```
array_test_deb7:     file format elf64-x86-64


Disassembly of section .init:

0000000000400470 <_init>:
_init():
  400470:   48 83 ec 08             sub    $0x8,%rsp
  400474:   e8 93 00 00 00          callq  40050c <call_gmon_start>
  400479:   48 83 c4 08             add    $0x8,%rsp
  40047d:   c3                      retq

Disassembly of section .plt:

0000000000400480 <free@plt-0x10>:
  400480:   ff 35 72 07 20 00       pushq  0x200772(%rip)        # 600bf8 <_G
LOBAL_OFFSET_TABLE_+0x8>
  400486:   ff 25 74 07 20 00       jmpq   *0x200774(%rip)        # 600c00 <_
GLOBAL_OFFSET_TABLE_+0x10>
  40048c:   0f 1f 40 00             nopl   0x0(%rax)

0000000000400490 <free@plt>:
  400490:   ff 25 72 07 20 00       jmpq   *0x200772(%rip)        # 600c08 <_
GLOBAL_OFFSET_TABLE_+0x18>
  400496:   68 00 00 00 00          pushq  $0x0
  40049b:   e9 e0 ff ff ff          jmpq   400480 <_init+0x10>

00000000004004a0 <putchar@plt>:
  4004a0:   ff 25 6a 07 20 00       jmpq   *0x20076a(%rip)        # 600c10 <_
GLOBAL_OFFSET_TABLE_+0x20>
  4004a6:   68 01 00 00 00          pushq  $0x1
  4004ab:   e9 d0 ff ff ff          jmpq   400480 <_init+0x10>

00000000004004b0 <printf@plt>:
  4004b0:   ff 25 62 07 20 00       jmpq   *0x200762(%rip)        # 600c18 <_
GLOBAL_OFFSET_TABLE_+0x28>
  4004b6:   68 02 00 00 00          pushq  $0x2
  4004bb:   e9 c0 ff ff ff          jmpq   400480 <_init+0x10>

00000000004004c0 <__libc_start_main@plt>:
  4004c0:   ff 25 5a 07 20 00       jmpq   *0x20075a(%rip)        # 600c20 <_
GLOBAL_OFFSET_TABLE_+0x30>
  4004c6:   68 03 00 00 00          pushq  $0x3
  4004cb:   e9 b0 ff ff ff          jmpq   400480 <_init+0x10>

00000000004004d0 <malloc@plt>:
  4004d0:   ff 25 52 07 20 00       jmpq   *0x200752(%rip)        # 600c28 <_
GLOBAL_OFFSET_TABLE_+0x38>
  4004d6:   68 04 00 00 00          pushq  $0x4
  4004db:   e9 a0 ff ff ff          jmpq   400480 <_init+0x10>

Disassembly of section .text:

00000000004004e0 <_start>:
_start():
  4004e0:   31 ed                   xor    %ebp,%ebp
  4004e2:   49 89 d1                mov    %rdx,%r9
  4004e5:   5e                      pop    %rsi
  4004e6:   48 89 e2                mov    %rsp,%rdx
  4004e9:   48 83 e4 f0             and    $0xfffffffffffffff0,%rsp
  4004ed:   50                      push   %rax
  4004ee:   54                      push   %rsp
  4004ef:   49 c7 c0 20 08 40 00    mov    $0x400820,%r8
  4004f6:   48 c7 c1 30 08 40 00    mov    $0x400830,%rcx
  4004fd:   48 c7 c7 59 07 40 00    mov    $0x400759,%rdi
  400504:   e8 b7 ff ff ff          callq  4004c0 <__libc_start_main@plt>
  400509:   f4                      hlt
  40050a:   90                      nop
  40050b:   90                      nop

000000000040050c <call_gmon_start>:
call_gmon_start():
  40050c:   48 83 ec 08             sub    $0x8,%rsp
  400510:   48 8b 05 d1 06 20 00    mov    0x2006d1(%rip),%rax        # 600be
8 <_DYNAMIC+0x1e0>
  400517:   48 85 c0                test   %rax,%rax
  40051a:   74 02                   je     40051e <call_gmon_start+0x12>
  40051c:   ff d0                   callq  *%rax
  40051e:   48 83 c4 08             add    $0x8,%rsp
  400522:   c3                      retq
  400523:   90                      nop
  400524:   90                      nop
  400525:   90                      nop
