

2º Relatório de Laboratório de Montagem

Bruno Cardoso Maciel RA: 141046279

Rafael Stefanini Carreira RA: 141040726

07/10/2015

Introdução

Através de um valor informado pelo usuário, o objetivo deste trabalho é comparar se o mesmo é menor que 100, se verdadeiro, calcular o quadrado do número e mostrá-lo, caso contrário, exibir uma mensagem informando que o número é maior que 100.

Para tal, devemos enfatizar que os dados lidos através da *syscall read* são tratados como cadeia de caracteres, ou seja, cada algarismo é tratado internamente como o valor numérico que o representa na tabela ASCII. Dessa forma, para realizar manipulações numéricas com tais dados é necessário convertê-los para um valor numérico único. Tal conversão pode ser feita através de multiplicações e somas acumuladas. Finalmente, para representar novamente os valores numéricos em forma de cadeia de caracteres, ou seja, cada algarismo do número representado por um caractere, devemos realizar divisões suscetivas e utilizar o resto de cada divisão na ordem inversa para representar os caracteres do número desejado.

Metodologia

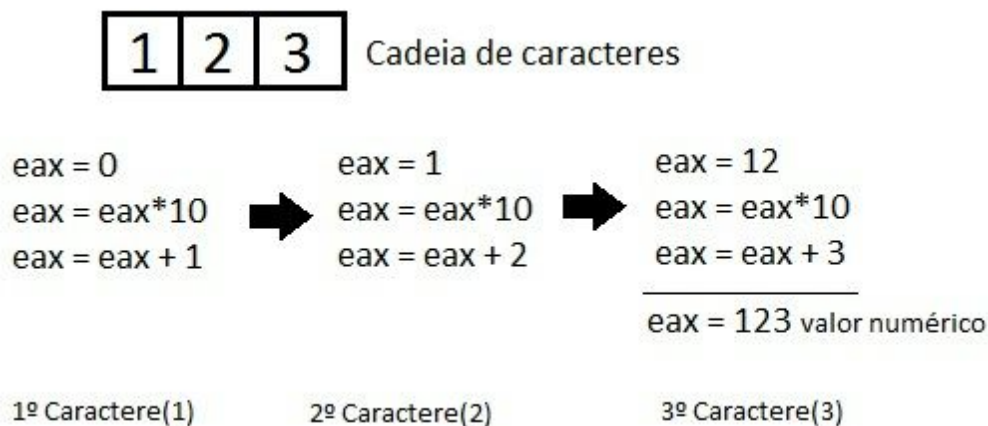
Solicitamos um número de até 4 dígitos ao usuário. O valor informado é armazenado na forma de cadeia de caracteres através da *syscall read*. Antes de verificar se o número é menor que 100, devemos convertê-lo em um valor numérico. Para tal processo utilizamos os registradores *eax* (acumulador), *ebx* (contador), *ecx* (multiplicador) e *edx* (recebe cada caractere da cadeia).

Primeiramente inicializamos o registrador *ecx* em 10 e os demais registradores com o valor 0. Por se tratar de um processo iterativo, a cada execução, o valor contido em *eax* é multiplicado por *ecx*, em seguida, o próximo caractere é colocado em *dl* (1 byte de *edx*) e convertido em valor numérico (*sub dl,"0"*) e o mesmo é somado em *eax*.

Para que o registrador *dl* receba o próximo caractere, a cada iteração o registrador *ebx* é incrementado. O processo termina quando o ultimo caractere contém o valor 10 em

hexadecimal, representando o final da linha. Por fim, o valor numérico resultante é acumulado em `eax`.

O processo pode ser ilustrado da seguinte forma:



Uma vez com o valor convertido, verificamos se o mesmo é maior ou igual a 100. Para realizar tal comparação utilizamos a instrução “`cmp`” passando como parâmetro o valor convertido e o valor 100, em seguida utilizamos “`jge`” passando uma *label* de destino como parâmetro. A instrução “`jge`” (Jump Greater or Equal) verifica o resultado da comparação anterior, se o primeiro valor for maior que o segundo, ocorre um salto na execução do programa para a *label* informada, que por sua vez, mostra na tela uma mensagem ao usuário dizendo que o número informado é maior que 100 e o programa é encerrado. Caso contrário, não é realizado nenhum salto, portanto, segue-se a execução do mesmo.

Em seguida é calculado o quadrado do valor digitado pelo usuário. O cálculo pode ser realizado através da instrução “`mul eax`”, que por sua vez, multiplica o valor passado como parâmetro pelo valor em `eax`, ou seja, é calculado o quadrado do valor contido em `eax`.

Um vez calculado, o próximo passo é informá-lo ao usuário. Lembramos que o número resultante está na forma numérica, portanto é preciso convertê-lo em uma cadeia de caracteres para mostrá-lo. Para realizar tal conversão utilizamos os registradores de forma semelhante, porém, o registrador `ecx` funciona como divisor, fixado com valor 10.

A cada iteração, o valor em `eax` é dividido por `ecx` e o resto da divisão é armazenado em `edx`, que por sua vez é convertido em caractere (ASCII) e salvo em uma variável “`resul`”. O quociente fica armazenado em `eax` para as futuras iterações e `ebx` é incrementado para armazenar os restos de forma contínua (sem sobreposições).

Ao fim processo, os restos das divisões estarão armazenados em uma variável, representando o número em forma de caracteres, porém, em ordem inversa. Portanto devemos

imprimí-lo na ordem inversa, do último caractere até o primeiro. Para tal, utilizamos o registrador ebx, que ao fim do processo de conversão, conterá o tamanho da cadeia de caracteres formada, e dessa forma, a cada impressão da variável “resul” na posição ebx, o seu valor é decrementado. O processo de impressão dos elementos da variável chega ao fim quando o contador ebx chega a 0, dessa forma temos o número impresso na tela, encerrando assim o programa.

Conclusão

Concluimos que através de operações como multiplicação e divisão podemos converter uma cadeia de caracteres em um valor numérico e vice-versa. Tal procedimento é útil para podermos manipular dados digitados pelo usuário, sendo úteis também para representarmos os dados resultantes de cálculos numéricos em forma de caracteres, podendo assim, ser informados novamente ao usuário.