

Assembly: Operações aritméticas e lógicas

1. Neste laboratório vamos começar a programar em linguagem de montagem. Baixe o programa “prog.s” do Moodle.

Obs: Alguns trechos estão marcados como “não mexa”, eles servem para imprimir o número que tiver sido armazenado no registrador “%eax”.

Para compilar esse programa, basta chamar o GCC:

```
gcc -o prog prog.s
```

(Pela extensão “.s” o GCC sabe que o programa está em Assembly.)

Para executá o programa, faça como sempre:

```
./prog
```

Compare o código Assembly com o código C abaixo e veja se você consegue fazer uma correspondência entre eles.

```
int nums[] = {10, -21, -30, 45};
int main() {
    int i, *p;
    for (i = 0, p = nums; i != 4; i++, p++)
        printf("%d\n", *p);
    return 0;
}
```

2. O programa C abaixo imprime a soma dos elementos do array. Traduza este programa para Assembly, usando “prog.s” como base.

Obs: Na sua modificação, mantenha o contador “i” em “ebx” e o endereço de “nums” em “r12”.

```
int nums[] = {10, -21, -30, 45};
int main() {
    int i, *p;
    int sum = 0;
    for (i = 0, p = nums; i != 4; i++, p++)
        sum += *p;
    printf("%d\n", sum);
    return 0;
}
```

3. O programa em C abaixo imprime os elementos **pares** do array. Traduza este programa para Assembly, usando “prog.s” como base.

```
int nums[] = {10, -21, -30, 45};
int main() {
    int i, *p;
    for (i = 0, p = nums; i != 4; i++, p++)
        if ((*p % 2) == 0)
            printf("%d\n", *p);
    return 0;
}
```

Dicas:

- Use uma operação lógica para testar se o elemento é par, como fizemos em laboratórios anteriores.

4. Usando “prog.s” como base, modifique o array de inteiros de 32bits para um array de bytes (char). Faça as modificações necessárias para percorrer e mostrar os valores do array de bytes corretamente na tela.

Obs: O valor a ser impresso por “printf” é um inteiro de 32 bits armazenado em “eax”. Você deve estender cada byte para o valor equivalente de 32 bits antes de imprimi-lo.