  400526:   90                      nop
  400527:   90                      nop
  400528:   90                      nop
  400529:   90                      nop
  40052a:   90                      nop
  40052b:   90                      nop
  40052c:   90                      nop
  40052d:   90                      nop
  40052e:   90                      nop
  40052f:   90                      nop

0000000000400530 <deregister_tm_clones>:
deregister_tm_clones():
  400530:   b8 47 0c 60 00          mov    $0x600c47,%eax
  400535:   55                      push   %rbp
  400536:   48 2d 40 0c 60 00       sub    $0x600c40,%rax
  40053c:   48 83 f8 0e             cmp    $0xe,%rax
  400540:   48 89 e5                mov    %rsp,%rbp
  400543:   77 02                   ja     400547 <deregister_tm_clones+0x17>
  400545:   5d                      pop    %rbp
  400546:   c3                      retq
  400547:   b8 00 00 00 00          mov    $0x0,%eax
  40054c:   48 85 c0                test   %rax,%rax
  40054f:   74 f4                   je     400545 <deregister_tm_clones+0x15>
  400551:   5d                      pop    %rbp
  400552:   bf 40 0c 60 00          mov    $0x600c40,%edi
  400557:   ff e0                   jmpq   *%rax
  400559:   0f 1f 80 00 00 00 00    nopl   0x0(%rax)

0000000000400560 <register_tm_clones>:
register_tm_clones():
  400560:   b8 40 0c 60 00          mov    $0x600c40,%eax
  400565:   55                      push   %rbp
  400566:   48 2d 40 0c 60 00       sub    $0x600c40,%rax
  40056c:   48 c1 f8 03             sar    $0x3,%rax
  400570:   48 89 e5                mov    %rsp,%rbp
  400573:   48 89 c2                mov    %rax,%rdx
  400576:   48 c1 ea 3f             shr    $0x3f,%rdx
  40057a:   48 01 d0                add    %rdx,%rax
  40057d:   48 89 c6                mov    %rax,%rsi
  400580:   48 d1 fe                sar    %rsi
  400583:   75 02                   jne    400587 <register_tm_clones+0x27>
  400585:   5d                      pop    %rbp
  400586:   c3                      retq
  400587:   ba 00 00 00 00          mov    $0x0,%edx
  40058c:   48 85 d2                test   %rdx,%rdx
  40058f:   74 f4                   je     400585 <register_tm_clones+0x25>
  400591:   5d                      pop    %rbp
  400592:   bf 40 0c 60 00          mov    $0x600c40,%edi
  400597:   ff e2                   jmpq   *%rdx
  400599:   0f 1f 80 00 00 00 00    nopl   0x0(%rax)

00000000004005a0 <__do_global_dtors_aux>:
__do_global_dtors_aux():
  4005a0:   80 3d 99 06 20 00 00    cmpb   $0x0,0x200699(%rip)        # 600c4
0 <__bss_start>
  4005a7:   75 11                   jne    4005ba <__do_global_dtors_aux+0x1a
>
  4005a9:   55                      push   %rbp
  4005aa:   48 89 e5                mov    %rsp,%rbp
  4005ad:   e8 7e ff ff ff          callq  400530 <deregister_tm_clones>
  4005b2:   5d                      pop    %rbp
  4005b3:   c6 05 86 06 20 00 01    movb   $0x1,0x200686(%rip)        # 600c4
0 <__bss_start>
  4005ba:   f3 c3                   repz retq
  4005bc:   0f 1f 40 00             nopl   0x0(%rax)

00000000004005c0 <frame_dummy>:
frame_dummy():
  4005c0:   48 83 3d 38 04 20 00    cmpq   $0x0,0x200438(%rip)        # 600a0
0 <__JCR_END__>
  4005c7:   00
  4005c8:   74 1b                   je     4005e5 <frame_dummy+0x25>
  4005ca:   b8 00 00 00 00          mov    $0x0,%eax
  4005cf:   48 85 c0                test   %rax,%rax
  4005d2:   74 11                   je     4005e5 <frame_dummy+0x25>
  4005d4:   55                      push   %rbp
  4005d5:   bf 00 0a 60 00          mov    $0x600a00,%edi
  4005da:   48 89 e5                mov    %rsp,%rbp
  4005dd:   ff d0                   callq  *%rax
  4005df:   5d                      pop    %rbp
  4005e0:   e9 7b ff ff ff          jmpq   400560 <register_tm_clones>
  4005e5:   e9 76 ff ff ff          jmpq   400560 <register_tm_clones>
  4005ea:   90                      nop
  4005eb:   90                      nop

00000000004005ec <dead_compute01>:
dead_compute01():
/root/deadwriteBenchmark/dead_array_test/main.c:4
#include <stdio.h>
#include <stdlib.h>

int dead_compute01( int *a, int size){
  4005ec:   55                      push   %rbp
  4005ed:   48 89 e5                mov    %rsp,%rbp
  4005f0:   48 83 ec 30             sub    $0x30,%rsp
  4005f4:   48 89 7d d8             mov    %rdi,-0x28(%rbp)
  4005f8:   89 75 d4                mov    %esi,-0x2c(%rbp)
/root/deadwriteBenchmark/dead_array_test/main.c:6

   int j=0 ;
  4005fb:   c7 45 fc 00 00 00 00    movl   $0x0,-0x4(%rbp)
/root/deadwriteBenchmark/dead_array_test/main.c:7
   long count=0;
  400602:   48 c7 45 f0 00 00 00    movq   $0x0,-0x10(%rbp)
  400609:   00
/root/deadwriteBenchmark/dead_array_test/main.c:9

   for (j=0; j<size*3; j++){
  40060a:   c7 45 fc 00 00 00 00    movl   $0x0,-0x4(%rbp)
  400611:   e9 2a 01 00 00          jmpq   400740 <dead_compute01+0x154>
/root/deadwriteBenchmark/dead_array_test/main.c:11

    int i, tmp=0;
  400616:   c7 45 e8 00 00 00 00    movl   $0x0,-0x18(%rbp)
/root/deadwriteBenchmark/dead_array_test/main.c:13
```

```
    for (i = 0; i<size; i++){
  40061d:   c7 45 ec 00 00 00 00    movl   $0x0,-0x14(%rbp)
  400624:   eb 2a                   jmp    400650 <dead_compute01+0x64>
/root/deadwriteBenchmark/dead_array_test/main.c:14
        a[i] = i*i-tmp;
  400626:   8b 45 ec                mov    -0x14(%rbp),%eax
  400629:   48 98                   cltq
  40062b:   48 8d 14 85 00 00 00    lea    0x0(,%rax,4),%rdx
  400632:   00
  400633:   48 8b 45 d8             mov    -0x28(%rbp),%rax
  400637:   48 01 c2                add    %rax,%rdx
  40063a:   8b 45 ec                mov    -0x14(%rbp),%eax
  40063d:   0f af 45 ec             imul   -0x14(%rbp),%eax
  400641:   2b 45 e8                sub    -0x18(%rbp),%eax
  400644:   89 02                   mov    %eax,(%rdx)
/root/deadwriteBenchmark/dead_array_test/main.c:15
            tmp += i;
  400646:   8b 45 ec                mov    -0x14(%rbp),%eax
  400649:   01 45 e8                add    %eax,-0x18(%rbp)
/root/deadwriteBenchmark/dead_array_test/main.c:13

   for (j=0; j<size*3; j++){

    int i, tmp=0;

    for (i = 0; i<size; i++){
  40064c:   83 45 ec 01             addl   $0x1,-0x14(%rbp)
  400650:   8b 45 ec                mov    -0x14(%rbp),%eax
  400653:   3b 45 d4                cmp    -0x2c(%rbp),%eax
  400656:   7c ce                   jl     400626 <dead_compute01+0x3a>
/root/deadwriteBenchmark/dead_array_test/main.c:18
        a[i] = i*i-tmp;
            tmp += i;
    }

    printf("%s: %d\t", __FUNCTION__, a[(i+j)/4]);
  400658:   8b 45 fc                mov    -0x4(%rbp),%eax
  40065b:   8b 55 ec                mov    -0x14(%rbp),%edx
  40065e:   01 d0                   add    %edx,%eax
  400660:   8d 50 03                lea    0x3(%rax),%edx
  400663:   85 c0                   test   %eax,%eax
  400665:   0f 48 c2                cmovs  %edx,%eax
  400668:   c1 f8 02                sar    $0x2,%eax
  40066b:   48 98                   cltq
  40066d:   48 8d 14 85 00 00 00    lea    0x0(,%rax,4),%rdx
  400674:   00
  400675:   48 8b 45 d8             mov    -0x28(%rbp),%rax
  400679:   48 01 d0                add    %rdx,%rax
  40067c:   8b 00                   mov    (%rax),%eax
  40067e:   89 c2                   mov    %eax,%edx
  400680:   be de 08 40 00          mov    $0x4008de,%esi
  400685:   bf cc 08 40 00          mov    $0x4008cc,%edi
  40068a:   b8 00 00 00 00          mov    $0x0,%eax
  40068f:   e8 1c fe ff ff          callq  4004b0 <printf@plt>
/root/deadwriteBenchmark/dead_array_test/main.c:19
    if(j%10 == 0) printf("\n");
  400694:   8b 4d fc                mov    -0x4(%rbp),%ecx
  400697:   ba 67 66 66 66          mov    $0x66666667,%edx
  40069c:   89 c8                   mov    %ecx,%eax
  40069e:   f7 ea                   imul   %edx
  4006a0:   c1 fa 02                sar    $0x2,%edx
  4006a3:   89 c8                   mov    %ecx,%eax
  4006a5:   c1 f8 1f                sar    $0x1f,%eax
  4006a8:   29 c2                   sub    %eax,%edx
  4006aa:   89 d0                   mov    %edx,%eax
  4006ac:   c1 e0 02                shl    $0x2,%eax
  4006af:   01 d0                   add    %edx,%eax
  4006b1:   01 c0                   add    %eax,%eax
  4006b3:   89 ca                   mov    %ecx,%edx
  4006b5:   29 c2                   sub    %eax,%edx
  4006b7:   85 d2                   test   %edx,%edx
  4006b9:   75 0a                   jne    4006c5 <dead_compute01+0xd9>
  4006bb:   bf 0a 00 00 00          mov    $0xa,%edi
  4006c0:   e8 db fd ff ff          callq  4004a0 <putchar@plt>
/root/deadwriteBenchmark/dead_array_test/main.c:21

    for (i =0 ; i< size; i++){
  4006c5:   c7 45 ec 00 00 00 00    movl   $0x0,-0x14(%rbp)
  4006cc:   eb 3d                   jmp    40070b <dead_compute01+0x11f>
/root/deadwriteBenchmark/dead_array_test/main.c:22
        a[i] = i*i/2;
  4006ce:   8b 45 ec                mov    -0x14(%rbp),%eax
  4006d1:   48 98                   cltq
  4006d3:   48 8d 14 85 00 00 00    lea    0x0(,%rax,4),%rdx
  4006da:   00
  4006db:   48 8b 45 d8             mov    -0x28(%rbp),%rax
  4006df:   48 01 c2                add    %rax,%rdx
  4006e2:   8b 45 ec                mov    -0x14(%rbp),%eax
  4006e5:   0f af 45 ec             imul   -0x14(%rbp),%eax
  4006e9:   89 c1                   mov    %eax,%ecx
  4006eb:   c1 e9 1f                shr    $0x1f,%ecx
  4006ee:   01 c8                   add    %ecx,%eax
  4006f0:   d1 f8                   sar    %eax
  4006f2:   89 02                   mov    %eax,(%rdx)
/root/deadwriteBenchmark/dead_array_test/main.c:23
             tmp = tmp + 2*i -200;
  4006f4:   8b 45 ec                mov    -0x14(%rbp),%eax
  4006f7:   8d 14 00                lea    (%rax,%rax,1),%edx
  4006fa:   8b 45 e8                mov    -0x18(%rbp),%eax
  4006fd:   01 d0                   add    %edx,%eax
  4006ff:   2d c8 00 00 00          sub    $0xc8,%eax
  400704:   89 45 e8                mov    %eax,-0x18(%rbp)
/root/deadwriteBenchmark/dead_array_test/main.c:21
    }

    printf("%s: %d\t", __FUNCTION__, a[(i+j)/4]);
    if(j%10 == 0) printf("\n");

    for (i =0 ; i< size; i++){
  400707:   83 45 ec 01             addl   $0x1,-0x14(%rbp)
  40070b:   8b 45 ec                mov    -0x14(%rbp),%eax
  40070e:   3b 45 d4                cmp    -0x2c(%rbp),%eax
  400711:   7c bb                   jl     4006ce <dead_compute01+0xe2>
/root/deadwriteBenchmark/dead_array_test/main.c:26
        a[i] = i*i/2;
             tmp = tmp + 2*i -200;
    }

    count += tmp * 2 - tmp / 3;
  400713:   8b 45 e8                mov    -0x18(%rbp),%eax
  400716:   8d 34 00                lea    (%rax,%rax,1),%esi
  400719:   8b 4d e8                mov    -0x18(%rbp),%ecx
  40071c:   ba 56 55 55 55          mov    $0x55555556,%edx
  400721:   89 c8                   mov    %ecx,%eax
  400723:   f7 ea                   imul   %edx
  400725:   89 c8                   mov    %ecx,%eax
  400727:   c1 f8 1f                sar    $0x1f,%eax
  40072a:   89 d1                   mov    %edx,%ecx
  40072c:   29 c1                   sub    %eax,%ecx
  40072e:   89 c8                   mov    %ecx,%eax
  400730:   89 f2                   mov    %esi,%edx
  400732:   29 c2                   sub    %eax,%edx
  400734:   89 d0                   mov    %edx,%eax
  400736:   48 98                   cltq
  400738:   48 01 45 f0             add    %rax,-0x10(%rbp)
/root/deadwriteBenchmark/dead_array_test/main.c:9
int dead_compute01( int *a, int size){

   int j=0 ;
   long count=0;

   for (j=0; j<size*3; j++){
  40073c:   83 45 fc 01             addl   $0x1,-0x4(%rbp)
  400740:   8b 55 d4                mov    -0x2c(%rbp),%edx
  400743:   89 d0                   mov    %edx,%eax
  400745:   01 c0                   add    %eax,%eax
  400747:   01 d0                   add    %edx,%eax
  400749:   3b 45 fc                cmp    -0x4(%rbp),%eax
  40074c:   0f 8f c4 fe ff ff       jg     400616 <dead_compute01+0x2a>
/root/deadwriteBenchmark/dead_array_test/main.c:29
    }

    count += tmp * 2 - tmp / 3;

    }
    return 0;
  400752:   b8 00 00 00 00          mov    $0x0,%eax
/root/deadwriteBenchmark/dead_array_test/main.c:30
}
  400757:   c9                      leaveq
  400758:   c3                      retq

0000000000400759 <main>:
main():
/root/deadwriteBenchmark/dead_array_test/main.c:32

int main(){
  400759:   55                      push   %rbp
  40075a:   48 89 e5                mov    %rsp,%rbp
  40075d:   48 83 ec 10             sub    $0x10,%rsp
/root/deadwriteBenchmark/dead_array_test/main.c:33
    int size = 2000;
  400761:   c7 45 f8 d0 07 00 00    movl   $0x7d0,-0x8(%rbp)
/root/deadwriteBenchmark/dead_array_test/main.c:34
    int *a = malloc (size * sizeof(int));
  400768:   8b 45 f8                mov    -0x8(%rbp),%eax
  40076b:   48 98                   cltq
  40076d:   48 c1 e0 02             shl    $0x2,%rax
  400771:   48 89 c7                mov    %rax,%rdi
  400774:   e8 57 fd ff ff          callq  4004d0 <malloc@plt>
  400779:   48 89 45 f0             mov    %rax,-0x10(%rbp)
/root/deadwriteBenchmark/dead_array_test/main.c:35
    dead_compute01( a, size);
