h
- SUB dest, valor – SUB AL, 2h
- DEC reg – DEC BX
- INC reg – INC BX
- JMP POS – JMP FIM_PROC
- CALL POS – CALL LER_INT
- LOOP LABEL – MOV CX, 03H; LOOP M_LABEL;
- LODSB – carrega um byte da memória
- LODSW – carrega um word da memória
- STOSB – salva um byte na memória
- STOSW – salva um word na memória

Conjunto de Instruções – Resumo

- CMP AX, BX
- Comparação sem Sinal
  - JA – Jump AX > BX
  - JAE – Jump AX >= BX
  - JB – Jump AX < BX
  - JBE – Jump AX <= BX
  - JNA – Jump !(AX > BX)
  - JNAE – Jump !(AX >= BX)
  - JNB – Jump !(AX < BX)
  - JNBE – Jump !(AX <= BX)
  - JZ – Mesmo que JE
  - JE – Jump AX == BX

Exemplo de Código – MASM

```
.MODEL SMALL
.STACK 100H
.DATA
    Str1 db “HELLO WORLD”,13,10,’$’
.CODE
    ;atualiza o DS com o segmento que guarda STR1
    MOV AX,SEG Str1
    MOV DS, AX
    ;chama a INT 21 para imprimir a string na tela
    MOV DX, OFFSET Str1
    MOV AH, 09
    INT 21H
    ;encerra o programa
    MOV AX, 4c00H
    INT 21 H
END
```

Gerando Executável

- >> masm.exe /c /l arquivo.asm arquivo.obj arquivo.lst arquivo.crf

- >> link.exe arquivo.obj
  - Run File [arquivo.exe]: arquivo.exe
  - List File [null.map]: arquivo.map
  - Libraries [.LIB]:

- **O Visual Studio também pode ser utilizado.**

Atividade

- Ver os códigos de exemplo do livro “Assembly Language for x86 Processors (6th edition)” de Kip Irvine.

- Tentar montar seus próprios códigos.