Paul Brodhead

Lab 1

C Code:

|  |
| --- |
| #include<stdio.h>  #define NUMPRINT 5  /\* function to print string to console \*/  void helloWorld(){  char output[] = "Hello World!";  printf("%s \n", output);  }  /\* main function designed to call helloWorld 5 times \*/  int main(void){  for(int i = 0; i < NUMPRINT; i++){  helloWorld();  }  } |

Assembly Code:

|  |
| --- |
| .file "Lab1.c"  .text  .section .rodata  .LC0:  .string "%s \n"  .text  .globl helloWorld  .type helloWorld, @function  helloWorld:  .LFB0:  .cfi\_startproc  pushq %rbp //push frame base to stack  .cfi\_def\_cfa\_offset 16  .cfi\_offset 6, -16  movq %rsp, %rbp //change the frame base to the stack pointer  .cfi\_def\_cfa\_register 6  subq $32, %rsp //subtract 32 from stack pointer  movq %fs:40, %rax //change value of rax to value at mem addr 40  movq %rax, -8(%rbp) //change value of mem at addr (rbp-8) to rax  xorl %eax, %eax //XOR eax with itself  movabsq $8022916924116329800, %rax //set rax to massive constant  movq %rax, -21(%rbp) //change value of mem at addr (rbp-21) to rax  movl $560229490, -13(%rbp) //change value of mem at addr (rbp-13) to constant  movb $0, -9(%rbp) //change value of mem at addr (rbp-9) to 0  leaq -21(%rbp), %rax //change value of rax to (rbp-21)  movq %rax, %rsi //change value of rsi to rax  leaq .LC0(%rip), %rdi //change value of rdi to rip  movl $0, %eax //change value of eax to 0  call printf@PLT //part of stdio.h  nop //skip instruction  movq -8(%rbp), %rax //change value of rax to mem at addr (rbp-8)  xorq %fs:40, %rax //rax gets rax XOR mem at addr 40  je .L2 //jump if zero to label L2  call \_\_stack\_chk\_fail@PLT  .L2:  leave  .cfi\_def\_cfa 7, 8  ret  .cfi\_endproc  .LFE0:  .size helloWorld, .-helloWorld  .globl main  .type main, @function  main:  .LFB1:  .cfi\_startproc  pushq %rbp //push frame base to stack  .cfi\_def\_cfa\_offset 16  .cfi\_offset 6, -16  movq %rsp, %rbp //change rbp to rsp  .cfi\_def\_cfa\_register 6  subq $16, %rsp //subtract 16 from rsp  movl $0, -4(%rbp) //change value of mem at addr (rbp-40) to 0  jmp .L4 //unconditional jump to label L4  .L5:  movl $0, %eax //change eax to 0  call helloWorld //run helloWorld subroutine  addl $1, -4(%rbp) //add 1 to mem at addr (rbp-4)  .L4:  cmpl $4, -4(%rbp) //compare mem at addr (rbp-4) with 4  jle .L5 //jump if <= to label L5  movl $0, %eax //change eax to 0  leave  .cfi\_def\_cfa 7, 8  ret  .cfi\_endproc  .LFE1:  .size main, .-main  .ident "GCC: (Ubuntu 8.2.0-7ubuntu1) 8.2.0"  .section .note.GNU-stack,"",@progbits |