  40077d:   8b 55 f8                mov    -0x8(%rbp),%edx
  400780:   48 8b 45 f0             mov    -0x10(%rbp),%rax
  400784:   89 d6                   mov    %edx,%esi
  400786:   48 89 c7                mov    %rax,%rdi
  400789:   e8 5e fe ff ff          callq  4005ec <dead_compute01>
/root/deadwriteBenchmark/dead_array_test/main.c:37
    int i;
    for (i = size-1; i>=0; i--){
  40078e:   8b 45 f8                mov    -0x8(%rbp),%eax
  400791:   83 e8 01                sub    $0x1,%eax
  400794:   89 45 fc                mov    %eax,-0x4(%rbp)
  400797:   eb 5f                   jmp    4007f8 <main+0x9f>
/root/deadwriteBenchmark/dead_array_test/main.c:38
       printf("a[%d]=%d\t", i, a[i]);
  400799:   8b 45 fc                mov    -0x4(%rbp),%eax
  40079c:   48 98                   cltq
```

```
  40079e:   48 8d 14 85 00 00 00    lea    0x0(,%rax,4),%rdx
  4007a5:   00
  4007a6:   48 8b 45 f0             mov    -0x10(%rbp),%rax
  4007aa:   48 01 d0                add    %rdx,%rax
  4007ad:   8b 10                   mov    (%rax),%edx
  4007af:   8b 45 fc                mov    -0x4(%rbp),%eax
  4007b2:   89 c6                   mov    %eax,%esi
  4007b4:   bf d4 08 40 00          mov    $0x4008d4,%edi
  4007b9:   b8 00 00 00 00          mov    $0x0,%eax
  4007be:   e8 ed fc ff ff          callq  4004b0 <printf@plt>
/root/deadwriteBenchmark/dead_array_test/main.c:39
        if(i%10==0) printf("\n");
  4007c3:   8b 4d fc                mov    -0x4(%rbp),%ecx
  4007c6:   ba 67 66 66 66          mov    $0x66666667,%edx
  4007cb:   89 c8                   mov    %ecx,%eax
  4007cd:   f7 ea                   imul   %edx
  4007cf:   c1 fa 02                sar    $0x2,%edx
  4007d2:   89 c8                   mov    %ecx,%eax
  4007d4:   c1 f8 1f                sar    $0x1f,%eax
  4007d7:   29 c2                   sub    %eax,%edx
  4007d9:   89 d0                   mov    %edx,%eax
  4007db:   c1 e0 02                shl    $0x2,%eax
  4007de:   01 d0                   add    %edx,%eax
  4007e0:   01 c0                   add    %eax,%eax
  4007e2:   89 ca                   mov    %ecx,%edx
  4007e4:   29 c2                   sub    %eax,%edx
  4007e6:   85 d2                   test   %edx,%edx
  4007e8:   75 0a                   jne    4007f4 <main+0x9b>
  4007ea:   bf 0a 00 00 00          mov    $0xa,%edi
  4007ef:   e8 ac fc ff ff          callq  4004a0 <putchar@plt>
/root/deadwriteBenchmark/dead_array_test/main.c:37
int main(){
    int size = 2000;
    int *a = malloc (size * sizeof(int));
    dead_compute01( a, size);
    int i;
    for (i = size-1; i>=0; i--){
  4007f4:   83 6d fc 01             subl   $0x1,-0x4(%rbp)
  4007f8:   83 7d fc 00             cmpl   $0x0,-0x4(%rbp)
  4007fc:   79 9b                   jns    400799 <main+0x40>
/root/deadwriteBenchmark/dead_array_test/main.c:41
        printf("a[%d]=%d\t", i, a[i]);
        if(i%10==0) printf("\n");
    }
    free(a);
  4007fe:   48 8b 45 f0             mov    -0x10(%rbp),%rax
  400802:   48 89 c7                mov    %rax,%rdi
  400805:   e8 86 fc ff ff          callq  400490 <free@plt>
/root/deadwriteBenchmark/dead_array_test/main.c:42
    return 0;
  40080a:   b8 00 00 00 00          mov    $0x0,%eax
/root/deadwriteBenchmark/dead_array_test/main.c:43
}
  40080f:   c9                      leaveq
