

## DEBUG assembly

- Learn how to debug in GDB easily ☺

### 1. Run GDB

1. bash> gdb ./<file to debug >

### 2. Adding BreakPoints

2. (gdb) break <assembly fn name>

- or

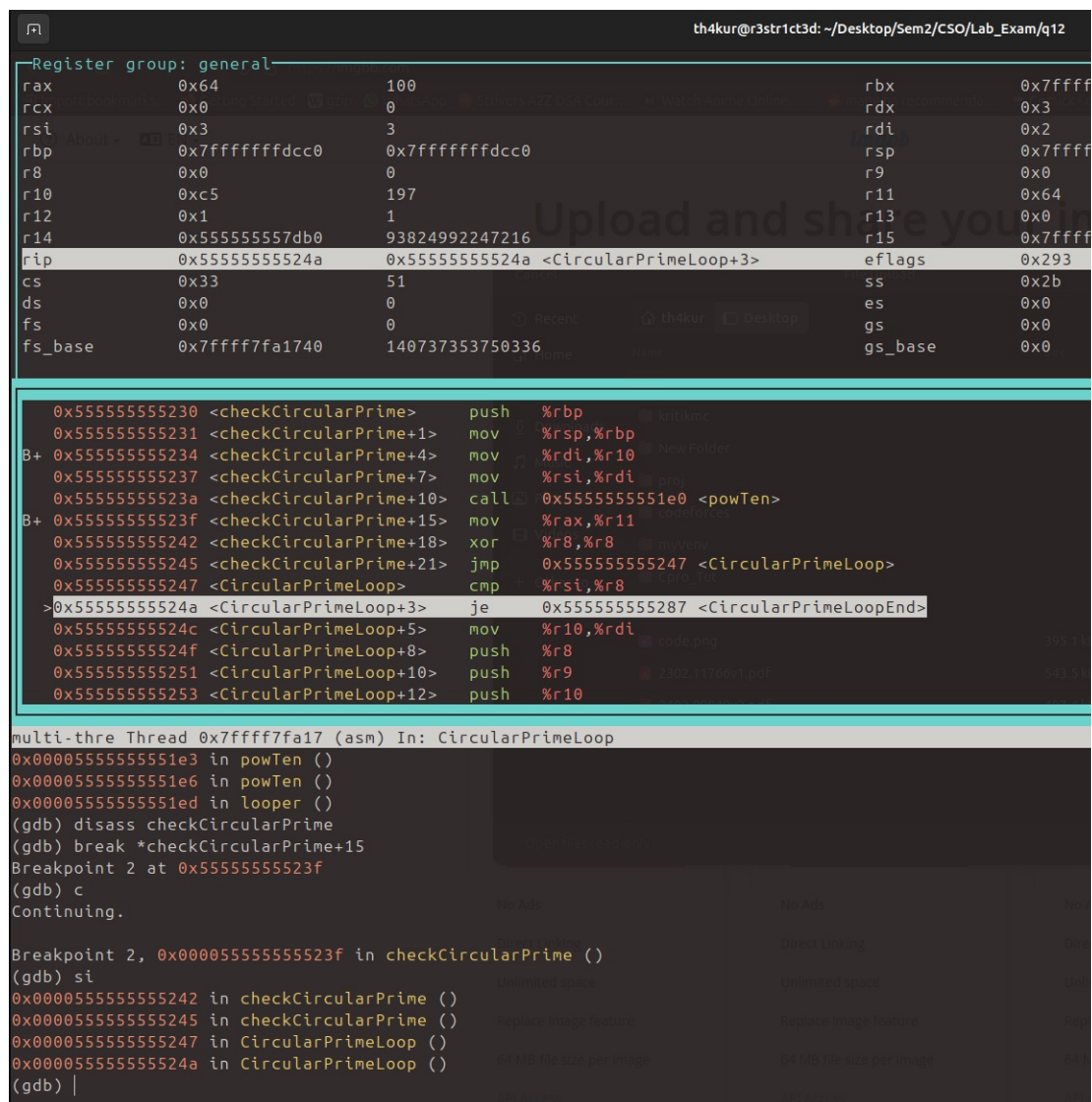
3. (gdb) break \*<assembly\_fn\_name>+offset

### 2. run/re-run the program

9. (gdb) r

## DEBUG UI

- inorder to show executing instructions and all registers at each step type these commands in order:
- highlights current executing line
- highlights all changed register(s)



```
th4kur@r3str1ct3d: ~/Desktop/Sem2/CSO/Lab_Exam/q12

Register group: general
rax      0x64      100      rbx      0x7fffff
rcx      0x0      0      rdx      0x3
rsi      0x3      3      rdi      0x2
rbp      0x7fffffddcc0  0x7fffffddcc0  rsp      0x7fffff
r8       0x0      0      r9       0x0
r10      0xc5     197     r11      0x64
r12      0x1      1      r13      0x0
r14      0x55555557db0  93824992247216 r15      0x7fffff
rip      0x5555555524a  0x5555555524a  <CircularPrimeLoop+3>  eflags    0x293
cs       0x33     51      ss       0x2b
ds       0x0      0      es       0x0
fs       0x0      0      gs       0x0
fs_base  0x7ffff7fa1740  140737353750336 gs_base  0x0

0x55555555230 <checkCircularPrime> push    %rbp
0x55555555231 <checkCircularPrime+1> mov     %rsp,%rbp
B+ 0x55555555234 <checkCircularPrime+4> mov     %rdi,%r10
0x55555555237 <checkCircularPrime+7> mov     %rsi,%rdi
0x5555555523a <checkCircularPrime+10> call    0x555555551e0 <powTen>
B+ 0x5555555523f <checkCircularPrime+15> mov     %rax,%r11
0x55555555242 <checkCircularPrime+18> xor     %r8,%r8
0x55555555245 <checkCircularPrime+21> jmp     0x55555555247 <CircularPrimeLoop>
0x55555555247 <CircularPrimeLoop> cmp     %rsi,%r8
> 0x5555555524a <CircularPrimeLoop+3> je      0x55555555287 <CircularPrimeLoopEnd>
0x5555555524c <CircularPrimeLoop+5> mov     %r10,%rdi
0x5555555524f <CircularPrimeLoop+8> push    %r8
0x55555555251 <CircularPrimeLoop+10> push    %r9
0x55555555253 <CircularPrimeLoop+12> push    %r10

multi-thre Thread 0x7ffff7fa17 (asm) In: CircularPrimeLoop
0x0000555555551e3 in powTen ()
0x0000555555551e6 in powTen ()
0x0000555555551ed in looper ()
(gdb) disass checkCircularPrime
(gdb) break *checkCircularPrime+15
Breakpoint 2 at 0x5555555523f
(gdb) c
Continuing.

Breakpoint 2, 0x00005555555523f in checkCircularPrime ()
(gdb) si
0x000055555555242 in checkCircularPrime ()
0x000055555555245 in checkCircularPrime ()
0x000055555555247 in CircularPrimeLoop ()
0x00005555555524a in CircularPrimeLoop ()
(gdb) |
```

- add breakpoints (preferably after input)
- run the program & give inputs
- program halts at breakpoint
- Now execute below instructions inorder

4. (gdb) layout reg

5. (gdb) layout asm

- Then update the ui by going to next instruction

6. (gdb) si

**One instruction at a time**

7. (gdb) si

**continue till next breakpoint**

8. (gdb) c

- by [e.](#)