Objective Disassembly

|  |
| --- |
| 0000000000001000 <\_init>:  1000: 48 83 ec 08 sub $0x8,%rsp  1004: 48 8b 05 dd 2f 00 00 mov 0x2fdd(%rip),%rax # 3fe8 <\_\_gmon\_start\_\_>  100b: 48 85 c0 test %rax,%rax  100e: 74 02 je 1012 <\_init+0x12>  1010: ff d0 callq \*%rax  1012: 48 83 c4 08 add $0x8,%rsp  1016: c3 retq  Disassembly of section .plt:  0000000000001020 <.plt>:  1020: ff 35 92 2f 00 00 pushq 0x2f92(%rip) # 3fb8 <\_GLOBAL\_OFFSET\_TABLE\_+0x8>  1026: ff 25 94 2f 00 00 jmpq \*0x2f94(%rip) # 3fc0 <\_GLOBAL\_OFFSET\_TABLE\_+0x10>  102c: 0f 1f 40 00 nopl 0x0(%rax)  0000000000001030 <\_\_stack\_chk\_fail@plt>:  1030: ff 25 92 2f 00 00 jmpq \*0x2f92(%rip) # 3fc8 <\_\_stack\_chk\_fail@GLIBC\_2.4>  1036: 68 00 00 00 00 pushq $0x0  103b: e9 e0 ff ff ff jmpq 1020 <.plt>  0000000000001040 <printf@plt>:  1040: ff 25 8a 2f 00 00 jmpq \*0x2f8a(%rip) # 3fd0 <printf@GLIBC\_2.2.5>  1046: 68 01 00 00 00 pushq $0x1  104b: e9 d0 ff ff ff jmpq 1020 <.plt>  Disassembly of section .plt.got:  0000000000001050 <\_\_cxa\_finalize@plt>:  1050: ff 25 a2 2f 00 00 jmpq \*0x2fa2(%rip) # 3ff8 <\_\_cxa\_finalize@GLIBC\_2.2.5>  1056: 66 90 xchg %ax,%ax  Disassembly of section .text:  0000000000001060 <\_start>:  1060: 31 ed xor %ebp,%ebp  1062: 49 89 d1 mov %rdx,%r9  1065: 5e pop %rsi  1066: 48 89 e2 mov %rsp,%rdx  1069: 48 83 e4 f0 and $0xfffffffffffffff0,%rsp  106d: 50 push %rax  106e: 54 push %rsp  106f: 4c 8d 05 ba 01 00 00 lea 0x1ba(%rip),%r8 # 1230 <\_\_libc\_csu\_fini>  1076: 48 8d 0d 53 01 00 00 lea 0x153(%rip),%rcx # 11d0 <\_\_libc\_csu\_init>  107d: 48 8d 3d 20 01 00 00 lea 0x120(%rip),%rdi # 11a4 <main>  1084: ff 15 56 2f 00 00 callq \*0x2f56(%rip) # 3fe0 <\_\_libc\_start\_main@GLIBC\_2.2.5>  108a: f4 hlt  108b: 0f 1f 44 00 00 nopl 0x0(%rax,%rax,1)  0000000000001090 <deregister\_tm\_clones>:  1090: 48 8d 3d 79 2f 00 00 lea 0x2f79(%rip),%rdi # 4010 <\_\_TMC\_END\_\_>  1097: 48 8d 05 72 2f 00 00 lea 0x2f72(%rip),%rax # 4010 <\_\_TMC\_END\_\_>  109e: 48 39 f8 cmp %rdi,%rax  10a1: 74 15 je 10b8 <deregister\_tm\_clones+0x28>  10a3: 48 8b 05 2e 2f 00 00 mov 0x2f2e(%rip),%rax # 3fd8 <\_ITM\_deregisterTMCloneTable>  10aa: 48 85 c0 test %rax,%rax  10ad: 74 09 je 10b8 <deregister\_tm\_clones+0x28>  10af: ff e0 jmpq \*%rax  10b1: 0f 1f 80 00 00 00 00 nopl 0x0(%rax)  10b8: c3 retq  10b9: 0f 1f 80 00 00 00 00 nopl 0x0(%rax)  00000000000010c0 <register\_tm\_clones>:  10c0: 48 8d 3d 49 2f 00 00 lea 0x2f49(%rip),%rdi # 4010 <\_\_TMC\_END\_\_>  10c7: 48 8d 35 42 2f 00 00 lea 0x2f42(%rip),%rsi # 4010 <\_\_TMC\_END\_\_>  10ce: 48 29 fe sub %rdi,%rsi  10d1: 48 c1 fe 03 sar $0x3,%rsi  10d5: 48 89 f0 mov %rsi,%rax  10d8: 48 c1 e8 3f shr $0x3f,%rax  10dc: 48 01 c6 add %rax,%rsi  10df: 48 d1 fe sar %rsi  10e2: 74 14 je 10f8 <register\_tm\_clones+0x38>  10e4: 48 8b 05 05 2f 00 00 mov 0x2f05(%rip),%rax # 3ff0 <\_ITM\_registerTMCloneTable>  10eb: 48 85 c0 test %rax,%rax  10ee: 74 08 je 10f8 <register\_tm\_clones+0x38>  10f0: ff e0 jmpq \*%rax  10f2: 66 0f 1f 44 00 00 nopw 0x0(%rax,%rax,1)  10f8: c3 retq  10f9: 0f 1f 80 00 00 00 00 nopl 0x0(%rax)  0000000000001100 <\_\_do\_global\_dtors\_aux>:  1100: 80 3d 09 2f 00 00 00 cmpb $0x0,0x2f09(%rip) # 4010 <\_\_TMC\_END\_\_>  1107: 75 2f jne 1138 <\_\_do\_global\_dtors\_aux+0x38>  1109: 55 push %rbp  110a: 48 83 3d e6 2e 00 00 cmpq $0x0,0x2ee6(%rip) # 3ff8 <\_\_cxa\_finalize@GLIBC\_2.2.5>  1111: 00  1112: 48 89 e5 mov %rsp,%rbp  1115: 74 0c je 1123 <\_\_do\_global\_dtors\_aux+0x23>  1117: 48 8b 3d ea 2e 00 00 mov 0x2eea(%rip),%rdi # 4008 <\_\_dso\_handle>  111e: e8 2d ff ff ff callq 1050 <\_\_cxa\_finalize@plt>  1123: e8 68 ff ff ff callq 1090 <deregister\_tm\_clones>  1128: c6 05 e1 2e 00 00 01 movb $0x1,0x2ee1(%rip) # 4010 <\_\_TMC\_END\_\_>  112f: 5d pop %rbp  1130: c3 retq  1131: 0f 1f 80 00 00 00 00 nopl 0x0(%rax)  1138: c3 retq  1139: 0f 1f 80 00 00 00 00 nopl 0x0(%rax)  0000000000001140 <frame\_dummy>:  1140: e9 7b ff ff ff jmpq 10c0 <register\_tm\_clones>  0000000000001145 <helloWorld>:  1145: 55 push %rbp  1146: 48 89 e5 mov %rsp,%rbp  1149: 48 83 ec 20 sub $0x20,%rsp  114d: 64 48 8b 04 25 28 00 mov %fs:0x28,%rax  1154: 00 00  1156: 48 89 45 f8 mov %rax,-0x8(%rbp)  115a: 31 c0 xor %eax,%eax  115c: 48 b8 48 65 6c 6c 6f movabs $0x6f57206f6c6c6548,%rax  1163: 20 57 6f  1166: 48 89 45 eb mov %rax,-0x15(%rbp)  116a: c7 45 f3 72 6c 64 21 movl $0x21646c72,-0xd(%rbp)  1171: c6 45 f7 00 movb $0x0,-0x9(%rbp)  1175: 48 8d 45 eb lea -0x15(%rbp),%rax  1179: 48 89 c6 mov %rax,%rsi  117c: 48 8d 3d 81 0e 00 00 lea 0xe81(%rip),%rdi # 2004 <\_IO\_stdin\_used+0x4>  1183: b8 00 00 00 00 mov $0x0,%eax  1188: e8 b3 fe ff ff callq 1040 <printf@plt>  118d: 90 nop  118e: 48 8b 45 f8 mov -0x8(%rbp),%rax  1192: 64 48 33 04 25 28 00 xor %fs:0x28,%rax  1199: 00 00  119b: 74 05 je 11a2 <helloWorld+0x5d>  119d: e8 8e fe ff ff callq 1030 <\_\_stack\_chk\_fail@plt>  11a2: c9 leaveq  11a3: c3 retq  00000000000011a4 <main>:  11a4: 55 push %rbp  11a5: 48 89 e5 mov %rsp,%rbp  11a8: 48 83 ec 10 sub $0x10,%rsp  11ac: c7 45 fc 00 00 00 00 movl $0x0,-0x4(%rbp)  11b3: eb 0e jmp 11c3 <main+0x1f>  11b5: b8 00 00 00 00 mov $0x0,%eax  11ba: e8 86 ff ff ff callq 1145 <helloWorld>  11bf: 83 45 fc 01 addl $0x1,-0x4(%rbp)  11c3: 83 7d fc 04 cmpl $0x4,-0x4(%rbp)  11c7: 7e ec jle 11b5 <main+0x11>  11c9: b8 00 00 00 00 mov $0x0,%eax  11ce: c9 leaveq  11cf: c3 retq  00000000000011d0 <\_\_libc\_csu\_init>:  11d0: 41 57 push %r15  11d2: 49 89 d7 mov %rdx,%r15  11d5: 41 56 push %r14  11d7: 49 89 f6 mov %rsi,%r14  11da: 41 55 push %r13  11dc: 41 89 fd mov %edi,%r13d  11df: 41 54 push %r12  11e1: 4c 8d 25 c8 2b 00 00 lea 0x2bc8(%rip),%r12 # 3db0 <\_\_frame\_dummy\_init\_array\_entry>  11e8: 55 push %rbp  11e9: 48 8d 2d c8 2b 00 00 lea 0x2bc8(%rip),%rbp # 3db8 <\_\_init\_array\_end>  11f0: 53 push %rbx  11f1: 4c 29 e5 sub %r12,%rbp  11f4: 48 83 ec 08 sub $0x8,%rsp  11f8: e8 03 fe ff ff callq 1000 <\_init>  11fd: 48 c1 fd 03 sar $0x3,%rbp  1201: 74 1b je 121e <\_\_libc\_csu\_init+0x4e>  1203: 31 db xor %ebx,%ebx  1205: 0f 1f 00 nopl (%rax)  1208: 4c 89 fa mov %r15,%rdx  120b: 4c 89 f6 mov %r14,%rsi  120e: 44 89 ef mov %r13d,%edi  1211: 41 ff 14 dc callq \*(%r12,%rbx,8)  1215: 48 83 c3 01 add $0x1,%rbx  1219: 48 39 dd cmp %rbx,%rbp  121c: 75 ea jne 1208 <\_\_libc\_csu\_init+0x38>  121e: 48 83 c4 08 add $0x8,%rsp  1222: 5b pop %rbx  1223: 5d pop %rbp  1224: 41 5c pop %r12  1226: 41 5d pop %r13  1228: 41 5e pop %r14  122a: 41 5f pop %r15  122c: c3 retq  122d: 0f 1f 00 nopl (%rax)  0000000000001230 <\_\_libc\_csu\_fini>:  1230: c3 retq  Disassembly of section .fini:  0000000000001234 <\_fini>:  1234: 48 83 ec 08 sub $0x8,%rsp  1238: 48 83 c4 08 add $0x8,%rsp  123c: c3 retq |