(1).

1. cpu hardware provides MMU, it takes the virtual address from the program and translate it into a physical address
2. os maintains the MMU's translation tables.
3. cpu will try to find the corresponding physical address in MMU's translation tables.
4. if not found, the cpu will raise an exception and give control to os
5. os will handle the page fault exception by loading the pages from disk or other place

(2).

Feature	**Segmentation**	**Paging**
Size of chunks	Segments can be of any size	divides memory into fixed-sized pages
Management of free space	segment table about each segment's size, location, and access rights managed by the operating system	managed using a page table of page's location, access rights, and status bits
Context switch overhead	Context switches are more expensive with segmentation, because the segment registers need to be saved and restored.	a lower context switch overhead because only the page table needs to be updated
Fragmentation	external fragmentation, where there are gaps between segments that cannot be used	internal fragmentation, where there is unused space within a page
Status Bits and Protection Bits	Valid bit indicates whether the segment is valid, Protection bits can be used to control access to a segment, such as whether it can be read, written, or executed.	Valid bit indicates whether the page is valid, Accessed bit indicates whether the page has been accessed recently. Dirty bit: This bit indicates whether the page has been modified, Protection bits used to control access to the page

(3).

The number of levels of page tables required to map the entire virtual address space is 3.

a-page page table contains 2^13/2^2 page table entries, pointing to 2^11 pages, adding a second level will add another 2^11 pages of page table, adding another level will result in 2^11 * 2^11 * 2^11 *2 ^13 = 2^46.

(4).

1. the page size is 2^12 = 4096 bytes.
   Maximum page table size: 2^20 = 1048576 entries
2. `0xC302C302` is `0b11000011000000101100001100000010`
   1-st level page is `0b1100001100` = 780
   offset is `0b001100000010` = 770
3. `0xEC6666AB` is `0b11101100011001100110011010101011`

   2-st level page is `0b1001100110` = 614
   offset is `0b011010101011` = 1707