

Explicativo sobre o programa praticas_10c.s

O presente programa exemplifica o uso de funções de biblioteca matemática sobre números em ponto flutuante, bem como a conversão de strings para números inteiros ou em ponto flutuante.

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

```
as praticas_10c.s -o praticas_10c.o
```

e depois link dinamicamente com o seguinte comando.

```
ld praticas_10c.o -lc -lm -dynamic-linker /lib/ld-linux.so.2 -o praticas_10c
```

Observe o uso da opção de montagem "-lm" que indica o uso da biblioteca libmath. Usando essa opção é possível usar funções de seno, cosseno, tangente, potência e raiz quadrada entre outras.

O executável se chamara praticas_10c, sem extensão, e para executá-lo digite:

```
./praticas_10c
```

```
.section .data
```

```
titulo:      .asciz "\nPROGRAMA PARA TESTAR FUNCOES REAIS\n"
titulo1:     .asciz "\nTESTE DA FUNCAO SENO(X)"
titulo2:     .asciz "\nTESTE DA FUNCAO COSSENO(X)"
titulo3:     .asciz "\nTESTE DA FUNCAO TANGENTE(X)"
titulo4:     .asciz "\nTESTE DA FUNCAO DE POTENCIA >> X ELEVADO A N <<"
titulo5:     .asciz "\nTESTE DA FUNCAO RAIZ QUADRADA (X)"
titulo6:     .asciz "\nTESTE DE CONVERSAO DE STRING EM VALOR NUMERICO"
```

```
menu: .asciz "\nEscolha uma Opcao: \n<1> Seno \n<2> Cosseno\n<3> Tangente\n<4> Potencia \n<5> Raiz Quadrada \n<6> Converte String \n<?> Nenhuma\n>>> "
```

```
pedestrf:   .asciz "\nDigite uma String contendo um Valor em Float => "
pedestri:   .asciz "\nDigite uma String contendo um Valor Inteiro => "
```

```
pedextrad:  .asciz "\nDigite um Valor float de X em Radianos\npi = 3.1416, digite 0 < X < 2pi => "
```

```
pederealx:  .asciz "\nDigite um Valor float para X => "
pederealn:  .asciz "Digite um Valor float para N => "
```

```
pergcont:   .asciz "\nDeseja nova execucao ?\n<1> Sim \n<2> Nao\n\n"
```

```
mostraval:  .asciz "\nValor da Funcao = %.2lf\n"
```

```
mostraint:  .asciz "\nValor Inteiro = %d\n"
mostrafloat: .asciz "\nValor em Float = %.4lf\n"
```

```
strfloat:   .asciz "12345678901234567890"
strint:      .asciz "12345678901234567890"
```

```
val1: .double    0.0
val2: .double    0.0
```

```
formatof: .asciz "%lf"
formatod: .asciz "%d"
formatos: .asciz "%s"
```

```
opcao: .int      0
```

```
# valores conhecidos para testar
```

| # | radianos | graus | sen | cos | tan |
|--------|----------|-------|------|------|------|
| # pi: | 3.1416 | 180 | 0 | -1 | 0 |
| # 2*pi | 6.2832 | 360 | 0 | 1 | 0 |
| # pi/2 | 1.5708 | 90 | 1 | 0 | --- |
| # pi/3 | 1.0472 | 60 | R3/2 | 1/2 | R3 |
| # pi/4 | 0.7854 | 45 | R2/2 | R2/2 | 1 |
| # pi/6 | 0.5236 | 30 | 1/2 | R3/2 | R3/3 |

```
.section .text
.globl _start
_start:
```

```
    pushl $titulo
    call  printf
    pushl $menu
    call  printf
    pushl $opcao
    pushl $formatod
    call  scanf
    addl  $16, %esp
```

```
    movl  opcao, %eax
    cmpl  $1, %eax
    je    calcsen
```

```
    cmpl  $2, %eax
    je    calccos
```

```
    cmpl  $3, %eax
    je    calctan
```

```
    cmpl  $4, %eax
    je    calcpot
```

```
    cmpl  $5, %eax
    je    calcraizq
```

```
    cmpl  $6, %eax
    je    convstr
```

```
    jmp   fim
```

