Control Flow Graph

```
0x40105b (0x40105b)
0x0040105b:  nop   dword ptr [rax + rax]
```

```
0x401060 (0x401060)  _start
0x00401060:  endbr64
0x00401064:  xor    ebp, ebp
0x00401066:  mov    r9, rdx
0x00401069:  pop    rsi
0x0040106a:  mov    rdx, rsp
0x0040106d:  and    rsp, 0xfffffffffffffff0
0x00401071:  push   rax
0x00401071:  push   rsp
0x00401072:  lea    r8, [rip + 0x166]
0x00401073:  lea    rcx, [rip + 0xef]
0x0040107a:  lea    rdi, [rip + 0xc1]
0x00401081:  call   qword ptr [rip + 0x2f52]
```

```
0x40116f (0x40116f)
0x0040116f:  nop
```

```
0x500000 (0x500000)  __libc_start_main SIMP NORET
```

```
0x4011d5 (0x4011d5)
0x004011d5:  nop word ptr cs:[rax + rax]
```

```
0x4010f9 (0x4010f9)
0x004010f9:  nop dword ptr [rax]
```

```
0x40108e (0x40108e)
0x0040108e:  hlt
```

```
0x4011e8 (0x4011e8)  _fini
0x004011e8:  endbr64
0x004011ec:  sub    rsp, 8
0x004011f0:  add    rsp, 8
0x004011f4:  ret
```

```
0x401100 (0x401100)  __do_global_dtors_aux
0x00401100:  endbr64
0x00401104:  cmp    byte ptr [rip + 0x2f05], 0
0x0040110b:  jne    0x401138
```

```
0x401135 (0x401135)
0x00401135:  nop dword ptr [rax]
```

```
0x401138 (0x401100)  __do_global_dtors_aux+0x38
0x00401138:  ret
```

```
0x40110d (0x401100)  __do_global_dtors_aux+0xd
0x0040110d:  push   rbp
0x0040110e:  cmp    qword ptr [rip + 0x2ee2], 0
0x00401116:  mov    rbp, rsp
0x00401119:  je     0x401127
```

```
0x40111b (0x401100)  __do_global_dtors_aux+0x1b
0x0040111b:  mov    rdi, qword ptr [rip + 0x2ee6]
0x00401122:  call   0x401040      ; __cxa_finalize
```

```
0x40103f (0x40103f)
0x0040103f:  nop
```

```
0x401139 (0x401139)
0x00401139:  nop dword ptr [rax]
```

```
0x401170 (0x401170)  __libc_csu_init
0x00401170:  endbr64
0x00401174:  push   r15
0x00401176:  lea    r15, [rip + 0x2c3b]
0x0040117d:  push   r14
0x0040117f:  mov    r14, rdx
0x00401182:  push   r13
0x00401184:  mov    r13, rsi
0x00401187:  push   r12
0x00401189:  mov    r12d, edi
0x0040118c:  push   rbp
0x0040118d:  lea    rbp, [rip + 0x2c2c]
0x00401194:  push   rbx
0x00401195:  sub    rbp, r15
0x00401198:  sub    rsp, 8
0x0040119c:  call   0x401000      ; _init
```

```
0x401149 (0x401149)  main
0x00401149:  endbr64
0x0040114d:  push   rbp
0x0040114e:  mov    rbp, rsp
0x00401151:  sub    rsp, 0x10
0x00401155:  mov    dword ptr [rbp - 4], edi
0x00401158:  mov    qword ptr [rbp - 0x10], rsi
0x0040115c:  lea    rdi, [rip + 0xea1]
0x00401163:  call   0x401050      ; puts
```

```
0x40104b (0x40104b)
0x0040104b:  nop dword ptr [rax + rax]
```

```
0x4011e0 (0x4011e0)  __libc_csu_fini
0x004011e0:  endbr64
0x004011e4:  ret
```

```
0x401040 (0x401040)
0x00401040:  endbr64
0x00401044:  bnd jmp qword ptr [rip + 0x2fad]
```

```
0x401127 (0x401100)  __do_global_dtors_aux+0x27
0x00401127:  call 0x401090   ; deregister_tm_clones
```

```
0x40108f (0x40108f)
0x0040108f:  nop
```

```
0x401140 (0x401140)  frame_dummy
0x00401140:  endbr64
0x00401144:  jmp    0x4010c0
```

```
0x4010b9 (0x4010b9)
0x004010b9:  nop dword ptr [rax]
```

```
0x401000 (0x401000)  _init
0x00401000:  endbr64
0x00401004:  sub    rsp, 8
0x00401008:  mov    rax, qword ptr [rip + 0x2fd9]
0x0040100f:  test   rax, rax
