

```

section .data                                ; Inicio da segmento de dados
pedido db "Digite o seu nome> " ; Mensagem que pede o nome
PEDIDO_TAM equ $ - pedido           ; Tamanho da mensagem (constante)
informe db "seu nome eh "
INFOR_TAM equ $ - informe

NOME_MAX equ 100

section .bss                                ; Inicio da segmento de dados nao
inicializados
buf resb NOME_MAX           ; Buffer para receber nome
tam2 resd 1

section .text                                ; Inicio da segmento de codigo
global main
main:
; BLOCO 1
; Escreve a mensagem que pede o nome
mov     eax, 4           ; sys_write
mov     ebx, 1           ; stdout
mov     ecx, pedido      ; Mensagem a escrever
mov     edx, PEDIDO_TAM  ; Tamanho da mensagem
int     0x80             ; Chamar o Linux

; BLOCO 2 ; Le^ o nome a partir do teclado
mov     eax, 3           ; Funcao do sistema (system call); 3 = sys_read
mov     ebx, 0           ; Para sys_read, ebx = handle de leitura; 0 = stdin
mov     ecx, buf         ; Para sys_read, ecx = endereco do buffer para a
leitura
mov     edx, NOME_MAX    ; Para sys_read, edx = tamanho maximo do buffer
int     0x80             ; Chamar o Linux

mov     ebx, eax ; Guarda o tamanho do que foi efectivamente lido

mov     esi, 0

inic:

mov     al, [buf+esi]
cmp     al, "A"
jb      fora
cmp     al, "Z"
ja      fora
add     al, 0x20
mov     [buf+esi], al

fora:

inc     esi
cmp     esi, ebx
jna     inic

; BLOCO 4 ; Escreve o nome lido
mov     edx, INFOR_TAM
mov     ecx, informe
mov     ebx, 1
mov     eax, 4
int     0x80

mov     edx, [tam2]
mov     ecx, buf
mov     ebx, 1
mov     eax, 4
int     0x80

```

```
; BLOCO 5      ; Termina o programa
mov     eax, 1      ; sys_exit
mov     ebx, 0      ; "Sucesso"
int     0x80        ; Chamar o Linux
```