

Brandon Fowler

CSCD340

lab1prob1

a) Does the stack grow up or down? How do you know? Justify your answer.

The stack grows down, because the memory address hex values become smaller in the order that variables and pointers are declared.

b) What version of GCC are you using?

x86_64-linux-gnu-gcc-4.6

c) What version of Linux are you using?

Linux 3.13.0-36-generic x86_64

d) What is odd about how memory is arranged compared to the declarations?

The memory addresses are printed in a different order than they are declared.

e) Run the program twice and each time construct a memory map.

Memory Map 1

x	val	dptr	array[5]	array	val2
12	4.9567	0x7ffc79002b0	60	10	3.14159
0x7ffc79002bc	0x7ffc79002b0	0x7ffc79002a8	0x7ffc79002a4	0x7ffc7900290	0x7ffc7900288
y	dptr2	ptr2	word	ptr	
99	0x7ffc7900288	0x7ffc7900284	0x169f010	0x7ffc79002bc	
0x7ffc7900284	0x7ffc7900278	0x7ffc7900270	0x7ffc7900268	0x7ffc7900260	

Memory Map 2

x	val	dptr	array[5]	array	val2
12	4.9567	0x7ff9b247740	60	10	3.14159
0x7ff9b24774c	0x7ff9b247740	0x7ff9b247738	0x7ff9b247734	0x7ff9b247720	0x7ff9b247718
y	dptr2	ptr2	word	ptr	
99	0x7ff9b247718	0x7ff9b247714	0x1d8d010	0x7ff9b24774c	
0x7ff9b247714	0x7ff9b247708	0x7ff9b247700	0x7ff9b2476f8	0x7ff9b2476f0	

f) Did the addresses change between runs? Why or why not? Justify your answer.

Yes the addresses did change. When the program ended the first time, the memory was cleared. The second time the program ran, entirely different memory locations were assigned.

g) How many bytes are allocated by the calloc?

Memory was allocated for 10 chars. Each char has a size of one byte. Therefore, the entire amount of bytes allocated by the calloc was 10 bytes.

h) How many bytes are leaked? Provide the valgrind output below.

The memory for the calloc was never freed, so 10 bytes were leaked.

```
brandonf@cslinux:~/netstorage$ valgrind ./a.out
```

```
==25142== Memcheck, a memory error detector
```

```
==25142== Copyright (C) 2002-2011, and GNU GPL'd, by Julian Seward et al.
```

```
==25142== Using Valgrind-3.7.0 and LibVEX; rerun with -h for copyright info
```

```
==25142== Command: ./a.out
```

```
==25142==
```

```
x: 0x7ff00042c
```

```
ptr: 0x7ff0003d0
```

```
val: 0x7ff000420
```

```
dptr: 0x7ff000418
```

```
array: 0x7ff000400
```

```
array[5]: 0x7ff000414
```

```
val2: 0x7ff0003f8
```

```
y:0x7ff0003f4
```

```
dptr2: 0x7ff0003e8
```

```
ptr2: 0x7ff0003e0
```

```
word: 0x7ff0003d8
```

```
word: 0x51f1040
```

==25142==

==25142== HEAP SUMMARY:

==25142== in use at exit: 10 bytes in 1 blocks

==25142== total heap usage: 1 allocs, 0 frees, 10 bytes allocated

==25142==

==25142== LEAK SUMMARY:

==25142== definitely lost: 10 bytes in 1 blocks

==25142== indirectly lost: 0 bytes in 0 blocks

==25142== possibly lost: 0 bytes in 0 blocks

==25142== still reachable: 0 bytes in 0 blocks

==25142== suppressed: 0 bytes in 0 blocks

==25142== Rerun with --leak-check=full to see details of leaked memory

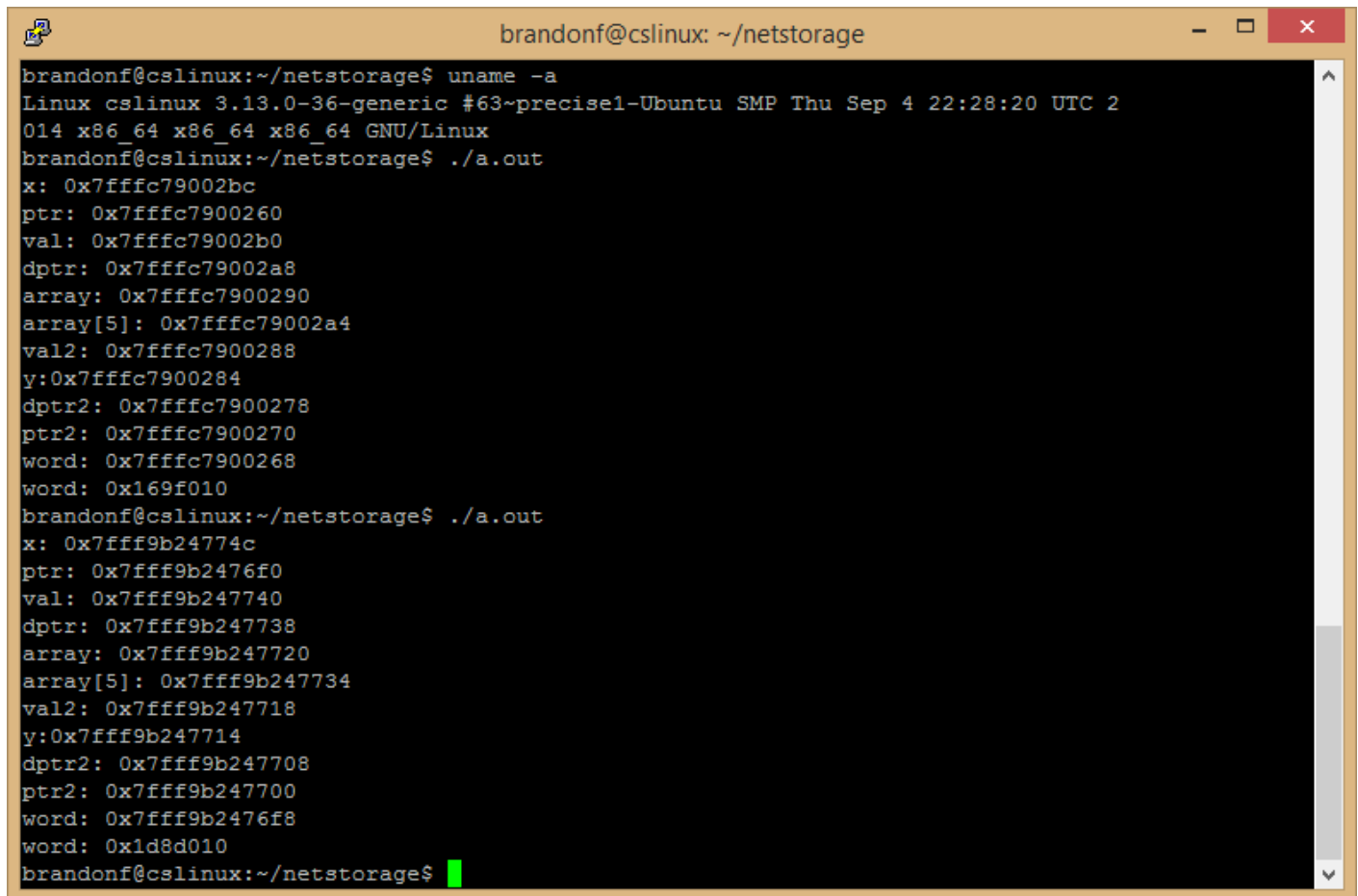
==25142==

==25142== For counts of detected and suppressed errors, rerun with: -v

==25142== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 2 from 2)



Program run screen shot for memory map 1 and memory map 2

A terminal window titled 'brandonf@cslinux: ~/netstorage' with standard window controls. The terminal shows the output of 'uname -a' and two runs of './a.out'. The first run shows memory addresses in the 0x7ffffc79002 range, and the second run shows addresses in the 0x7ffff9b2477 range. The prompt is green in the second run.

```
brandonf@cslinux:~/netstorage$ uname -a
Linux cslinux 3.13.0-36-generic #63~precise1-Ubuntu SMP Thu Sep 4 22:28:20 UTC 2
014 x86_64 x86_64 x86_64 GNU/Linux
brandonf@cslinux:~/netstorage$ ./a.out
x: 0x7ffffc79002bc
ptr: 0x7ffffc7900260
val: 0x7ffffc79002b0
dptr: 0x7ffffc79002a8
array: 0x7ffffc7900290
array[5]: 0x7ffffc79002a4
val2: 0x7ffffc7900288
y:0x7ffffc7900284
dptr2: 0x7ffffc7900278
ptr2: 0x7ffffc7900270
word: 0x7ffffc7900268
word: 0x169f010
brandonf@cslinux:~/netstorage$ ./a.out
x: 0x7ffff9b24774c
ptr: 0x7ffff9b2476f0
val: 0x7ffff9b247740
dptr: 0x7ffff9b247738
array: 0x7ffff9b247720
array[5]: 0x7ffff9b247734
val2: 0x7ffff9b247718
y:0x7ffff9b247714
dptr2: 0x7ffff9b247708
ptr2: 0x7ffff9b247700
word: 0x7ffff9b2476f8
word: 0x1d8d010
brandonf@cslinux:~/netstorage$
```