stack ↓

Grows downward as things are added, so as you call more functions etc, you'll be using lower and lower addresses. Things like local variables & function calls live in the stack.

mmap

memory mapped objects, such as libraries

sparsely populated to save room

heap ↑

dynamic memory allocation. If your program is running & needs to map out new memory for an object, that ends up on the heap. Say for example, you call malloc for some dynamic string variable. That lives here, in the heap.

bss
data

Global static data. Initialized & uninitialized. When you write code, some variables have global data, it's stored here. Another possibility here is global buffer data, which might be important later.

text

where your code lives, in text ~~matya~~ form, when you compile it.