

Linguagem Assembly

Ponto Flutuante



PONTOS FLUTUANTES

- Números representados com casas decimais
 - float
 - 32 bits
 - um único registrador do coprocessador I
 - double
 - 64 bits
 - dois registradores do coprocessador I

PONTOS FLUTUANTES

Registers	Coproc 1	Coproc 0
Name	Float	Double
\$f0	0.0	0.0
\$f1	0.0	
\$f2	0.0	0.0
\$f3	0.0	
\$f4	0.0	0.0
\$f5	0.0	
\$f6	0.0	0.0
\$f7	0.0	
\$f8	0.0	0.0
\$f9	0.0	
\$f10	0.0	0.0
\$f11	0.0	
\$f12	0.0	0.0
\$f13	0.0	
\$f14	0.0	0.0
\$f15	0.0	
\$f16	0.0	0.0
\$f17	0.0	
\$f18	0.0	0.0
\$f19	0.0	
\$f20	0.0	0.0
\$f21	0.0	
\$f22	0.0	0.0
\$f23	0.0	
\$f24	0.0	0.0
\$f25	0.0	
\$f26	0.0	0.0
\$f27	0.0	
\$f28	0.0	0.0
\$f29	0.0	
\$f30	0.0	0.0
\$f31	0.0	

IMPRIMINDO FLOAT

```
.data
    PI: .float 3.141592
.text
    li $v0, 2 #sistema, prepare-se pra imprimir um float
    lwcl $f12, PI #no caso dos float, os registradores estão
    #no co-processador 1 (cp1)
    #SEMPRE devemos colocar o valor do float em $f12, ou
    #o valor correto não é impresso
    syscall
```

COMANDOS LI \$VO

Comando	Significado
li \$v0, 1	imprimir inteiro
li \$v0, 2	imprimir float
li \$v0, 3	imprimir double
li \$v0, 4	imprimir String ou char
li \$v0, 5	ler inteiro
li \$v0, 6	ler float
li \$v0, 7	ler double
li \$v0, 8	ler String ou char
li \$v0, 10	encerrar programa principal

LENDÔ FLOAT

```
li $v0, 6
```

```
syscall
```

#valor lido como float vai para \$f0

EXERCÍCIO

Implementar um programa que lê um float e o imprime.