Observe que, no geral, as funções de manipulação de números em ponto flutuante (real) funcionam da seguinte forma: O programa coloca na pilha o valor do ângulo em radianos e chama a função com call. Então, a função retorna o valor no registrador %st0, ou seja, no topo dos registradores de ponto flutuante.

calcsen:

```
pushl $titulo1
call  printf
pushl $pedexrad
call  printf
pushl $val1
pushl $formatolf
call  scanf
addl  $16, %esp

finit
fldl  val1
subl  $8, %esp
fstl  (%esp)
call  sin

subl  $8, %esp
fstl  (%esp)
pushl $mostraval
call  printf

addl  $20, %esp
jmp   fim
```

calccos:

```
pushl $titulo2
call  printf
pushl $pedexrad
call  printf
pushl $val1
pushl $formatolf
call  scanf
addl  $16, %esp

finit
fldl  val1
subl  $8, %esp
fstl  (%esp)
call  cos

subl  $8, %esp
fstl  (%esp)
pushl $mostraval
call  printf

addl  $20, %esp
jmp   fim
```

calctan:

```
    pushl $titulo3
    call  printf
    pushl $pedextrad
    call  printf
    pushl $val1
    pushl $formatolf
    call  scanf
    addl  $16, %esp

    finit
    fldl  val1
    subl  $8, %esp
    fstl  (%esp)

    call  tan

    subl  $8, %esp
    fstl  (%esp)
    pushl $mostraval
    call  printf

    addl  $20, %esp
    jmp   fim
```

calcpot:

```
    pushl $titulo4
    call  printf

    pushl $pederealex
    call  printf
    pushl $val1
    pushl $formatolf
    call  scanf

    pushl $pederealn
    call  printf
    pushl $val2
    pushl $formatolf
    call  scanf

    addl  $28, %esp

    finit
    fldl  val2
    subl  $8, %esp
    fstl  (%esp)

    fldl  val1
    subl  $8, %esp
    fstl  (%esp)

    call  pow

    subl  $8, %esp
```

```

fstl    (%esp)
pushl   $mostraval
call    printf

addl    $28, %esp

jmp     fim

```

calcraizq:

```

pushl   $titulo5
call    printf
pushl   $pederealx
call    printf
pushl   $val1
pushl   $formatolf
call    scanf
addl    $16, %esp

finit
fldl    val1
subl    $8, %esp
fstl    (%esp)

call    sqrt

subl    $8, %esp
fstl    (%esp)
pushl   $mostraval
call    printf

addl    $20, %esp

jmp     fim

```

convstr:

```

pushl   $titulo6
call    printf

pushl   $pedestri
call    printf
pushl   $strint
pushl   $formatos
call    scanf
addl    $12, %esp

pushl   $strint
call    atoi
pushl   %eax
pushl   $mostraint
call    printf
addl    $12, %esp

pushl   $pedestrf
call    printf
pushl   $strfloat

```

```

pushl $formatos
call  scanf
addl  $12, %esp

finit
pushl $strfloat
call  atof
subl  $8, %esp
fstl  (%esp)
pushl $mostrafloat
call  printf
addl  $16, %esp

jmp   fim

```

Finaliza o programa

fim:

```

pushl $pergcont
call  printf

pushl $opcao
pushl $formatod
call  scanf
addl  $12, %esp

movl  opcao, %eax
cmpl  $1, %eax

je    _start

pushl $0
call  exit

```

Desafio: Altere o menu e crie uma função para a cotangente e outra para logaritmo. Implemente. Se não souber como fazer, pesquise. Entregue no prazo e receba um bonus de final de semestre.