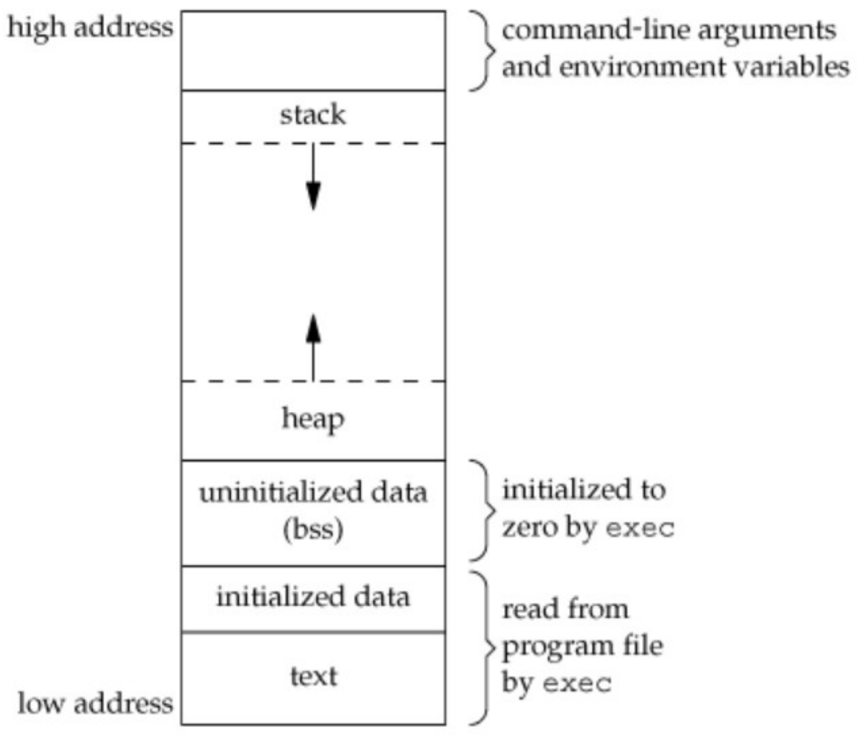
T1a- Addresses - based on usful link

565c7 – bss   565c5- text   56e - Heap

- &addr2: 0xffaaa79c - &addr3: 0xffaaa7a0	Both are stored at high address (ff)- STACK! Distance between addr3 and addr 2 ia 4 bytes (size of int)
- foo: 0x565c5779	Text segment (function)
- &addr5: 0x565c7018	Bss.
dist1: (size_t)&addr6 - (size_t)p: -4	*p = &addr5 both addr5 and addr6 are uninitialized segments, so it is located at bss. This is why dist1 = -4.
dist2: (size_t)&local - (size_t)p: -1454491852	regarding dist2, which is very very big: local is located on the stack, so basically, local seats at high addresses while p (addr5) seats at low addresses. Overflow.
dist3: (size_t)&foo - (size_t)p: -6303	This distance is relativaly small. Foo- text. close to bss.
- addr0: 0x565c7008	Initialized data
- addr1: 0x565c7010	Static. On bss next to addr5,l addr6
- &addr6: 0x565c7014	Bss. With addr5
- yos: 0x565c5970	Text (“ree”)
- addr4: 0x56e83160	Heap. Address of data it points at. (malloc from line 17)
- &addr4: 0xffaaa7a4	pointer’s address. Stack. High address (ff)
- &foo1: 0x565c5867 - &foo1: 0x565c5892	Text. function
- &foo2 - &foo1: 43	Probably size of function
4	My addition- Sizeof(long). Not enough. Long works in two’s completment. Thats why the range is smaller (now we consider also negative numbers). We even received a “warning: format ‘%ld’ expects argument of type ‘long int’, but argument 2 has type ‘int’ [-Wformat=]”


high address
low address
stack
heap
uninitialized data (bss)
initialized data
text
command-line arguments and environment variables
initialized to zero by exec
read from program file by exec

T1b- Distances
?

T1c- Arrays memory layout

```
0xfff898d8 0xfff898dc
0xfff898e4 0xfff898e8
0xfff898f0 0xfff898f8
0xfff89909 0xfff8990a
```

int, float -> 4 bytes
double -> 8 bytes
char -> 1 byte

‘+’ operator is acknowledge to the size of variables at the array.

Memory layout of arrays- sequence of cells in memory. In this case, on the stack.

* As expected, all cells are distanced by 4 (sizeof(int)) bytes from each other.

T1d- Pointers and arrays

P is on stack.