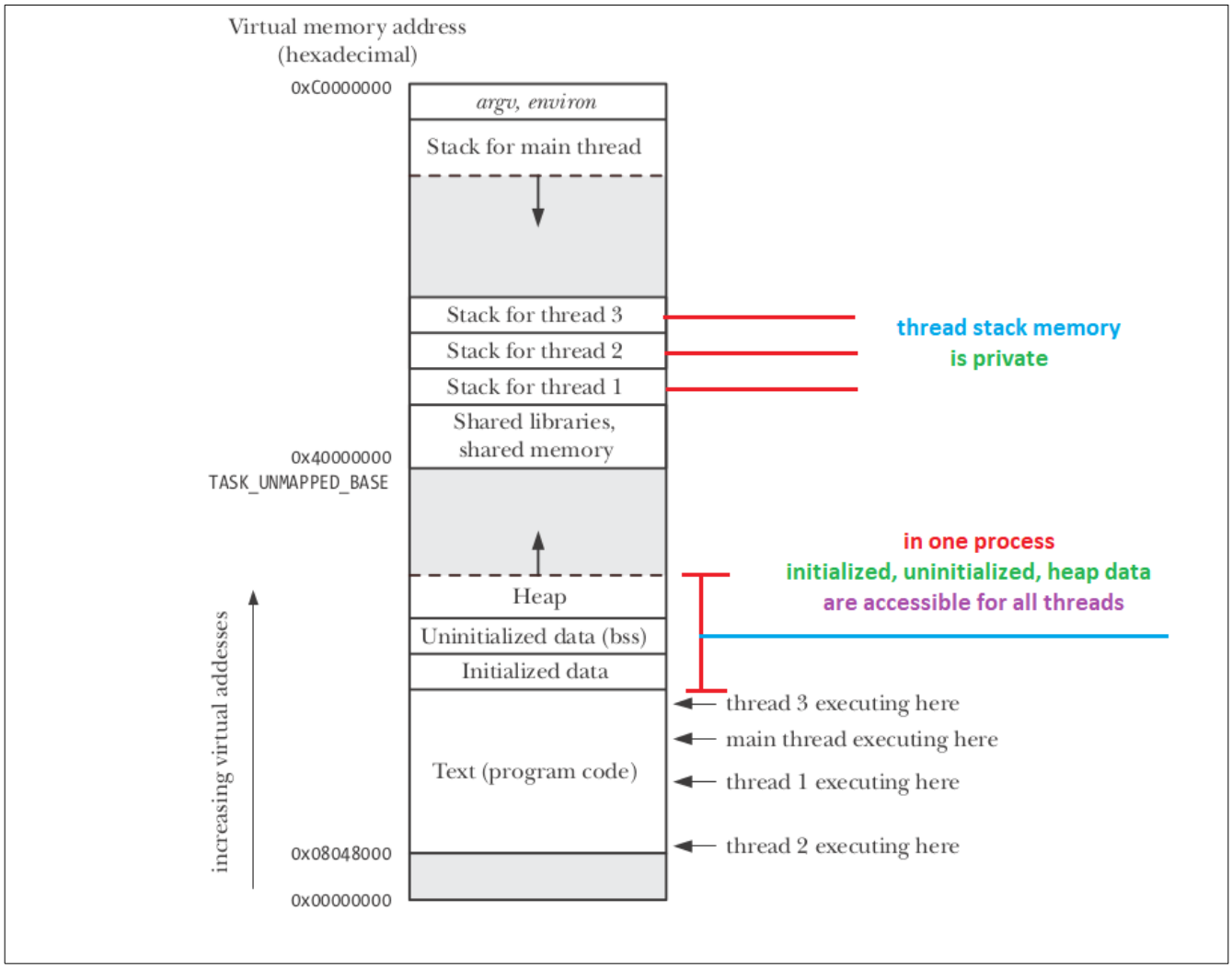
# THREAD

Thread is a mechanism that permits an application to perform multiple tasks concurrently.
A process can contain multiple threads.




**Memory**

***Stack***

Stack is a portion of memory (in RAM). It grows and shinks similar to a stack data struct, things can be added and removed at the top.
When thread is created, it is allocated a stack. If we store more information than stack capacity, we will get a stack overflow and crash.
Default thread stack size for a few architectures

| | | |
|---|---|---|
| i386 | 2 | MB |
| x86_64 | 2 | MB |
| IA-64 | 32 | MB |

| | |
|---|---|
| PowerPC | 4 MB |

However, the stack size can be explicitly set in the attribute argument used to create thread by using pthread_attr_setstacksize.

### *Initialized Data, Uninitalized Data & Heap*

In one process, threads all share the same global memory, including:

| | |
|---|---|
| Initalized data | *global initalized variables*<br>*global static initalized variables*<br>*local static initalized variables* |
| Uninitalized data | *global uninitalized variables*<br>*global static uninitalized variables*<br>*local static uninitalized variables* |
| Heap data | *allocated by new, malloc, calloc* |

**Practise**

Check thread default stack size

| |
|---|
| ulimit -s |