  400810:   c3                      retq
  400811:   90                      nop
  400812:   90                      nop
  400813:   90                      nop
  400814:   90                      nop
  400815:   90                      nop
  400816:   90                      nop
  400817:   90                      nop
  400818:   90                      nop
  400819:   90                      nop
  40081a:   90                      nop
  40081b:   90                      nop
  40081c:   90                      nop
  40081d:   90                      nop
  40081e:   90                      nop
  40081f:   90                      nop

0000000000400820 <__libc_csu_fini>:
__libc_csu_fini():
  400820:   f3 c3                   repz retq
  400822:   66 66 66 66 66 2e 0f    data32 data32 data32 data32 nopw %cs:0x0(
%rax,%rax,1)
  400829:   1f 84 00 00 00 00 00

0000000000400830 <__libc_csu_init>:
__libc_csu_init():
  400830:   48 89 6c 24 d8          mov    %rbp,-0x28(%rsp)
  400835:   4c 89 64 24 e0          mov    %r12,-0x20(%rsp)
  40083a:   48 8d 2d b7 01 20 00    lea    0x2001b7(%rip),%rbp        # 6009f
8 <__init_array_end>
  400841:   4c 8d 25 a8 01 20 00    lea    0x2001a8(%rip),%r12        # 6009f
0 <__frame_dummy_init_array_entry>
  400848:   4c 89 6c 24 e8          mov    %r13,-0x18(%rsp)
  40084d:   4c 89 74 24 f0          mov    %r14,-0x10(%rsp)
  400852:   4c 89 7c 24 f8          mov    %r15,-0x8(%rsp)
  400857:   48 89 5c 24 d0          mov    %rbx,-0x30(%rsp)
  40085c:   48 83 ec 38             sub    $0x38,%rsp
  400860:   4c 29 e5                sub    %r12,%rbp
  400863:   41 89 fd                mov    %edi,%r13d
  400866:   49 89 f6                mov    %rsi,%r14
  400869:   48 c1 fd 03             sar    $0x3,%rbp
  40086d:   49 89 d7                mov    %rdx,%r15
  400870:   e8 fb fb ff ff          callq  400470 <_init>
  400875:   48 85 ed                test   %rbp,%rbp
  400878:   74 1c                   je     400896 <__libc_csu_init+0x66>
  40087a:   31 db                   xor    %ebx,%ebx
  40087c:   0f 1f 40 00             nopl   0x0(%rax)
  400880:   4c 89 fa                mov    %r15,%rdx
  400883:   4c 89 f6                mov    %r14,%rsi
  400886:   44 89 ef                mov    %r13d,%edi
  400889:   41 ff 14 dc             callq  *(%r12,%rbx,8)
  40088d:   48 83 c3 01             add    $0x1,%rbx
  400891:   48 39 eb                cmp    %rbp,%rbx
  400894:   72 ea                   jb     400880 <__libc_csu_init+0x50>
  400896:   48 8b 5c 24 08          mov    0x8(%rsp),%rbx
  40089b:   48 8b 6c 24 10          mov    0x10(%rsp),%rbp
  4008a0:   4c 8b 64 24 18          mov    0x18(%rsp),%r12
  4008a5:   4c 8b 6c 24 20          mov    0x20(%rsp),%r13
  4008aa:   4c 8b 74 24 28          mov    0x28(%rsp),%r14
  4008af:   4c 8b 7c 24 30          mov    0x30(%rsp),%r15
  4008b4:   48 83 c4 38             add    $0x38,%rsp
  4008b8:   c3                      retq
  4008b9:   90                      nop
  4008ba:   90                      nop
  4008bb:   90                      nop

Disassembly of section .fini:

00000000004008bc <_fini>:
_fini():
  4008bc:   48 83 ec 08             sub    $0x8,%rsp
  4008c0:   48 83 c4 08             add    $0x8,%rsp
  4008c4:   c3                      retq
```