Praticas 01

O objetivo desta prática é aprender os passos básicos para montar, linkar (ligar) e depurar um programa escrito em *gnu assembly*, bem como verificar a execução de algumas instruções em assembly para a **arquitetura IA-32**.

No final desta explicação existe um pequeno exemplo, introdutório, de um programa fonte, escrito na linguagem "***gnu assembly***", para você testar. O arquivo do programa fonte exemplo aqui é chamado de "praticas_01.s", mas você poderá utilizar outro nome com outra extensão se quiser.

Execute os seguintes passos:

1) Digite o programa fonte, que está no final deste documento, em um editor de texto qualquer, por exemplo, o ***"gedit".*** Para encontrar o *gedit* abra um terminal e execute o programa *gedit*, ou, procure um editor de texto no menu de acessórios ou em outro menu do seu ambiente linux. Depois que você digitar o programa fonte, salve-o com o nome "praticas_01.s" na sua área de trabalho.

2) Com o terminal aberto, faça a montagem do programa fonte, ou seja, gere o programa objeto, executando a seguinte linha de comando no ***prompt*** do terminal. Observe que o diretório corrente deverá ser o mesmo que está o programa fonte. Para verificar se o programa fonte está no diretório corrente, use o comando ***ls*** do linux. Se não for, mude para ele usando o comando ***cd*** do linux.

as praticas_01.s -o praticas_01.o

"***as***" é o nome do montador da linguagem gnu assembly. Nesse caso, o programa objeto gerado terá o nome "praticas_01.o", mas você poderia usar outro nome.

3) Faça a linkagem (ligação) do programa objeto, ou seja, finalize o processo de geração do programa executável efetuando os ajustes de endereços e a ligação com códigos de biblioteca, caso sejam utilizados, usando o seguinte comando:

ld praticas_01.o -o praticas_01

"***ld***" é o nome do linkador do gnu assembly. Nesse caso, o executável gerado terá o nome "praticas_01", sem extensão, mas poderia ser diferente.

4) Execute o código executável gerado, digitando:

./praticas_01

5) Verifique oque aconteceu na execução.

Certamente a execução não mostrara nada, pois o programa não apresenta instruções de saída de dados, nem de entrada de dados. Para saber oque aconteceu na execução, a execução pode ser depurada.

6) Para depurar o programa, monte novamente da seguinte forma:

as -gstabs praticas_01.s -o praticas_01.o

O parãmetro “***-gstabs***” insere controles no código objeto gerado de forma a possibilitar que o programa seja depurado pelo aplicativo “***gdb***”.

7) Faça novamnte a linkagem da forma tradicional, conforme segue:

ld praticas_01.o -o praticas_01

8) Execute novamente o executável usando o gdb, da seguinte forma:

gdb praticas_01

Durante a execução dentro do *gdb*, use o comando "***help***" para saber quais são as classes de opções do *gdb*; depois você pode usar o comando *help* seguido do nome da classe para saber quais são as opções da classe; depois você pode usar o comando *help* seguido do nome da opção para saber sobre a ela.

Alguns comandos gdb são bem uteis, tais como: *run*, *break*, *cont*, *info all-registers* e *print* seguido do nome do rótulo. Leia sobre eles.

9) Experimente executar os seguintes comandos, um após o outro, no prompt do *gdb*:

(gdb) help <enter>
(gdb) help info <enter>
(gdb) help break <enter>
(gdb) help print <enter>

Para depurar um programa no *gdb*, você precisa antes definir os pontos de checagem, ou seja, os pontos de *breakpoint*. Para isso use o comando *break* seguido do nome de um rotulo dentro do programa.

10) Crie 3 pontos de checagem executando a seguinte sequencia de comandos dentro do *gdb*:

(gdb) break _start
(gdb) break _passo1
(gdb) break _fim

Note que "*_start*", "*_passo1*" e "*_fim*" são nomes de rótulos. Os rótulos são marcadores de posição de memoria, que podem ser usados na área de dados ou na de código. Na área de dados são interpretados como variáveis.

11) Execute o programa “praticas_01” até o primeiro ponto de checagem e veja os conteúdos dos registradores do processador executando os seguintes comandos:

(gdb) run
(gdb) info all-register

12) Continue executando a partir do último ponto de checagem e continue verificando os conteúdos dos registradores executando os seguintes comandos:

(gdb) cont
(gdb) info all-register

```
(gdb) cont
(gdb) info all-register
```

13) Veja os conteúdos de variáveis, de endereços de variáveis e de registradores em específico:

```
(gdb) print x
(gdb) print &x
(gdb) info address x
(gdb) info reg eax
```

14) continue testando, por conta própria, alguns outros comandos do *gdb*. Use o comando *help* para descobrir outros comandos.

Para sair do *gdb* digite *quit*, ou somente q, aliás, no *gdb*, os comandos podem ser simplificados pelas primeiras letras que o identificam unicamente.

==========

Programa fonte "praticas_01.s"

==========

```
.section .data

x:  .int   10
y:  .int    5
z:  .int   -1

.section .text

.globl _start

_start:
    movl    x, %eax

_passo1:
    movl    $x, %ebx

_passo2:
    movl    %eax, y

_passo3:
    movl    %ebx, y

_passo4:
    addl    %eax, %ebx

_passo5:
    addl    %ebx, %eax

_fim:
    movl    $1, %eax
    movl    $0, %ebx
    int     $0x80
```