**\*\*\*\*\*\*\*\*\*\*\*\*PART 1\*\*\*\*\*\*\*\*\*\*\*\*\***

**(A)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case | a.out | TEXT | DATA | BSS |
| (1) | 7236 | 1507 | 304 | 8 |
| (2) | 7240 | 1507 | 308 | 4 |
| (3) | 7236 | 1507 | 304 | 40032 |
| (4) | 47340 | 1710 | 40332 | 4 |
| (5) | 7316 | 1710 | 308 | 8 |
| (6) | 7328 | 1523 | 304 | 40068 |

In terms of the above classification and the variables g, a, b, c, d,

Which variables are in DATA? ­­DATA variables are any initialized global variable, or any initialized static local variable.

Which variables are in BSS? BSS variables are any uninitialized global variable, or any uninitialized static local variable.

In terms of the TEXT, DATA, and BSS sections,

Which sections are in a.out, which section is NOT in a.out? ­TEXT, and DATA are in a.out and BSS is NOT in a.out.

WHY? This is because a.out is only made up of the header, code (text), and data. So BSS is not involved in a.out.

**(B)**

Using cc – m32 -static t.c

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case | a.out | TEXT | DATA | BSS |
| (1) | 657628 | 581882 | 11264 | 3344 |
| (2) | 657628 | 581882 | 11264 | 3344 |
| (3) | 657628 | 581882 | 11264 | 43344 |
| (4) | 697660 | 581930 | 51296 | 3344 |
| (5) | 657628 | 581930 | 11264 | 3344 |
| (6) | 657720 | 581898 | 11264 | 43376 |

WHAT DO YOU SEE? I notice that the a.out file size is much bigger in each of the t.c files when they are statically linked

WHY? This happens because when static linking you use a static library, which means the linker includes all needed library function code into a.out. This makes the a.out file self-contained which makes it much larger than an a.out file that was dynamically linked.

**\*\*\*\*\*\*\*\*\*\*\*\*PART 2\*\*\*\*\*\*\*\*\*\*\*\*\***

enter main

&argc=ffed1620 argv=ffed16b4 env=ffed16c8

&a=ffed15dc &b=ffed15e0 &c=ffed15e4

argc=4

argv[0]=./a.out argv[1]=one argv[2]=two argv[3]=three

enter A

&d=ffed1590 &e=ffed1594 &f=ffed1598

enter B

&g=ffed1560 &h=ffed1564 &i=ffed1568

enter C

&u=ffed1528 &v=ffed152c &w=ffed1530 &i=ffed1534 &p=ffed1538

FP=ffed1548

Stack Frame FP Value = ffed1548

Stack Frame FP Value = ffed1578

Stack Frame FP Value = ffed15a8

Stack Frame FP Value = ffed1608

Stack Frame FP Value = 0

ffed1528 a **🡨 This is variable u | u = 10**

ffed152c b **🡨 This is variable v | v = 11**

ffed1530 c **🡨 This is variable w | w = 12**

ffed1534 3 **🡨 This is variable i | i = 3**

ffed1538 ffed1538 **🡨 This is variable p**

ffed153c e399e900

ffed1540 ffed1620

ffed1544 5662ffcc

ffed1548 ffed1578 **🡨 Stack Frame FP \*FP**

ffed154c 5662e81b

ffed1550 7

ffed1554 8

ffed1558 ffed1564

ffed155c ffed1568

ffed1560 7 **🡨 This is variable g | g = 7**

ffed1564 8 **🡨 This is variable h | h = 8**

ffed1568 9 **🡨 This is variable i | i = 9**

ffed156c e399e900

ffed1570 ffed1620

ffed1574 5662ffcc

ffed1578 ffed15a8 **🡨 Stack Frame FP \*FP**

ffed157c 5662e780

ffed1580 4

ffed1584 5

ffed1588 ffed1594

ffed158c ffed1598

ffed1590 4 **🡨 This is variable d | d = 4**

ffed1594 5 **🡨 This is variable e | e = 5**

ffed1598 6 **🡨 This is variable f | f = 6**

ffed159c e399e900

ffed15a0 ffed1620

ffed15a4 5662ffcc

ffed15a8 ffed1608 **🡨 Stack Frame FP \*FP**

ffed15ac 5662e6db

ffed15b0 1

ffed15b4 2

ffed15b8 ffed32f0

ffed15bc ffed15e4

ffed15c0 9

ffed15c4 ffed32e0

ffed15c8 ffed16c8

ffed15cc ffed16b4

ffed15d0 f7fb9000

ffed15d4 f7fb9000

ffed15d8 0

ffed15dc 1 **🡨 This is variable a | a = 1**

ffed15e0 2 **🡨 This is variable b | b = 2**

ffed15e4 3 **🡨 This is variable c | c = 3**

ffed15e8 4

ffed15ec e399e900

ffed15f0 4

ffed15f4 ffed16b4

ffed15f8 ffed16c8

ffed15fc ffed1620

ffed1600 0

ffed1604 f7fb9000

ffed1608 0 **🡨 Stack Frame FP \*FP**

ffed160c f7dfce91

ffed1610 f7fb9000

ffed1614 f7fb9000

ffed1618 0

ffed161c f7dfce91

ffed1620 4 **🡨 This is argc. argc = 1**

ffed1624 ffed16b4

ffed1628 ffed16c8

ffed162c ffed1644

ffed1630 4

ffed1634 ffed16b4

ffed1638 f7fb9000

ffed163c f7fde77a

ffed1640 ffed16b0

ffed1644 0

ffed1648 f7fb9000

ffed164c 0

ffed1650 0

ffed1654 b8135b86

ffed1658 dda3bd96

ffed165c 0

ffed1660 0

ffed1664 0

ffed1668 40

ffed166c f7ff6024

ffed1670 0

ffed1674 0

ffed1678 f7fde889

ffed167c 5662ffcc

ffed1680 4

ffed1684 5662e4a0

ffed1688 0

ffed168c 5662e4d1

ffed1690 5662e5dd

ffed1694 4

ffed1698 ffed16b4

ffed169c 5662e9b0

ffed16a0 5662ea10

ffed16a4 f7fde9d0

ffed16a8 ffed16ac

ffed16ac f7ff6940

ffed16b0 4

ffed16b4 ffed32e0 **🡨 This is argv**

ffed16b8 ffed32e8

ffed16bc ffed32ec

ffed16c0 ffed32f0

ffed16c4 0

ffed16c8 ffed32f6 **🡨 This is env**

ffed16cc ffed38e2

ffed16d0 ffed38f5

ffed16d4 ffed3905

ffed16d8 ffed3927

ffed16dc ffed393f

ffed16e0 ffed394c

ffed16e4 ffed3964

ffed16e8 ffed3987

ffed16ec ffed39a6

ffed16f0 ffed39c3

ffed16f4 ffed39d4

ffed16f8 ffed3a05

ffed16fc ffed3a1c

ffed1700 ffed3a31

ffed1704 ffed3a72

ffed1708 ffed3a82

ffed170c ffed3a96

ffed1710 ffed3a9e

ffed1714 ffed3ab2

ffed1718 ffed3fbe

ffed171c ffed3fc6

ffed1720 ffed3fe6

ffed1724 0

exit B

exit A

exit main