

Debugando com GDB TUI

```
gdb -q ./ass -tui
```

```
B+ 0x8049000 <_main>      mov     $0x4,%eax
0x8049005 <_main+5>     mov     $0x1,%ebx
0x804900a <_main+10>    mov     $0x804a000,%ecx
> 0x804900f <_main+15> mov     $0xf,%edx
0x8049014 <_main+20>    int     $0x80
0x8049016 <_main+22>    mov     $0x1,%eax
0x804901b <_main+27>    mov     $0x0,%ebx
0x8049020 <_main+32>    int     $0x80
0x8049022          add     %al,(%eax)
0x8049024          add     %al,(%eax)
0x8049026          add     %al,(%eax)
0x8049028          add     %al,(%eax)
0x804902a          add     %al,(%eax)
0x804902c          add     %al,(%eax)
0x804902e          add     %al,(%eax)
0x8049030          add     %al,(%eax)
0x8049032          add     %al,(%eax)
0x8049034          add     %al,(%eax)
0x8049036          add     %al,(%eax)
0x8049038          add     %al,(%eax)
0x804903a          add     %al,(%eax)
0x804903c          add     %al,(%eax)
0x804903e          add     %al,(%eax)
0x8049040          add     %al,(%eax)
```

```
native process 175108 In: _main
--Type <RET> for more, q to quit, c to continue without paging--es      0x2b
fs          0x0          0x2b      0          43
gs          0x0          0x2b      0          43
(gdb) layout regs          0x0          0          0
(gdb) layout asm          0x0          0          0
(gdb) si          0x0          0          0
0x08049005 in _main ()
(gdb) si          in _main ()
0x0804900a in _main ()
(gdb) si          in _main ()
0x0804900f in _main ()
(gdb) x/s 0x804a000          in _main ()
0x804a000: (gdb) "Desec Security\n"
(gdb) █
```

→ Podemos executar tudo igual da aula passada que veremos o resultado na tela apresentada