

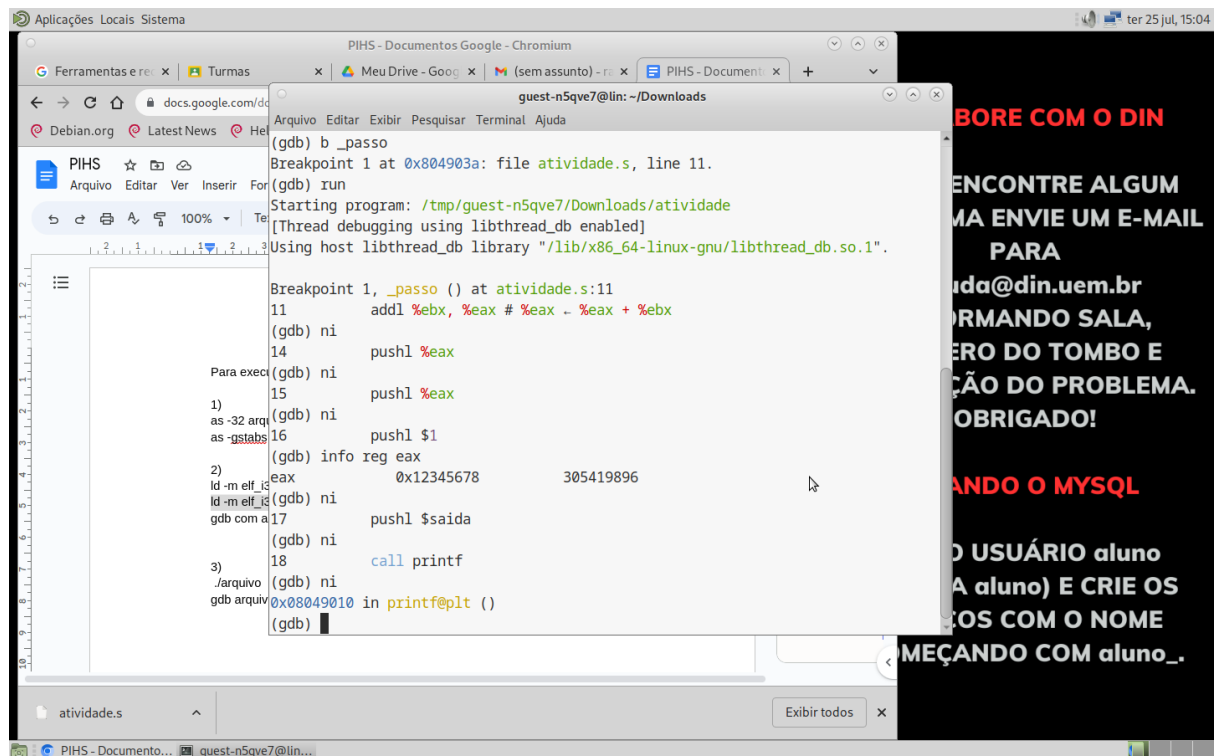
Prática 05 - Partes 1 e 2

RA117306 Felipe Gabriel Comin Scheffel

RA117741 Douglas Kenji Sakakibara

1)

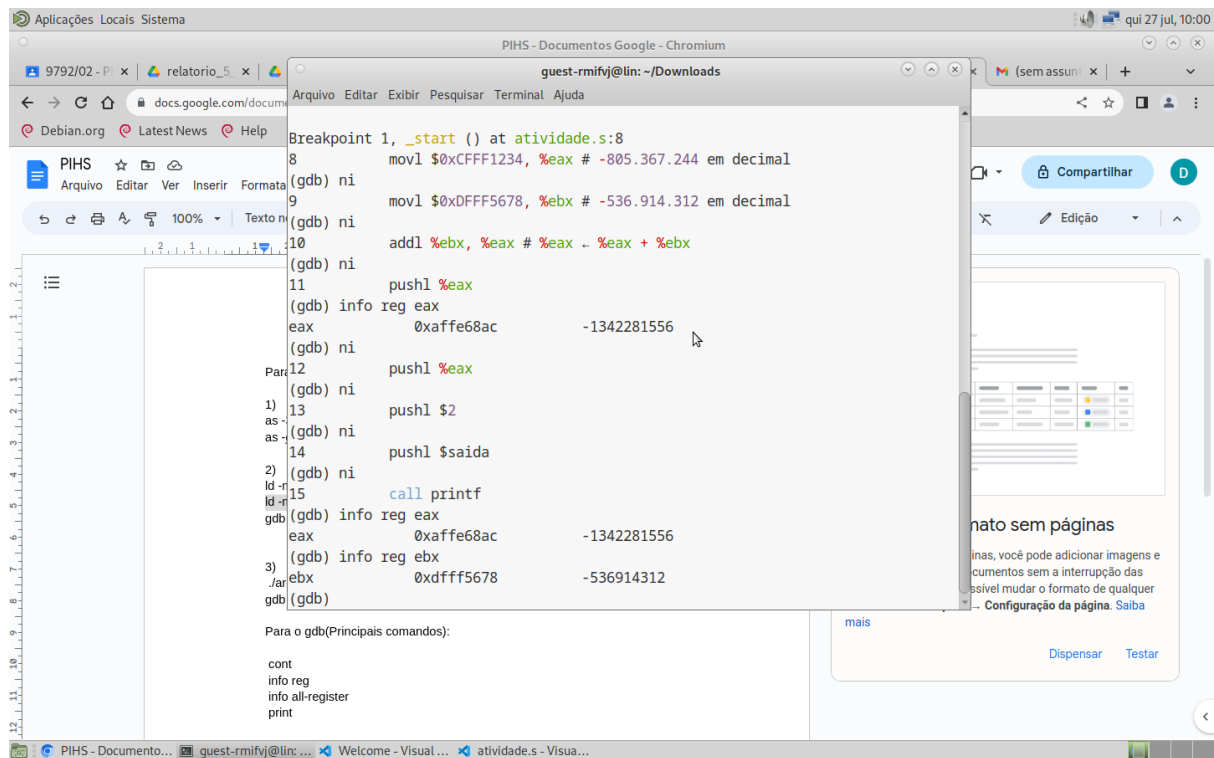
Teste 1:



```
guest-n5qe7@lin: ~/Downloads
(gdb) b _passo
Breakpoint 1 at 0x804903a: file atividade.s, line 11.
(gdb) run
Starting program: /tmp/guest-n5qe7/Downloads/atividade
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

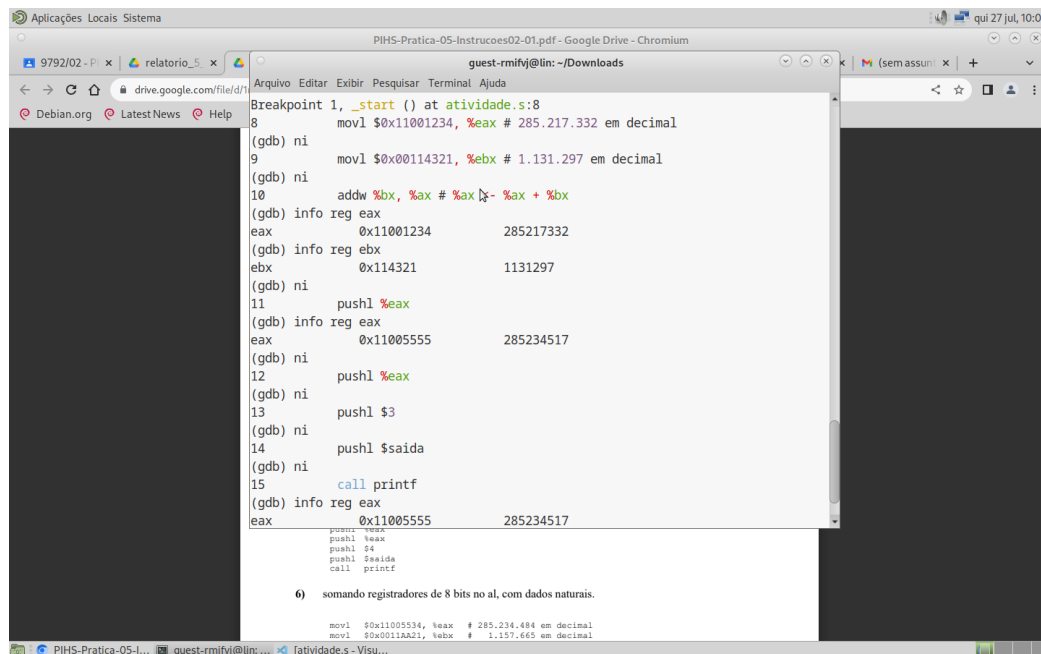
Breakpoint 1, _passo () at atividade.s:11
11      addl %ebx, %eax # %eax += %ebx
(gdb) ni
14      pushl %eax
(gdb) ni
15      pushl %eax
(gdb) ni
16      pushl $1
(gdb) info reg eax
eax             0x12345678      305419896
(gdb) ni
17      pushl $saida
(gdb) ni
18      call printf
(gdb) ni
0x08049010 in printf@plt ()
(gdb)
```

Teste 2:



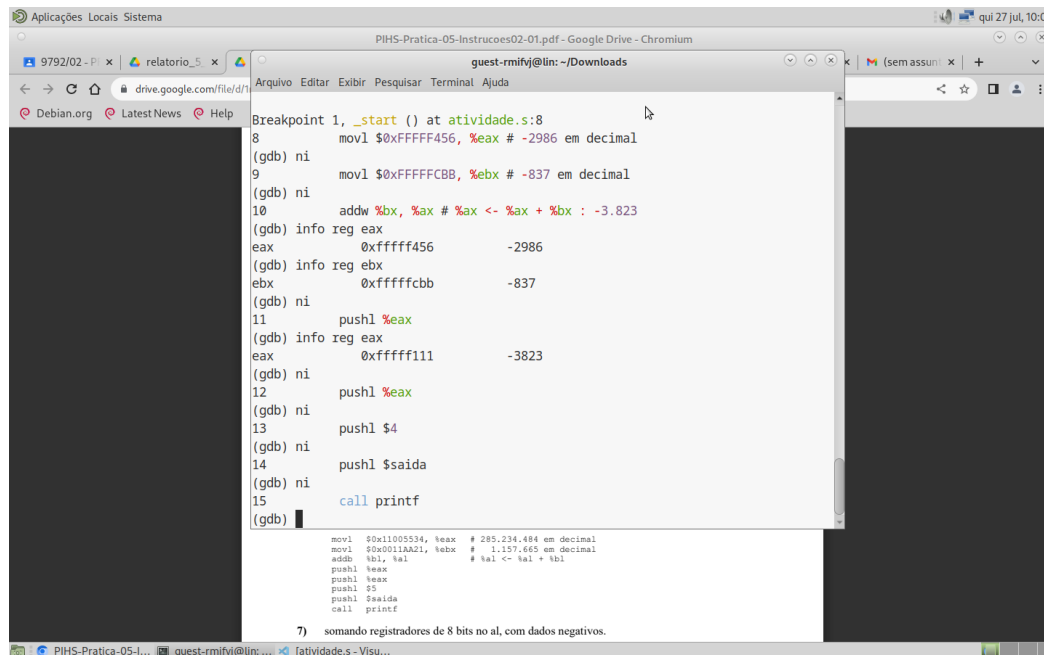
2)

Teste 3:



Na instrução `addw %bx, %ax`, é realizado uma soma dos últimos 16 bits do EBX com os últimos 16 bits do EAX, resultando num valor final de EAX: 11005555.

