Refo Yudhanto

CS360

LAB 1 pre-lab

Part 1:

A.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a.out | text | data | bss |
| T1.c | 15500 | 1547 | 296 | 8 |
| T2.c | 15504 | 1547 | 300 | 4 |
| T3.c | 15500 | 1547 | 296 | 40032 |
| T4.c | 55536 | 1547 | 40320 | 4 |
| T5.c | 15500 | 1547 | 296 | 8 |
| T6.c | 15592 | 1563 | 296 | 40068 |

1. DATA = Initialized globals

BSS = Uninitialized globals and Static locals

2. Text and data is in a.out. Bss is not included in the a.out, only its size is recorded in the header. It is not included as it is an uninitialized data and it runs on the RAM.

B.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a.out | text | data | bss |
| T1.c | 670800 | 585340 | 11320 | 600036 |
| T2.c | 670800 | 585340 | 11320 | 600036 |
| T3.c | 670800 | 585340 | 11320 | 640036 |
| T4.c | 710832 | 585340 | 51352 | 640068 |
| T5.c | 670800 | 585340 | 11320 | 600036 |
| T6.c | 670892 | 585356 | 11320 | 640068 |

Larger size, when using static linking the linker uses all the needed library function code and data into a.out, making it larger.

Part2:

Outfile

enter main

&argc=0xfffa7260 argv=0xfffa72f4 env=0xfffa7308

&a=0xfffa7228 &b=0xfffa7224 &c=0xfffa7220

argc=4

argv[0]=./lab1

argv[1]=one

argv[2]=two

argv[3]=three

enter A

&d=0xfffa71fc &e=0xfffa71f8 &f=0xfffa71f4

enter B

&g=0xfffa71cc &h=0xfffa71c8 &i=0xfffa71c4

enter C

&u=0xfffa7194 &v=0xfffa7190 &w=0xfffa718c &i=0xfffa7188 &p=0xfffa7184

print stack frame link list

0xfffa71a8 ->

0xfffa71d8 ->

0xfffa7208 ->

0xfffa7248 ->

0

print stack contents

0xfffa717c -> 5663a4a7

0xfffa7180 -> fffa7260

0xfffa7184 -> f7eea000

0xfffa7188 -> d

0xfffa718c -> c

0xfffa7190 -> b

0xfffa7194 -> a

0xfffa7198 -> fffffffc

0xfffa719c -> f7eea020

0xfffa71a0 -> fffa7260

0xfffa71a4 -> 5663d000

0xfffa71a8 -> fffa71d8

0xfffa71ac -> 5663a3a8

0xfffa71b0 -> 7

0xfffa71b4 -> 8

0xfffa71b8 -> fffa71c8

0xfffa71bc -> fffa71c4

0xfffa71c0 -> f7eead80

0xfffa71c4 -> 9

0xfffa71c8 -> 8

0xfffa71cc -> 7

0xfffa71d0 -> fffa7260

0xfffa71d4 -> 5663d000

0xfffa71d8 -> fffa7208

0xfffa71dc -> 5663a329

0xfffa71e0 -> 4

0xfffa71e4 -> 5

0xfffa71e8 -> fffa71f8

0xfffa71ec -> fffa71f4

0xfffa71f0 -> fffa7248

0xfffa71f4 -> 6

0xfffa71f8 -> 5

0xfffa71fc -> 4

0xfffa7200 -> fffa7260

0xfffa7204 -> 5663d000

0xfffa7208 -> fffa7248

0xfffa720c -> 5663a2a0

0xfffa7210 -> 1

0xfffa7214 -> 2

0xfffa7218 -> fffa858e

0xfffa721c -> fffa7220

0xfffa7220 -> 3

0xfffa7224 -> 2

0xfffa7228 -> 1

0xfffa722c -> 4

0xfffa7230 -> 4

0xfffa7234 -> fffa72f4

0xfffa7238 -> fffa7308

0xfffa723c -> fffa7260

0xfffa7240 -> 0

0xfffa7244 -> f7eea000

0xfffa7248 -> 0

0xfffa724c -> f7d2ab41

0xfffa7250 -> f7eea000

0xfffa7254 -> f7eea000

0xfffa7258 -> 0

0xfffa725c -> f7d2ab41

0xfffa7260 -> 4

0xfffa7264 -> fffa72f4

0xfffa7268 -> fffa7308

0xfffa726c -> fffa7284

0xfffa7270 -> 1

0xfffa7274 -> 0

0xfffa7278 -> f7eea000

0xfffa727c -> ffffffff

0xfffa7280 -> f7f41000

0xfffa7284 -> 0

0xfffa7288 -> f7eea000

0xfffa728c -> f7eea000

0xfffa7290 -> 0

0xfffa7294 -> bb37ba88

0xfffa7298 -> ea857c98

0xfffa729c -> 0

exit B

exit A

exit main