0x00401012:  je     0x401016
```

```
0x401050 (0x401050)
0x00401050:  endbr64
0x00401054:  bnd jmp qword ptr [rip + 0x2f75]
```

```
0x401090 (0x401090)  deregister_tm_clones
0x00401090:  lea    rdi, [rip + 0x2f79]
0x00401097:  lea    rax, [rip + 0x2f72]
0x0040109e:  cmp    rax, rdi
0x004010a1:  je     0x4010b8
```

```
0x40102d (0x40102d)
0x0040102d:  nop dword ptr [rax]
```

```
0x4010c0 (0x4010c0)  register_tm_clones
0x004010c0:  lea    rdi, [rip + 0x2f49]
0x004010c7:  lea    rsi, [rip + 0x2f42]
0x004010ce:  sub    rsi, rdi
0x004010d1:  mov    rax, rsi
0x004010d4:  shr    rsi, 0x3f
0x004010d8:  sar    rax, 3
0x004010dc:  add    rsi, rax
0x004010df:  sar    rsi, 1
0x004010e2:  je     0x4010f8
```

```
0x401014 (0x401000)  _init+0x14
0x00401014:  call rax
```

```
0x500008 (0x500008)  puts SIMP
```

```
0x401016 (0x401000)  _init+0x16
0x00401016:  add rsp, 8
0x0040101a:  ret
```

```
0x4010b1 (0x401090)  deregister_tm_clones+0x21
0x004010b1:  nop dword ptr [rax]
```

```
0x4010a3 (0x401090)  deregister_tm_clones+0x13
0x004010a3:  mov    rax, qword ptr [rip + 0x2f2e]
0x004010aa:  test   rax, rax
0x004010ad:  je     0x4010b8
```

```
0x401030 (0x401030)
0x00401030:  endbr64
0x00401034:  push   0
0x00401039:  bnd jmp 0x401020
```

```
0x4010f2 (0x4010c0)  register_tm_clones+0x32
0x004010f2:  nop word ptr [rax + rax]
```

```
0x4010e4 (0x4010c0)  register_tm_clones+0x24
0x004010e4:  mov    rax, qword ptr [rip + 0x2f05]
0x004010eb:  test   rax, rax
0x004010ee:  je     0x4010f8
```

```
0x401168 (0x401149)  main+0x1f
0x00401168:  mov    eax, 0
0x0040116d:  leave
0x0040116e:  ret
```

```
0x4011a1 (0x401170)  __libc_csu_init+0x31
0x004011a1:  sar rbp, 3
0x004011a5:  je  0x4011c6
```

```
0x4010b8 (0x401090)  deregister_tm_clones+0x28
0x004010b8:  ret
```

```
0x4010af (0x401090)  deregister_tm_clones+0x1f
0x004010af:  jmp rax
```

```
0x401020 (0x401020)
0x00401020:  push   qword ptr [rip + 0x2fa9]
0x00401026:  bnd jmp qword ptr [rip + 0x2f9b]
```

```
0x4010f8 (0x4010c0)  register_tm_clones+0x38
0x004010f8:  ret
```

```
0x4010f0 (0x4010c0)  register_tm_clones+0x30
0x004010f0:  jmp rax
```

```
0x601050 (0x601050)  UnresolvableJumpTarget SIMP
```

```
0x4011a7 (0x401170)  __libc_csu_init+0x37
0x004011a7:  xor    ebx, ebx
0x004011a9:  nop    dword ptr [rax]
0x004011b0:  mov    rdx, r14
0x004011b3:  mov    rsi, r13
0x004011b6:  mov    edi, r12d
0x004011b9:  call   qword ptr [r15 + rbx*8]
```

```
0x4011b0 (0x401170)  __libc_csu_init+0x40
0x004011b0:  mov    rdx, r14
0x004011b3:  mov    rsi, r13
0x004011b6:  mov    edi, r12d
0x004011b9:  call   qword ptr [r15 + rbx*8]
```

```
0x40112c (0x401100)  __do_global_dtors_aux+0x2c
0x0040112c:  mov    byte ptr [rip + 0x2edd], 1
0x00401133:  pop    rbp
0x00401134:  ret
```

```
0x601058 (0x601058)  UnresolvableCallTarget SIMP
```

```
0x4011bd (0x401170)  __libc_csu_init+0x4d
0x004011bd:  add    rbx, 1
0x004011c1:  cmp    rbp, rbx
0x004011c4:  jne    0x4011b0
```

```
0x4011c6 (0x401170)  __libc_csu_init+0x56
0x004011c6:  add    rsp, 8
0x004011ca:  pop    rbx
0x004011cb:  pop    rbp
0x004011cc:  pop    r12
0x004011ce:  pop    r13
0x004011d0:  pop    r14
0x004011d2:  pop    r15
0x004011d4:  ret
```