Teste 4:



```
Breakpoint 1, _start () at atividade.s:8
8      movl $0xFFFFF456, %eax # -2986 em decimal
(gdb) ni
9      movl $0xFFFFCBB, %ebx # -837 em decimal
(gdb) ni
10     addw %bx, %ax # %ax <- %ax + %bx : -3.823
(gdb) info reg eax
eax    0xfffff456      -2986
(gdb) info reg ebx
ebx    0xffffcbb      -837
(gdb) ni
11     pushl %eax
(gdb) info reg eax
eax    0xfffff111      -3823
(gdb) ni
12     pushl %eax
(gdb) ni
13     pushl $4
(gdb) ni
14     pushl $saida
(gdb) ni
15     call printf
(gdb)

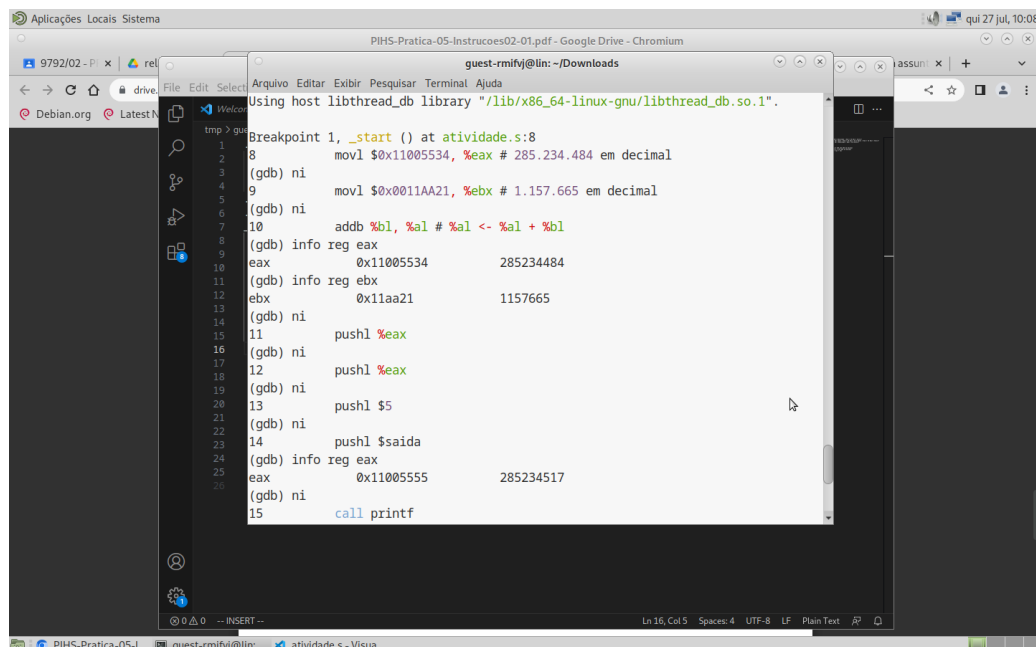
movl $0x11005534, %eax # 285.234.484 em decimal
movl $0x0011AA21, %ebx # 1.157.665 em decimal
addb %bl, %al
pushl %eax
pushl %eax
pushl $5
pushl $saida
call printf

7) somando registradores de 8 bits no al com dados negativos.
```

Na instrução `addw %bx, %ax`, é realizado uma soma dos últimos 16 bits do EBX com os últimos 16 bits do EAX, resultando num valor final de EAX: `fffff111`

3)

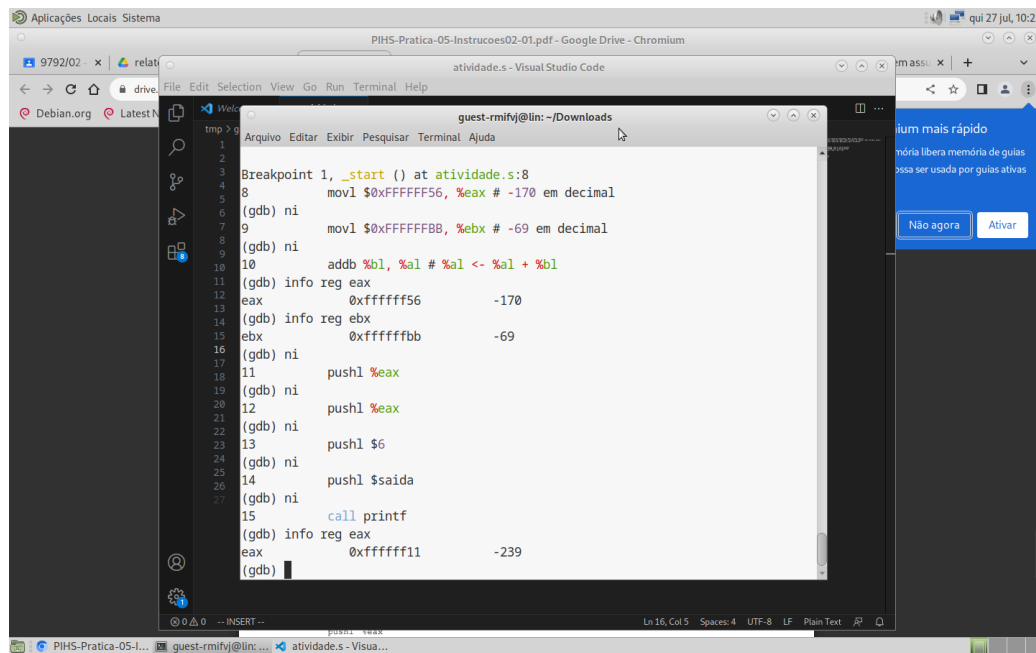
Teste 5:



```
Breakpoint 1, _start () at atividade.s:8
1      movl $0x11005534, %eax # 285.234.484 em decimal
2      (gdb) ni
3
4      movl $0x0011AA21, %ebx # 1.157.665 em decimal
5      (gdb) ni
6
7      addb %bl, %al # %al <- %al + %bl
8      (gdb) info reg eax
9      eax    0x11005534      285234484
10     (gdb) info reg ebx
11     ebx    0x11aa21        1157665
12     (gdb) ni
13     pushl %eax
14     (gdb) ni
15     pushl %eax
16     (gdb) ni
17     pushl $5
18     (gdb) ni
19     pushl $saida
20     (gdb) info reg eax
21     eax    0x11005555      285234517
22     (gdb) ni
23     call printf
24
25
```

Na instrução `addb %bl, %al`, é realizado uma soma dos últimos 8 bits do EBX com os últimos 8 bits do EAX, resultando num valor final de EAX: `11005555`.

Teste 6:



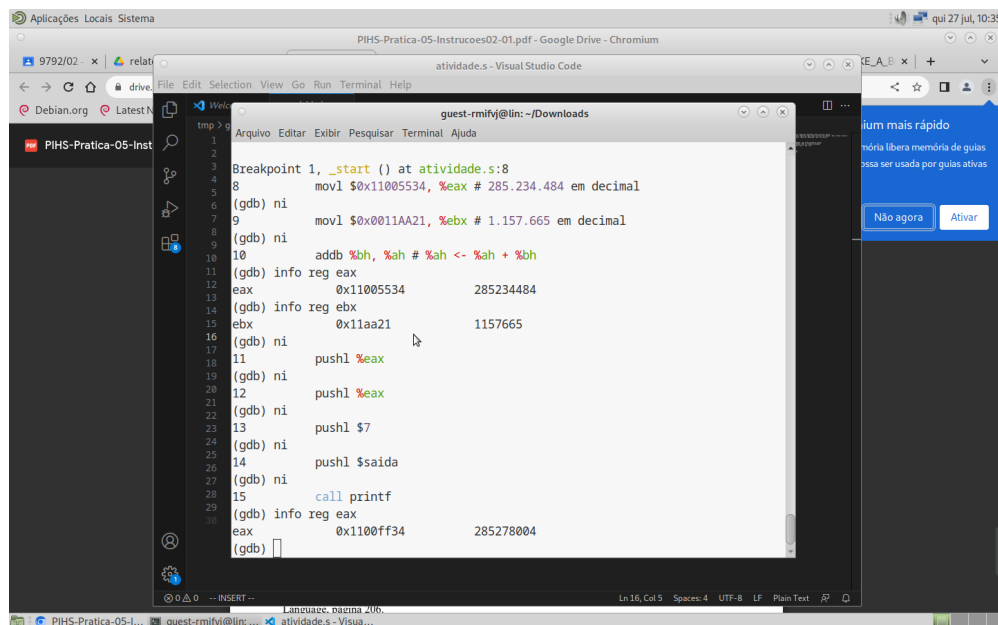
The screenshot shows a debugger window with the following assembly code and register values:

```
Breakpoint 1, _start () at atividade.s:8
8      movl $0xffffffff56, %eax # -170 em decimal
(gdb) ni
9      movl $0xffffffffb8, %ebx # -69 em decimal
(gdb) ni
10     addb %bl, %al # %al <- %al + %bl
(gdb) info reg eax
eax    0xffffffff56      -170
(gdb) info reg ebx
ebx    0xffffffffb8      -69
(gdb) ni
11     pushl %eax
(gdb) ni
12     pushl %eax
(gdb) ni
13     pushl $6
(gdb) ni
14     pushl $saida
(gdb) ni
15     call printf
(gdb) info reg eax
eax    0xffffffff11      -239
(gdb)
```

Na instrução `addb %bl, %al`, é realizado uma soma dos últimos 8 bits do EBX com os últimos 8 bits do EAX, resultando num valor final de EAX: `ffffff11`.

4)

Teste 7:

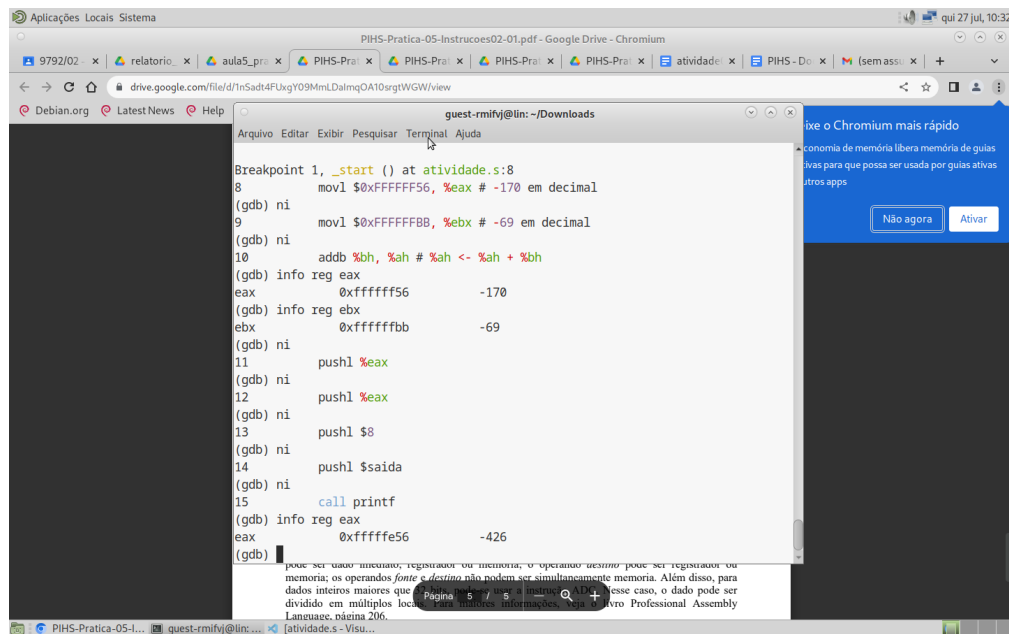


The screenshot shows a debugger window with the following assembly code and register values:

```
Breakpoint 1, _start () at atividade.s:8
8      movl $0x11005534, %eax # 285.234.484 em decimal
(gdb) ni
9      movl $0x0011AA21, %ebx # 1.157.665 em decimal
(gdb) ni
10     addb %bh, %ah # %ah <- %ah + %bh
(gdb) info reg eax
eax    0x11005534      285234484
(gdb) info reg ebx
ebx    0x11aa21        1157665
(gdb) ni
11     pushl %eax
(gdb) ni
12     pushl %eax
(gdb) ni
13     pushl $7
(gdb) ni
14     pushl $saida
(gdb) ni
15     call printf
(gdb) info reg eax
eax    0x1100ff34      285278004
(gdb)
```

Na instrução `addb %bh, %ah`, é realizado uma soma dos próximos 8 bits do EBX com os próximos 8 bits do EAX, resultando num valor final de EAX: `1100ff34`.

Teste 8:

