

PROVA 1

Victor Hugo Chimenez Queiroz NUSP: 11288405

Tarefa 1:

```
.text
.global main
dados: .word 0x1, 0x2, 3, 4, 5, 6, 7, 0x8, 0x9, 0

main:
    LDR R0, =dados           @ recebe o endereço do array de dados
    LDR R1, [R0]             @ dividendo - Carrega o primeiro elemento de dados em R1
    LDR R2, = 1000           @ divisor
    MOV R3, #0               @ quociente
    MOV R4, #1               @ variável que realiza a soma do quociente
pronto:
    alinhar divisor:         @ alinhamos o divisor para a "mesma" casa do divisor como fazemos co
```

Apenas as duas linhas com highlight foram necessárias para adicionar a mudança pedida. Primeiro carregamos o ponteiro para o array de dados em R0 e depois carregamos o primeiro elemento do array em R1.

Prints da execução:

```
remote Thread 1.60316 In:
(gdb) b main
Ponto de parada 1 at 0x10414: file tarefa1.s, line 10.
(gdb) b pronto
Ponto de parada 2 at 0x10428: file tarefa1.s, line 17.
(gdb) █
```

Pontos de parada em main e pronto, assim como solicitado.

Primeiro breakpoint:

```

2 @ 0 objetivo dele é realizar uma divisão baseada no algoritmo de shift e s
3
4 .text
5 .global main
6 dados: .word 0x1, 0x2, 3, 4, 5, 6, 7, 0x8, 0x9, 0
7
8
9 main:
B+> 10 LDR R0, =dados @ recebe o endereço do array de dados
11 LDR R1, [R0] @ dividendo - Carrega o primeiro elemento
12 LDR R2, = 1000 @ divisor
13 MOV R3, #0 @ quociente
14 MOV R4, #1 @ variável que realiza a soma do d
15 pronto:
b+ 16 alinhar_divisor: @ alinhamos o divisor para a "mesma" casa
17 CMP R1, R2, LSL #1 @ comparamos o dividendo com o divisor
18 BLT divisao @ se dividendo < divisor, passamos
19 LSL R2, R2, #1 @ anteriormente foi apenas uma ver

```

```

remote Thread 1.60681 In: main
(gdb) b main
Ponto de parada 1 at 0x10414: file tarefa1.s, line 10.
(gdb) b pronto
Ponto de parada 2 at 0x10428: file tarefa1.s, line 17.
(gdb) c
Continuing.

Breakpoint 1, main () at tarefa1.s:10
(gdb) x/10d dados
0x103ec <dados>:      1      2      3      4
0x103fc <dados+16>:    5      6      7      8
0x1040c <dados+32>:    9      0
(gdb)

```

Mostra os 10 elementos do array de dados (em decimal) em memória carregado na variável “dados”
 Veja que neste instance R1 possui “lixo”:

```

$1 = 1082130232
(gdb) p/d $r1
$2 = 1082130244
(gdb)

```

Segundo Breakpoint:

```

Breakpoint 2, pronto () at tarefa1.s:17
(gdb) p/d $r1
$3 = 1
(gdb) p/x $r1
$4 = 0x1
(gdb)

```

R1 recebeu o primeiro valor do array corretamente.

Tarefa 2

Não tive tempo de concluí-la