**main() Source Code:**

#include <stdio.h>

#include <stdlib.h>

int custom(int);

int main(int argc, char\*\* argv)

{

int user\_input;

user\_input = atoi(argv[1]);

printf("%d to the 5th power is %d\n", user\_input, custom(user\_input));

return 0;

}

**custom() Source Code:**

int custom(int user\_input)

{

int result = 1;

int iterations = 5;

for (int i = 0; i <= iterations; ++i)

{

result = result \* user\_input;

}

return result;

}

**Assembly Code from lab01.s file:**

.file "lab01.c"

.text

.section .rodata

.LC0:

.string "%d to the 5th power is %d\n"

.text

.globl main

.type main, @function

main:

.LFB5:

.cfi\_startproc

pushq %rbp

.cfi\_def\_cfa\_offset 16

.cfi\_offset 6, -16

movq %rsp, %rbp

.cfi\_def\_cfa\_register 6

subq $32, %rsp

movl %edi, -20(%rbp)

movq %rsi, -32(%rbp)

movq -32(%rbp), %rax

addq $8, %rax

movq (%rax), %rax

movq %rax, %rdi

call atoi@PLT

movl %eax, -4(%rbp)

movl -4(%rbp), %eax

movl %eax, %edi

call custom@PLT

movl %eax, %edx

movl -4(%rbp), %eax

movl %eax, %esi

leaq .LC0(%rip), %rdi

movl $0, %eax

call printf@PLT

movl $0, %eax

leave

.cfi\_def\_cfa 7, 8

ret

.cfi\_endproc

.LFE5:

.size main, .-main

.ident "GCC: (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0"

.section .note.GNU-stack,"",@progbits

**Disassembly Code:**

lab01: file format elf64-x86-64

Disassembly of section .init:

0000000000000528 <\_init>:

528: 48 83 ec 08 sub $0x8,%rsp

52c: 48 8b 05 b5 0a 20 00 mov 0x200ab5(%rip),%rax # 200fe8 <\_\_gmon\_start\_\_>

533: 48 85 c0 test %rax,%rax

536: 74 02 je 53a <\_init+0x12>

538: ff d0 callq \*%rax

53a: 48 83 c4 08 add $0x8,%rsp

53e: c3 retq

Disassembly of section .plt:

0000000000000540 <.plt>:

540: ff 35 72 0a 20 00 pushq 0x200a72(%rip) # 200fb8 <\_GLOBAL\_OFFSET\_TABLE\_+0x8>

546: ff 25 74 0a 20 00 jmpq \*0x200a74(%rip) # 200fc0 <\_GLOBAL\_OFFSET\_TABLE\_+0x10>

54c: 0f 1f 40 00 nopl 0x0(%rax)

0000000000000550 <printf@plt>:

The next line

550: ff 25 72 0a 20 00 jmpq \*0x200a72(%rip) # 200fc8 <printf@GLIBC\_2.2.5>

556: 68 00 00 00 00 pushq $0x0

55b: e9 e0 ff ff ff jmpq 540 <.plt>

0000000000000560 <atoi@plt>:

560: ff 25 6a 0a 20 00 jmpq \*0x200a6a(%rip) # 200fd0 <atoi@GLIBC\_2.2.5>

566: 68 01 00 00 00 pushq $0x1

56b: e9 d0 ff ff ff jmpq 540 <.plt>

Disassembly of section .plt.got:

0000000000000570 <\_\_cxa\_finalize@plt>:

570: ff 25 82 0a 20 00 jmpq \*0x200a82(%rip) # 200ff8 <\_\_cxa\_finalize@GLIBC\_2.2.5>

576: 66 90 xchg %ax,%ax

Disassembly of section .text:

0000000000000580 <\_start>:

580: 31 ed xor %ebp,%ebp

582: 49 89 d1 mov %rdx,%r9

585: 5e pop %rsi

586: 48 89 e2 mov %rsp,%rdx

589: 48 83 e4 f0 and $0xfffffffffffffff0,%rsp

58d: 50 push %rax

58e: 54 push %rsp

58f: 4c 8d 05 fa 01 00 00 lea 0x1fa(%rip),%r8 # 790 <\_\_libc\_csu\_fini>

596: 48 8d 0d 83 01 00 00 lea 0x183(%rip),%rcx # 720 <\_\_libc\_csu\_init>

59d: 48 8d 3d e6 00 00 00 lea 0xe6(%rip),%rdi # 68a <main>

5a4: ff 15 36 0a 20 00 callq \*0x200a36(%rip) # 200fe0 <\_\_libc\_start\_main@GLIBC\_2.2.5>

5aa: f4 hlt

5ab: 0f 1f 44 00 00 nopl 0x0(%rax,%rax,1)

00000000000005b0 <deregister\_tm\_clones>:

5b0: 48 8d 3d 59 0a 20 00 lea 0x200a59(%rip),%rdi # 201010 <\_\_TMC\_END\_\_>

5b7: 55 push %rbp

5b8: 48 8d 05 51 0a 20 00 lea 0x200a51(%rip),%rax # 201010 <\_\_TMC\_END\_\_>

5bf: 48 39 f8 cmp %rdi,%rax

5c2: 48 89 e5 mov %rsp,%rbp

5c5: 74 19 je 5e0 <deregister\_tm\_clones+0x30>

5c7: 48 8b 05 0a 0a 20 00 mov 0x200a0a(%rip),%rax # 200fd8 <\_ITM\_deregisterTMCloneTable>

5ce: 48 85 c0 test %rax,%rax

5d1: 74 0d je 5e0 <deregister\_tm\_clones+0x30>

5d3: 5d pop %rbp

5d4: ff e0 jmpq \*%rax

5d6: 66 2e 0f 1f 84 00 00 nopw %cs:0x0(%rax,%rax,1)

5dd: 00 00 00

5e0: 5d pop %rbp

5e1: c3 retq

5e2: 0f 1f 40 00 nopl 0x0(%rax)

5e6: 66 2e 0f 1f 84 00 00 nopw %cs:0x0(%rax,%rax,1)

5ed: 00 00 00

00000000000005f0 <register\_tm\_clones>:

5f0: 48 8d 3d 19 0a 20 00 lea 0x200a19(%rip),%rdi # 201010 <\_\_TMC\_END\_\_>

5f7: 48 8d 35 12 0a 20 00 lea 0x200a12(%rip),%rsi # 201010 <\_\_TMC\_END\_\_>

5fe: 55 push %rbp

5ff: 48 29 fe sub %rdi,%rsi

602: 48 89 e5 mov %rsp,%rbp

605: 48 c1 fe 03 sar $0x3,%rsi

609: 48 89 f0 mov %rsi,%rax

60c: 48 c1 e8 3f shr $0x3f,%rax

610: 48 01 c6 add %rax,%rsi

613: 48 d1 fe sar %rsi

616: 74 18 je 630 <register\_tm\_clones+0x40>

618: 48 8b 05 d1 09 20 00 mov 0x2009d1(%rip),%rax # 200ff0 <\_ITM\_registerTMCloneTable>

61f: 48 85 c0 test %rax,%rax

622: 74 0c je 630 <register\_tm\_clones+0x40>

624: 5d pop %rbp

625: ff e0 jmpq \*%rax

627: 66 0f 1f 84 00 00 00 nopw 0x0(%rax,%rax,1)

62e: 00 00

630: 5d pop %rbp

631: c3 retq

632: 0f 1f 40 00 nopl 0x0(%rax)

636: 66 2e 0f 1f 84 00 00 nopw %cs:0x0(%rax,%rax,1)

63d: 00 00 00

0000000000000640 <\_\_do\_global\_dtors\_aux>:

640: 80 3d c9 09 20 00 00 cmpb $0x0,0x2009c9(%rip) # 201010 <\_\_TMC\_END\_\_>

647: 75 2f jne 678 <\_\_do\_global\_dtors\_aux+0x38>

649: 48 83 3d a7 09 20 00 cmpq $0x0,0x2009a7(%rip) # 200ff8 <\_\_cxa\_finalize@GLIBC\_2.2.5>

650: 00

651: 55 push %rbp

652: 48 89 e5 mov %rsp,%rbp

655: 74 0c je 663 <\_\_do\_global\_dtors\_aux+0x23>

657: 48 8b 3d aa 09 20 00 mov 0x2009aa(%rip),%rdi # 201008 <\_\_dso\_handle>

65e: e8 0d ff ff ff callq 570 <\_\_cxa\_finalize@plt>

663: e8 48 ff ff ff callq 5b0 <deregister\_tm\_clones>

668: c6 05 a1 09 20 00 01 movb $0x1,0x2009a1(%rip) # 201010 <\_\_TMC\_END\_\_>

66f: 5d pop %rbp

670: c3 retq

671: 0f 1f 80 00 00 00 00 nopl 0x0(%rax)

678: f3 c3 repz retq

67a: 66 0f 1f 44 00 00 nopw 0x0(%rax,%rax,1)

0000000000000680 <frame\_dummy>:

680: 55 push %rbp

681: 48 89 e5 mov %rsp,%rbp

684: 5d pop %rbp

685: e9 66 ff ff ff jmpq 5f0 <register\_tm\_clones>

000000000000068a <main>:

68a: 55 push %rbp // push frame base reg to stack

68b: 48 89 e5 mov %rsp,%rbp // copy base pointer reg to stack pointer reg

68e: 48 83 ec 20 sub $0x20,%rsp // subtract 20 from stack pointer

692: 89 7d ec mov %edi,-0x14(%rbp)

695: 48 89 75 e0 mov %rsi,-0x20(%rbp)

699: 48 8b 45 e0 mov -0x20(%rbp),%rax

69d: 48 83 c0 08 add $0x8,%rax

6a1: 48 8b 00 mov (%rax),%rax

6a4: 48 89 c7 mov %rax,%rdi

6a7: e8 b4 fe ff ff callq 560 <atoi@plt>

6ac: 89 45 fc mov %eax,-0x4(%rbp)

6af: 8b 45 fc mov -0x4(%rbp),%eax

6b2: 89 c7 mov %eax,%edi

6b4: e8 1f 00 00 00 callq 6d8 <custom>

6b9: 89 c2 mov %eax,%edx

6bb: 8b 45 fc mov -0x4(%rbp),%eax

6be: 89 c6 mov %eax,%esi

6c0: 48 8d 3d dd 00 00 00 lea 0xdd(%rip),%rdi # 7a4 <\_IO\_stdin\_used+0x4>

6c7: b8 00 00 00 00 mov $0x0,%eax

6cc: e8 7f fe ff ff callq 550 <printf@plt>

6d1: b8 00 00 00 00 mov $0x0,%eax

6d6: c9 leaveq

6d7: c3 retq

00000000000006d8 <custom>:

6d8: 55 push %rbp

6d9: 48 89 e5 mov %rsp,%rbp

6dc: 89 7d ec mov %edi,-0x14(%rbp)

6df: c7 45 f4 01 00 00 00 movl $0x1,-0xc(%rbp)

6e6: c7 45 fc 05 00 00 00 movl $0x5,-0x4(%rbp)

6ed: c7 45 f8 00 00 00 00 movl $0x0,-0x8(%rbp)

6f4: eb 0e jmp 704 <custom+0x2c>

6f6: 8b 45 f4 mov -0xc(%rbp),%eax

6f9: 0f af 45 ec imul -0x14(%rbp),%eax

6fd: 89 45 f4 mov %eax,-0xc(%rbp)

700: 83 45 f8 01 addl $0x1,-0x8(%rbp)

704: 8b 45 f8 mov -0x8(%rbp),%eax

707: 3b 45 fc cmp -0x4(%rbp),%eax

70a: 7e ea jle 6f6 <custom+0x1e>

70c: 8b 45 f4 mov -0xc(%rbp),%eax

70f: 5d pop %rbp

710: c3 retq

711: 66 2e 0f 1f 84 00 00 nopw %cs:0x0(%rax,%rax,1)

718: 00 00 00

71b: 0f 1f 44 00 00 nopl 0x0(%rax,%rax,1)

0000000000000720 <\_\_libc\_csu\_init>:

720: 41 57 push %r15

722: 41 56 push %r14

724: 49 89 d7 mov %rdx,%r15

727: 41 55 push %r13

729: 41 54 push %r12

72b: 4c 8d 25 7e 06 20 00 lea 0x20067e(%rip),%r12 # 200db0 <\_\_frame\_dummy\_init\_array\_entry>

732: 55 push %rbp

733: 48 8d 2d 7e 06 20 00 lea 0x20067e(%rip),%rbp # 200db8 <\_\_init\_array\_end>

73a: 53 push %rbx

73b: 41 89 fd mov %edi,%r13d

73e: 49 89 f6 mov %rsi,%r14

741: 4c 29 e5 sub %r12,%rbp

744: 48 83 ec 08 sub $0x8,%rsp

748: 48 c1 fd 03 sar $0x3,%rbp

74c: e8 d7 fd ff ff callq 528 <\_init>

751: 48 85 ed test %rbp,%rbp

754: 74 20 je 776 <\_\_libc\_csu\_init+0x56>

756: 31 db xor %ebx,%ebx

758: 0f 1f 84 00 00 00 00 nopl 0x0(%rax,%rax,1)

75f: 00

760: 4c 89 fa mov %r15,%rdx

763: 4c 89 f6 mov %r14,%rsi

766: 44 89 ef mov %r13d,%edi

769: 41 ff 14 dc callq \*(%r12,%rbx,8)

76d: 48 83 c3 01 add $0x1,%rbx

771: 48 39 dd cmp %rbx,%rbp

774: 75 ea jne 760 <\_\_libc\_csu\_init+0x40>

776: 48 83 c4 08 add $0x8,%rsp

77a: 5b pop %rbx

77b: 5d pop %rbp

77c: 41 5c pop %r12

77e: 41 5d pop %r13

780: 41 5e pop %r14

782: 41 5f pop %r15

784: c3 retq

785: 90 nop

786: 66 2e 0f 1f 84 00 00 nopw %cs:0x0(%rax,%rax,1)

78d: 00 00 00

0000000000000790 <\_\_libc\_csu\_fini>:

790: f3 c3 repz retq

Disassembly of section .fini:

0000000000000794 <\_fini>:

794: 48 83 ec 08 sub $0x8,%rsp

798: 48 83 c4 08 add $0x8,%rsp

79c: c3 retq