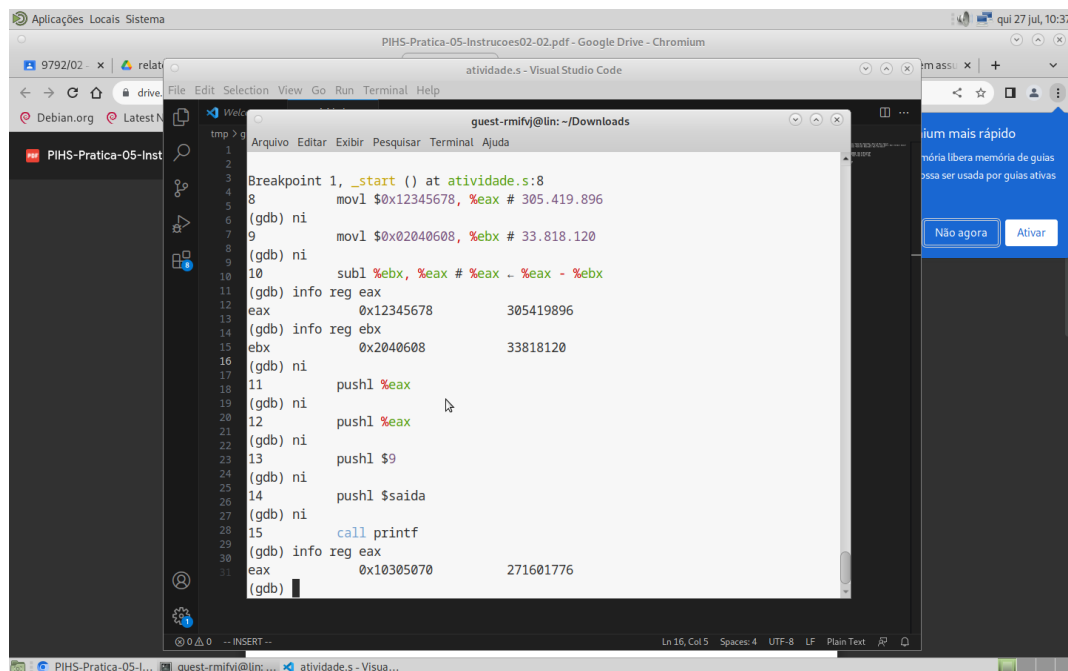


```
Breakpoint 1, _start () at atividade.s:8
8      movl $0xffffffff56, %eax # -170 em decimal
(gdb) ni
9      movl $0xffffffffb8, %ebx # -69 em decimal
(gdb) ni
10     addb %bh, %ah # %ah <- %ah + %bh
(gdb) info reg eax
eax    0xffffffff56    -170
(gdb) info reg ebx
ebx    0xffffffffb8    -69
(gdb) ni
11     pushl %eax
(gdb) ni
12     pushl %eax
(gdb) ni
13     pushl $8
(gdb) ni
14     pushl $saida
(gdb) ni
15     call printf
(gdb) info reg eax
eax    0xfffffe56    -426
(gdb)
```

Na instrução `addb %bh, %ah`, é realizado uma soma dos próximos 8 bits do EBX com os próximos 8 bits do EAX, resultando num valor final de EAX: `fffffe56`.

5)

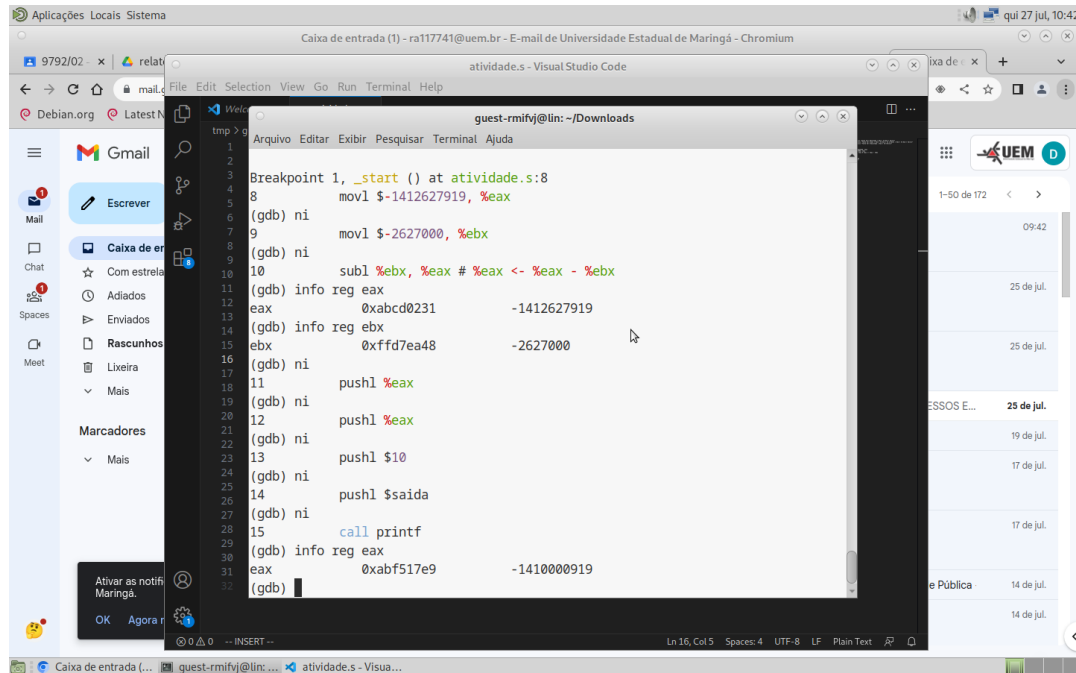
Teste 9:



```
Breakpoint 1, _start () at atividade.s:8
8      movl $0x12345678, %eax # 305.419.896
(gdb) ni
9      movl $0x02040608, %ebx # 33.818.120
(gdb) ni
10     subl %ebx, %eax # %eax = %eax - %ebx
(gdb) info reg eax
eax    0x12345678    305419896
(gdb) info reg ebx
ebx    0x02040608    33818120
(gdb) ni
11     pushl %eax
(gdb) ni
12     pushl %eax
(gdb) ni
13     pushl $9
(gdb) ni
14     pushl $saida
(gdb) ni
15     call printf
(gdb) info reg eax
eax    0x10305070    271601776
(gdb)
```

Na instrução `subl %ebx, %eax`, é realizado uma subtração do valor do registrador EBX com o valor do registrador EAX, resultando num valor final de EAX: `10305070`.

Teste 10:

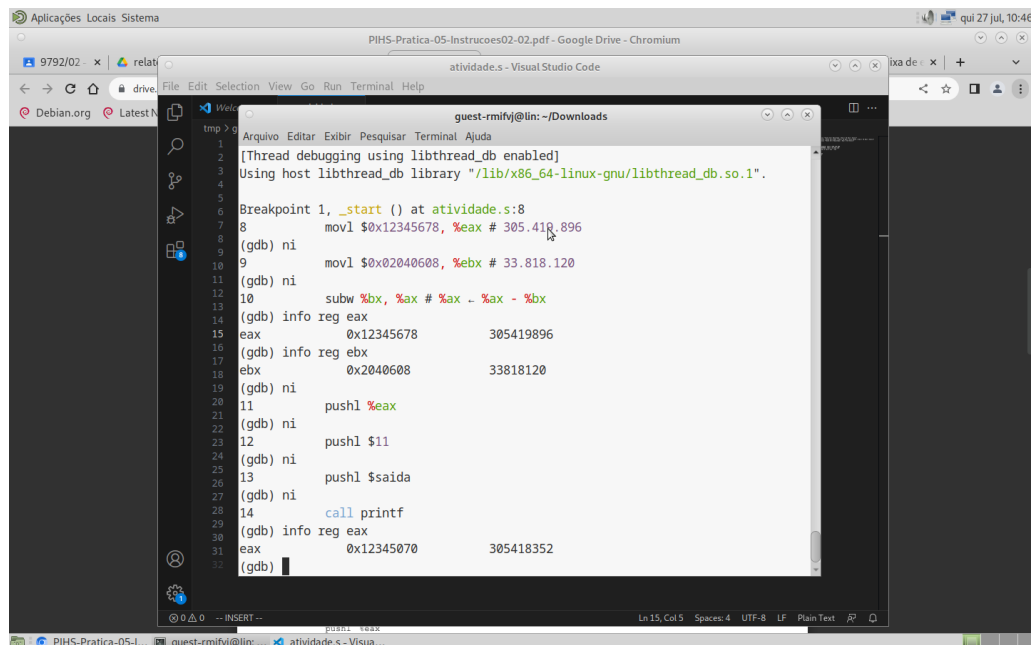


```
Breakpoint 1, _start () at atividade.s:8
8      movl $-1412627919, %eax
9      (gdb) ni
10     movl $-2627000, %ebx
11     (gdb) ni
12     subl %ebx, %eax # %eax <- %eax - %ebx
13     (gdb) info reg eax
14     eax      0xabcd0231      -1412627919
15     (gdb) info reg ebx
16     ebx      0xffd7ea48      -2627000
17     (gdb) ni
18     pushl %eax
19     (gdb) ni
20     pushl %eax
21     (gdb) ni
22     pushl $10
23     (gdb) ni
24     pushl $saida
25     (gdb) ni
26     call printf
27     (gdb) info reg eax
28     eax      0xabf517e9      -1410000919
29     (gdb)
```

Na instrução `subl %ebx, %eax`, é realizado uma subtração do valor do registrador EBX com o valor do registrador EAX, resultando num valor final de EAX: `abf517e9`.

6)

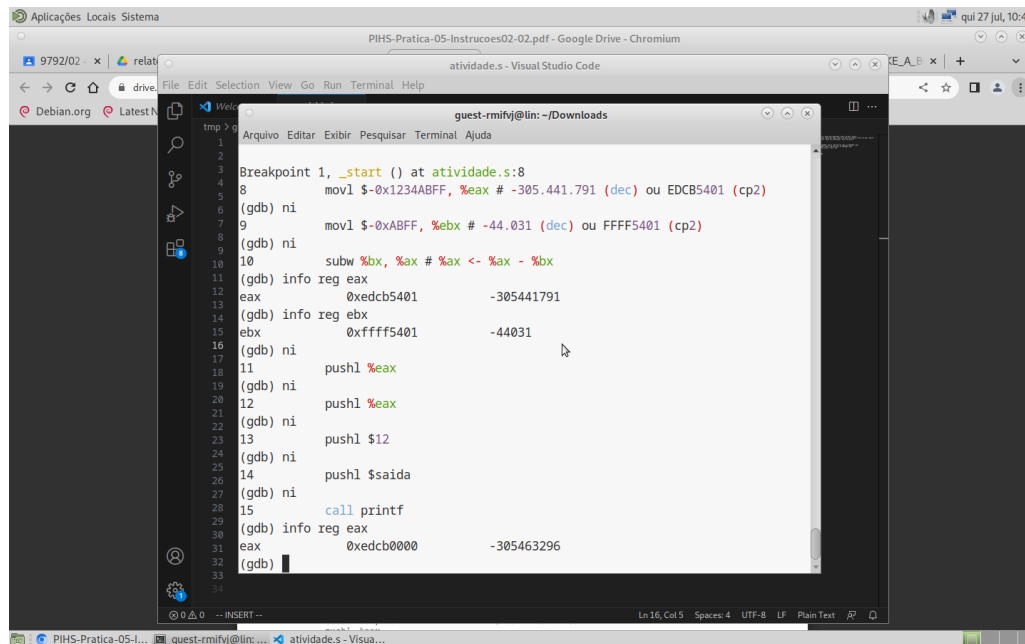
Teste 11:



```
Breakpoint 1, _start () at atividade.s:8
8      movl $0x12345678, %eax # 305.419.896
9      (gdb) ni
10     movl $0x2040608, %ebx # 33.818.120
11     (gdb) ni
12     subw %bx, %ax # %ax <- %ax - %bx
13     (gdb) info reg eax
14     eax      0x12345678      305419896
15     (gdb) info reg ebx
16     ebx      0x2040608      33818120
17     (gdb) ni
18     pushl %eax
19     (gdb) ni
20     pushl $11
21     (gdb) ni
22     pushl $saida
23     (gdb) ni
24     call printf
25     (gdb) info reg eax
26     eax      0x12345070      305418352
27     (gdb)
```

Na instrução `subw %bx, %ax`, é realizado uma subtração dos últimos 16 bits de EBX com os últimos 16 bits do EAX, resultando num valor final de EAX: 13245070.

Teste 12:

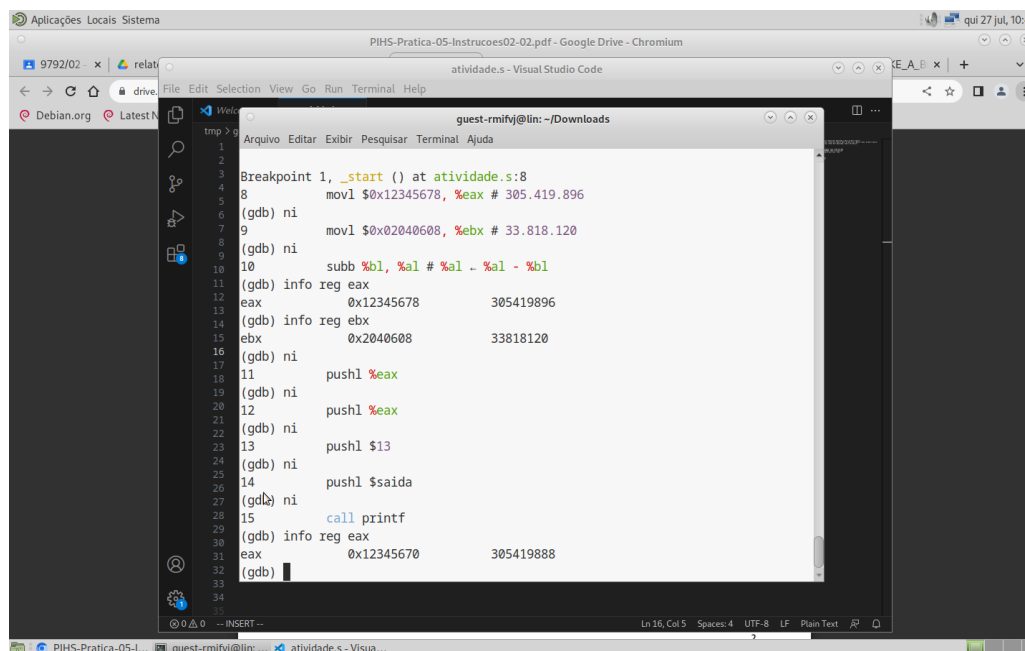


```
Breakpoint 1, _start () at atividade.s:8
8      movl $-0x1234ABFF, %eax # -305.441.791 (dec) ou EDCB5401 (cp2)
(gdb) ni
9      movl $-0xABFF, %ebx # -44.031 (dec) ou FFFF5401 (cp2)
(gdb) ni
10     subw %bx, %ax # %ax <- %ax - %bx
(gdb) info reg eax
eax    0xedcb5401    -305441791
(gdb) info reg ebx
ebx    0xffff5401    -44031
(gdb) ni
11     pushl %eax
(gdb) ni
12     pushl %eax
(gdb) ni
13     pushl $12
(gdb) ni
14     pushl $saida
(gdb) ni
15     call printf
(gdb) info reg eax
eax    0xedcb0000    -305463296
(gdb)
```

Na instrução `subw %bx, %ax`, é realizado uma subtração dos últimos 16 bits de EBX com os últimos 16 bits do EAX, resultando num valor final de EAX: edcb0000.

7)

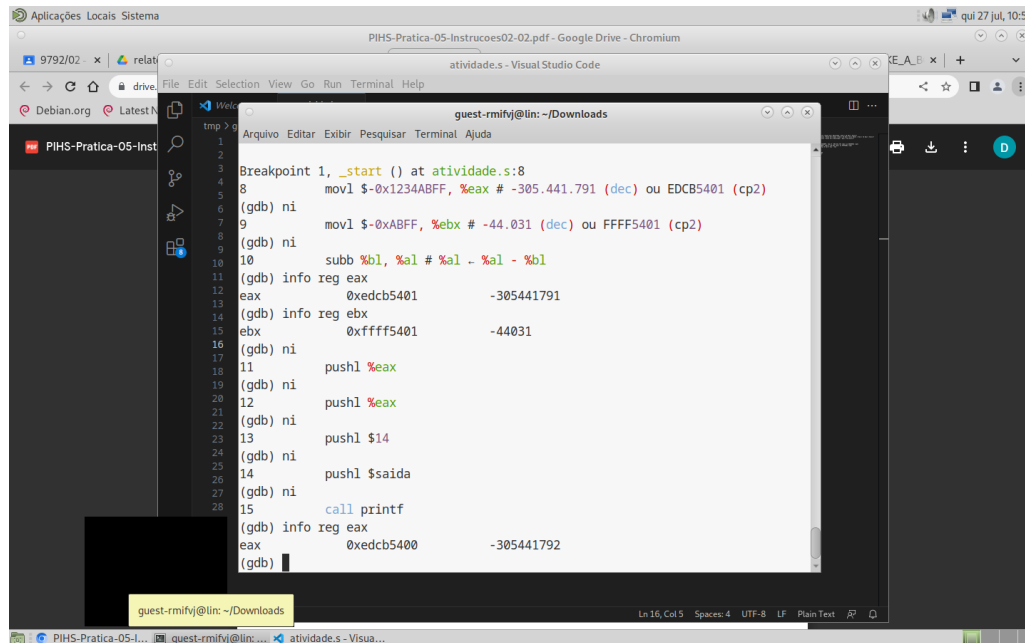
Teste 13:



```
Breakpoint 1, _start () at atividade.s:8
8      movl $0x12345678, %eax # 305.419.896
(gdb) ni
9      movl $0x02040608, %ebx # 33.818.120
(gdb) ni
10     subb %b1, %a1 # %a1 -= %b1
(gdb) info reg eax
eax    0x12345678    305419896
(gdb) info reg ebx
ebx    0x02040608    33818120
(gdb) ni
11     pushl %eax
(gdb) ni
12     pushl %eax
(gdb) ni
13     pushl $13
(gdb) ni
14     pushl $saida
(gdb) ni
15     call printf
(gdb) info reg eax
eax    0x12345670    305419888
(gdb)
```

Na instrução `subb %bl, %al`, é realizado uma subtração dos últimos 8 bits de EBX com os últimos 8 bits do EAX, resultando num valor final de EAX: 305419888.

Teste 14:

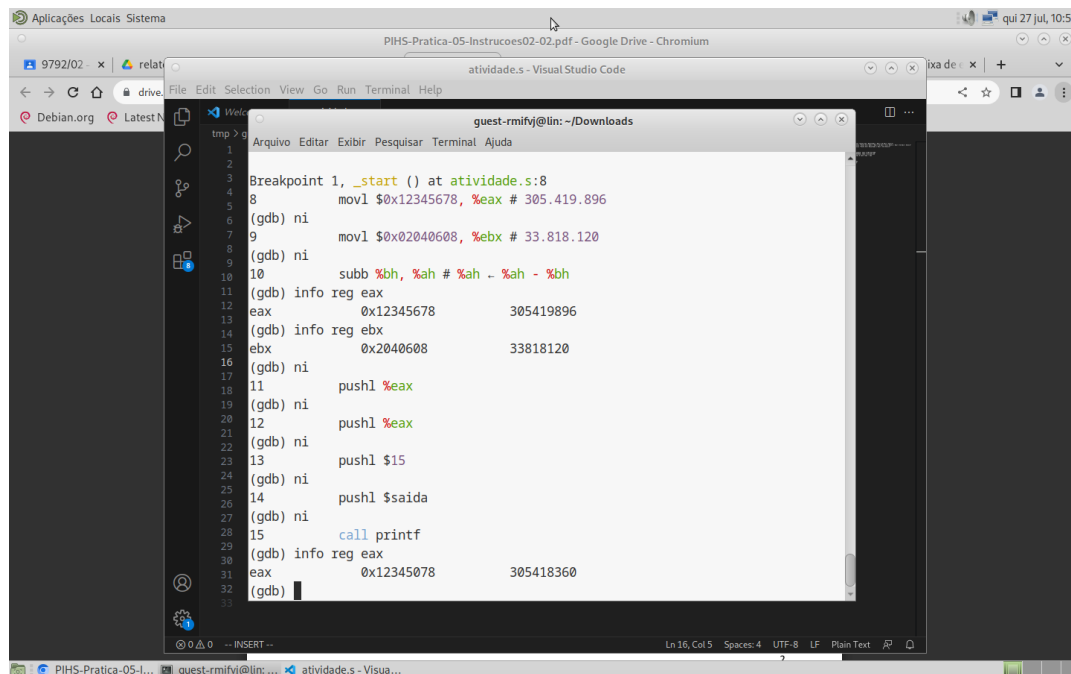


```
Breakpoint 1, _start () at atividade.s:8
8      movl $-0x1234ABFF, %eax # -305.441.791 (dec) ou EDCB5401 (cp2)
(gdb) ni
9      movl $-0xABFF, %ebx # -44.031 (dec) ou FFFF5401 (cp2)
(gdb) ni
10     subb %bl, %al # %al - %bl
(gdb) info reg eax
eax      0xedcb5401      -305441791
(gdb) info reg ebx
ebx      0xffff5401      -44031
(gdb) ni
11     pushl %eax
(gdb) ni
12     pushl %eax
(gdb) ni
13     pushl $14
(gdb) ni
14     pushl $saida
(gdb) ni
15     call printf
(gdb) info reg eax
eax      0xedcb5400      -305441792
(gdb)
```

Na instrução `subb %bl, %al`, é realizado uma subtração dos últimos 8 bits de EBX com os últimos 8 bits do EAX, resultando num valor final de EAX: edcb5400.

8)

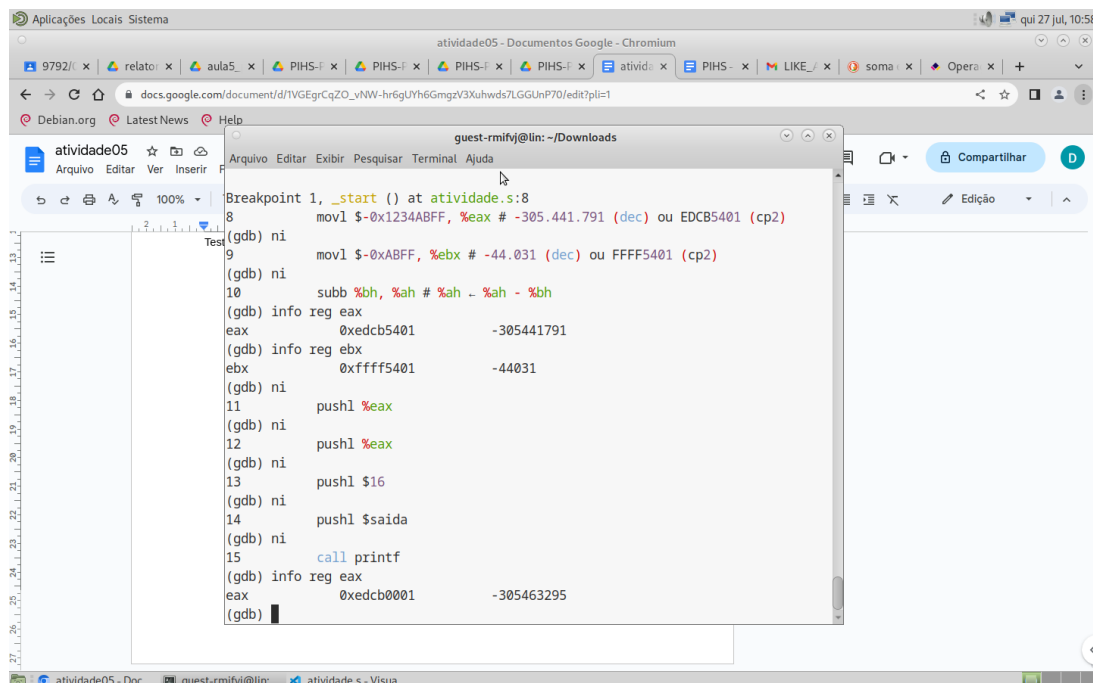
Teste 15:



```
Breakpoint 1, _start () at atividade.s:8
8      movl $0x12345678, %eax # 305.419.896
(gdb) ni
9      movl $0x02040608, %ebx # 33.818.120
(gdb) ni
10     subb %bh, %ah # %ah - %bh
(gdb) info reg eax
eax    0x12345678      305419896
(gdb) info reg ebx
ebx    0x02040608      33818120
(gdb) ni
11     pushl %eax
(gdb) ni
12     pushl %eax
(gdb) ni
13     pushl $15
(gdb) ni
14     pushl %saida
(gdb) ni
15     call printf
(gdb) info reg eax
eax    0x12345078      305418360
(gdb)
```

Na instrução `subb %bh, %ah`, é realizado uma subtração dos próximos 8 bits de `EBX` com os próximos 8 bits do `EAX`, resultando num valor final de `EAX`: 12345078.

Teste 16:



```
Breakpoint 1, _start () at atividade.s:8
8      movl $-0x1234ABFF, %eax # -305.441.791 (dec) ou EDCB5401 (cp2)
(gdb) ni
9      movl $-0xABFF, %ebx # -44.031 (dec) ou FFFF5401 (cp2)
(gdb) ni
10     subb %bh, %ah # %ah - %bh
(gdb) info reg eax
eax    0xedcb5401      -305441791
(gdb) info reg ebx
ebx    0xffff5401      -44031
(gdb) ni
11     pushl %eax
(gdb) ni
12     pushl %eax
(gdb) ni
13     pushl $16
(gdb) ni
14     pushl %saida
(gdb) ni
15     call printf
(gdb) info reg eax
eax    0xedcb0001      -305463295
(gdb)
```

Na instrução `subb %bh, %ah`, é realizado uma subtração dos próximos 8 bits de `EBX` com os próximos 8 bits do `EAX`, resultando num valor final de `EAX`: edcb0001.