RSS is the Resident Set Size and is used to show how much memory is allocated to that process and is in RAM. It does not include memory that is swapped out. It does include memory from shared libraries as long as the pages from those libraries are actually in memory. It does include all stack and heap memory.

VSZ is the Virtual Memory Size. It includes all memory that the process can access, including memory that is swapped out, memory that is allocated, but not used, and memory that is from shared libraries.

So if process A has a 500K binary and is linked to 2500K of shared libraries, has 200K of stack/heap allocations of which 100K is actually in memory (rest is swapped or unused), and it has only actually loaded 1000K of the shared libraries and 400K of its own binary then:

RSS: 400K + 1000K + 100K = 1500K

VSZ: 500K + 2500K + 200K = 3200K

Since part of the memory is shared, many processes may use it, so if you add up all of the RSS values you can easily end up with more space than your system has.

The memory that is allocated also may not be in RSS until it is actually used by the program. So if your program allocated a bunch of memory up front, then uses it over time, you could see RSS going up and VSZ staying the same.

There is also PSS (proportional set size). This is a newer measure which tracks the shared memory as a proportion used by the current process. So if there were two processes using the same shared library from before:

PSS: 400K + (1000K/2) + 100K = 400K + 500K + 100K = 1000K

Threads all share the same address space, so the RSS, VSZ and PSS for each thread is identical to all of the other threads in the process. Use ps or top to view this information in linux/unix.

There is way more to it than this, to learn more check the following references: