	Laboratório de Arquitetura de Computadores	Data:29/10/19
	Relatório de Experimentos	Módulo: 2

Nome: Giovanne Prestes Dias

RA 171029

Título: Arquitetura x86 – Assembly PC.

Objetivos:

- Adquirir conhecimentos em arquitetura de computadores;
- Uso do ambiente do Visual Studio;
- Instruções básicas usando linguagem Assembly;
- Manipulação de dados em registradores e memória.

Material Utilizado:

- Software Visual Studio.

Relatório:

Projeto a ser desenvolvido:

Criar um relógio utilizando Assembly x86 que conte os minutos “00 até 59” e os segundos “00 até 59”, quando chegar no final 59 59, recomeçar;

1) Introdução com a apresentação dos assuntos abordados no experimento;

Criada na década de 50, o Assembly foi das primeiras linguagens de programação a aparecer. Ela usa uma sintaxe complicada e "exageradamente" difícil, isto porque, antes da década de 50 os programadores de máquinas tinham que escrever instruções em código binário, qualquer coisa como: 0110010110011011010110011010111010110101 ... Para escrever uma instrução. Na verdade, o Assembly foi criado para facilitar o uso dessa tarefa, mas é considerado uma linguagem de baixo nível, pois tudo o que o processador interpreta tem que ser descrito pelo programador. Assim o código acima seria "add EAX" em Assembly. Bastava apenas, depois de estar concluída a escrita do código, rodar o compilador e tínhamos o programa.

Cada arquitetura de computador tem seu próprio código de máquina, e cada montador gera códigos para uma arquitetura específica. Cada um desses montadores tem sua própria versão de código Assembly, que pode diferir ao uso de registradores, representação de números, ou até mesmo instruções mnemônicas. E isso pode dificultar um pouco na portabilidade do código, tendo em vista que o mesmo precisaria ser reescrito para poder ser montado para outra arquitetura.

- Vantagens: programas extremamente rápidos e pequenos
- Desvantagens: tempo de desenvolvimento lento e sujeito a erros; código preso a uma arquitetura.

A Arquitetura x86 foi desenvolvida pela Intel Corporation no início de 1978, e se refere a uma família de processadores que tem como base o processador 8086 da Intel. O nome “x86” deriva dessa família, pois todos os outros processadores que vieram depois foram identificados com final 86.

Os primeiros processadores da família x86 eram baseados em uma arquitetura de 16 bits, porém com o tempo e com o avanço tecnológico, a x86 passou a trabalhar também com 32 bits.

Notação utilizada:

- <reg32>: Qualquer registrador de 32 bits: EAX, EBX, ECX, EDX, ESI, EDI, ESP ou EBP;
- <reg16>: Qualquer registrador de 16 bits: AX, BX, CX ou DX;
- <reg8>: Qualquer registrador de 8 bits: AH, BH, CH, DH, AL, BL, CL ou DL;
- <reg>: Qualquer registrador;
- <mem>: Endereço de memória;
- <con32>: Constante de 32 bits;
- <con16>: Constante de 16 bits;
- <con8>: Constante de 8 bits;
- <con>: Constante (qualquer tamanho).

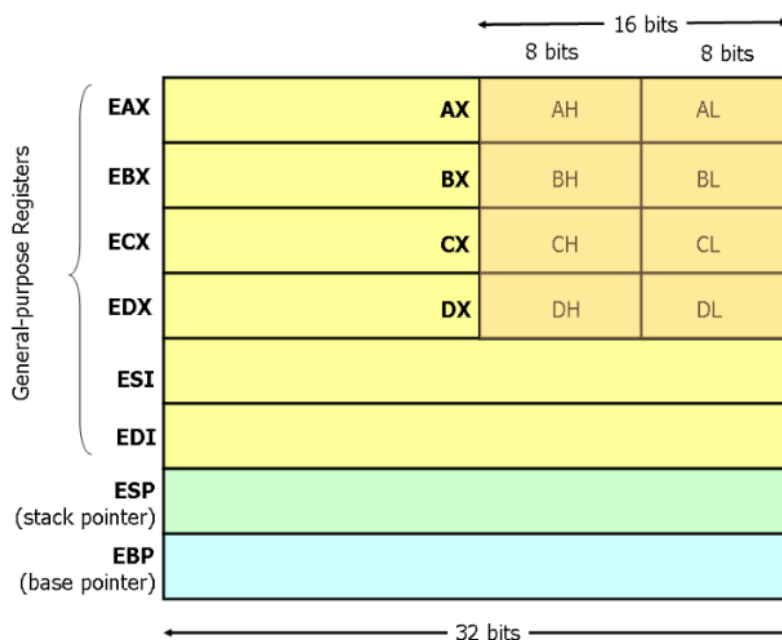



Figura 1: Registradores

2) Procedimento experimental executado (quais foram os passos executados para a criação do projeto);

Com o roteiro disponibilizado pelo professor, consegui criar um projeto no Visual Studio, para que a IDE conseguisse rodar Assembly a partir do C++.

Com o projeto criado, fui implementando as lógicas necessárias para a criação do objetivo deste relatório o relógio que conta os minutos e segundos; onde começam com 0 minutos e 0 segundos e terminam em 59 minutos e 59 segundos, após eles chegarem no término, recomeçar.

	Laboratório de Arquitetura de Computadores	Data:29/10/19
	Relatório de Experimentos	Módulo: 2

3) Análise dos resultados (acrescentar comentários sobre o funcionamento do projeto);

Com as instruções informadas acima e o desafio proposto, cheguei em uma lógica como resultado descrito na função a seguir:

```
void Relogio(void) {
    int minutos = 0;
    int segundos;

    __asm {
        loop_minutos:
            mov ecx, 0
            mov segundos, 0
            loop_segundos:
                add segundos, 1
                inc ecx
                cmp ecx, 59
                jl loop_segundos
            add minutos, 1
            cmp minutos, 59
            jl loop_minutos
    }
    Relogio();
}
```

4) Conclusão.

Por fim, concluí que para o desafio proposto este código alcançou o esperado e, portanto, o relógio conta os minutos de “00 até 59”, os segundos de “00 até 59”, e quando chega no final “59 e 59” ele recomeça.

Ao final deste experimento, obtivemos um conhecimento básico sobre assembly, mais conhecimento em arquitetura de computadores, utilizando a arquitetura x86.