**Assumptions:**

* Level\_4, Level\_3, Level\_2, Level\_1 start from 0x100000, 0x101000, 0x102000, 0x103000 respectively.
* The base address of the page table is 1MB.

**Design approach:**

We already switched to the x86-64 long mode.

We have four loops over the four levels: pml4, pdt, pd, and pte, and for each one, we take the base address and loop over the offset one by one until we reach the maximum number of entries in each page, 512 entries.

For each address, we take it and start to search for the address in each region to know its type, if it is not founded; we start to take the next address and so on until we have no more addresses matching the last physical address. At this point, the program should terminate.

For each loop, Firstly, we take the last mapped page address, add to its 4K, the size of the page, and then, oring with 3, and finally store it in the first entry of the table. Secondly, we take the next address and so on. This is done during every loop.

For checking the memory regions: we take the pointer to the memory region count, and then compare the address with the upper and lower limit of this specific region. If the address locates in the range of the region limits, it is mapped directly, Otherwise, it goes to the next region and start doing the same thing. If the address is not found, it start looping again to fetch the next address.