Praticas 07b

O objetivo do programa eh ler n números inteiros e armazena-los em um vetor/buffer. O vetor deve ser declarado estaticamente usando a diretiva .space, podendo armazenar até 20 números inteiros (4 bytes). Então, troque entre si os elementos em posicoes opostas correspondentes (o primeiro com o ultimo, o segundo com o penultimo etc). Para acessar as posições do vetor, seu endereço inicial é colocado no registrador %edi, o qual deve ser incrementado de 4 em 4 para saltar entre as posições do vetor. A instrução loop é usada tanto na leitura dos n elementos quanto no caminhamento dos n elementos do vetor. O vetor original e o invertido são mostrados para o usuário.

Para gerar o executável, gere primeiro o objeto executando o seguinte comando:

```
as praticas_07b.s -o praticas_07b.o
```

e depois link dinamicamente com o seguinte comando:

```
ld praticas_07b.o -l c -dynamic-linker /lib/ld-linux.so.2 -o praticas_07b
```

O executavel se chamara praticas_07b, sem extensão, e para executá-lo digite:

```
./praticas_07b
```

```
.section    .data

titulo:     .asciz      "\n*** Programa Inverte Vetor 1.0 ***\n\n"

pedetam:    .asciz      "Digite o tamanho do vetor (maximo=20) => "

formato:    .asciz      "%d"

pedenum:    .asciz      "Entre com o elemento %d => "

mostra1:    .asciz      "Elementos Lidos:"

mostra2:    .asciz      " %d"

mostra3:    .asciz      "Elementos Invertdos:"

pulalin:    .asciz      "\n"

maxtam:     .int        20

tam:        .int        0

num:        .int        0

soma:       .int        0

vetor:      .space      80   # 4 bytes para cada numero a ser armazenado


.section .text
```

```
.globl _start
_start:

      pushl $titulo
      call  printf


letam:
      pushl $pedetam
      call  printf
      pushl $tam
      pushl $formato
      call  scanf
      pushl $pulalin
      call  printf

      movl  tam, %ecx
      cmpl  $0, %ecx
      jle   letam
      cmpl  maxtam, %ecx
      jg    letam

      movl  $vetor, %edi      # endereço inicial do vetor (1o inteiro)
      addl  $16, %esp      # descarta os elementos empilhados
      movl  $0, %ebx    # para enumerar os elementos lidos na leitura

lenum:
      incl  %ebx
      pushl %edi        # backupeia %edi, %ecx, %ebx.
      pushl %ecx       # muitas funções de bibliotecas os modificam
      pushl %ebx

      pushl $pedenum
      call  printf
      pushl $num
      pushl $formato
      call  scanf
      pushl $pulalin
      call  printf
      addl  $16, %esp

      popl  %ebx        # recupera registradores backupeados
      popl  %ecx
      popl  %edi
      movl  num, %eax
      movl  %eax, (%edi)   # põe número na posição corrente do vetor
      addl  $4, %edi     # avanca posicao no vetor
      loop  lenum       # volta a ler o próximo

mostravet:
      pushl $mostra1
      call  printf
      addl  $4, %esp
      movl  tam, %ecx
      movl  $vetor, %edi

mostranum:
```

```
      movl  (%edi), %ebx
      addl  $4, %edi

      pushl %edi
      pushl %ecx
      pushl %ebx

      pushl $mostra2
      call  printf
      addl  $8, %esp

      popl  %ecx
      popl  %edi
      loop  mostranum

invertevetor:
      movl  $vetor, %edi
      movl  $vetor, %esi
      movl  $0, %edx
      movl  tam, %eax
      decl  %eax

      movl  $4, %ebx
      mull  %ebx
      addl  %eax, %esi

      movl  tam, %eax
      movl  $2, %ebx
      divl  %ebx
      movl  %eax, %ecx

gira:
      movl  (%edi), %eax
      movl  (%esi), %ebx
      movl  %eax, (%esi)
      movl  %ebx, (%edi)
      addl  $4, %edi
      subl  $4, %esi
      loop  gira

mostravet2:
      pushl $mostra3
      call  printf
      addl  $4, %esp
      movl  tam, %ecx
      movl  $vetor, %edi

mostranum2:
      movl  (%edi), %ebx
      addl  $4, %edi

      pushl %edi
      pushl %ecx
      pushl %ebx

      pushl $mostra2
      call  printf
      addl  $8, %esp
```

```
      popl  %ecx
      popl  %edi
      loop  mostranum2

      pushl $pulalin
      call  printf
      pushl $pulalin
      call  printf
      addl  $8, %esp

fim:
      pushl $0
      call  exit
```

DESAFIO 1: Altere o programa para buscar dentro do vetor, um determinado número fornecido pelo usuário. Informe o sucesso da busca e a posição em que foi encontrado, se for o caso, ou mensagem de “número inexistente”.

DESAFIO 2: Estenda o programa. Declare um vetor auxiliar e coloque neste vetor auxiliar todos os números menores do que o número fornecido. Mostre esse vetor.