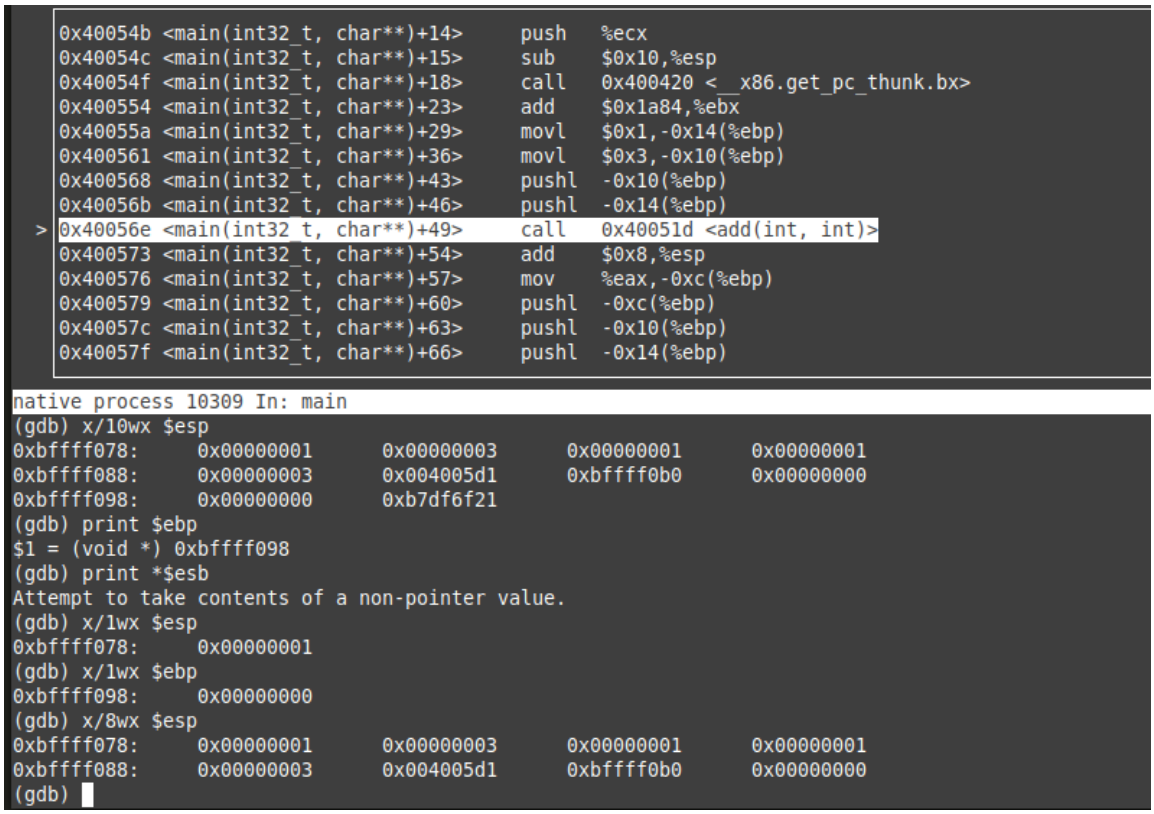
Exemple stack.c

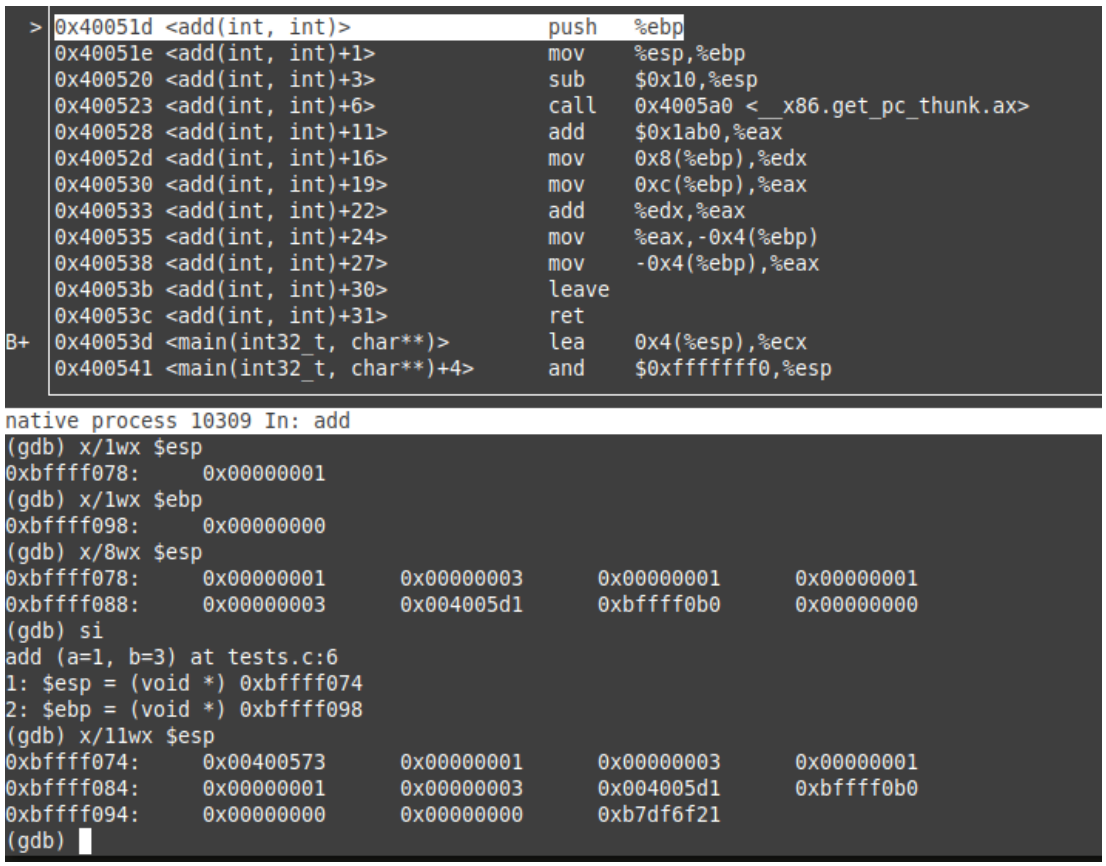
désassemblage par objdump -s -j stack

```
int add(int a, int b){
 51d:  55                       push   %ebp
 51e:  89 e5                    mov    %esp,%ebp
 520:  83 ec 10                 sub    $0x10,%esp
 523:  e8 78 00 00 00           call   5a0 <__x86.get_pc_thunk.ax>
 528:  05 b0 1a 00 00           add    $0x1ab0,%eax
       int c = a+b;
 52d:  8b 55 08                 mov    0x8(%ebp),%edx
 530:  8b 45 0c                 mov    0xc(%ebp),%eax
 533:  01 d0                    add    %edx,%eax
 535:  89 45 fc                 mov    %eax,-0x4(%ebp)
       return c;
 538:  8b 45 fc                 mov    -0x4(%ebp),%eax
}
 53b:  c9                       leave
 53c:  c3                       ret

0000053d <main>:

int main(int32_t argc, char** argv){
 53d:  8d 4c 24 04              lea    0x4(%esp),%ecx
 541:  83 e4 f0                 and    $0xfffffff0,%esp
 544:  ff 71 fc                 pushl  -0x4(%ecx)
 547:  55                       push   %ebp
 548:  89 e5                    mov    %esp,%ebp
 54a:  53                       push   %ebx
 54b:  51                       push   %ecx
 54c:  83 ec 10                 sub    $0x10,%esp
 54f:  e8 cc fe ff ff           call   420 <__x86.get_pc_thunk.bx>
 554:  81 c3 84 1a 00 00        add    $0x1a84,%ebx
       int i = 1;
 55a:  c7 45 ec 01 00 00 00     movl   $0x1,-0x14(%ebp)
       int j = 3;
 561:  c7 45 f0 03 00 00 00     movl   $0x3,-0x10(%ebp)
       int t = add(i,j);
 568:  ff 75 f0                 pushl  -0x10(%ebp)
 56b:  ff 75 ec                 pushl  -0x14(%ebp)
 56e:  e8 aa ff ff ff           call   51d <_Z3addii>
 573:  83 c4 08                 add    $0x8,%esp
 576:  89 45 f4                 mov    %eax,-0xc(%ebp)
       printf("%d + %d = %d\n",i,j,t);
 579:  ff 75 f4                 pushl  -0xc(%ebp)
 57c:  ff 75 f0                 pushl  -0x10(%ebp)
 57f:  ff 75 ec                 pushl  -0x14(%ebp)
 582:  8d 83 58 e6 ff ff        lea    -0x19a8(%ebx),%eax
 588:  50                       push   %eax
 589:  e8 22 fe ff ff           call   3b0 <printf@plt>
 58e:  83 c4 10                 add    $0x10,%esp
 591:  b8 00 00 00 00           mov    $0x0,%eax
 596:  8d 65 f8                 lea    -0x8(%ebp),%esp
 599:  59                       pop    %ecx
 59a:  5b                       pop    %ebx
 59b:  5d                       pop    %ebp
 59c:  8d 61 fc                 lea    -0x4(%ecx),%esp
 59f:  c3                       ret
```

État de la pile juste avant le `call <add(int,int)>`



État de la pile dans `call <add(int,int)>`



État de la pile juste avant la fin du `<add(int,int)>`