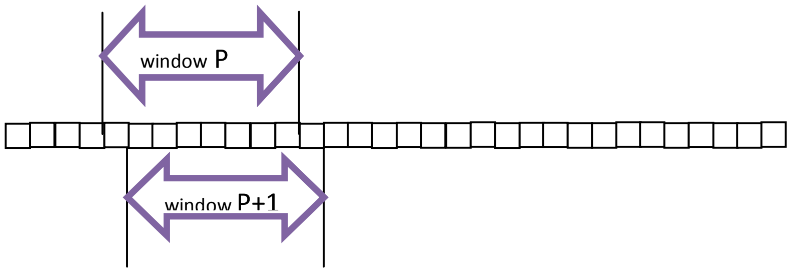
strstrsse42 How does it work?

At first let us define the string matching. String matching is to find out the occurrence of the pattern in the text. The length of the pattern is m and the length of the text is n. Usually the n is greater than m.

1. Let us review how does BOM work.

We call every m(the length of pattern) continuous characters in the text a window.



In window P, we try to find out the longest suffix of windowP which is also a prefix of the pattern. If the suffix is equal to the pattern, then we get a matching. If not, we must move forward in the text. How many characters do we can move forward? One is the most safe distance. But it definitely waist our time since we have got a matching of the prefix of the pattern. Yes, we must move forward in the text so that the suffix in windowP can be aligned with that prefix of the pattern. The distance that we can safely move is in the region [1, m].

2. How do sse4.2 instructions work

_mm_cmpistri(__m128i A, __m128i B, const int mode) with mode=0xc will return a value between[0,16].

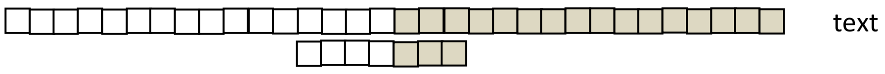
16 means there is no matching, we can move forward 16 chars.

a value between[0,15] means that in that position there is a matching or there is a suffix(of B) which is a prefix of A. We still can move forward that distance safely. Then we must check if we get a matching or just get a prefix matching.

We can see that the distance we move forward is between [0, 16]. we must use unaligned load instruction to load the text.

3. How to use aligned load?

Obviously the unaligned load is slower than aligned one. If we want to use aligned load, we must move forward 16 chars each time.



Let us divide the text as unoverlapped 16-length windows. And let us consider the pattern which length is smaller than 16.

If the matching happens in a window, we definitely can find out it with _mm_cmpistri. So let us focus on the occurrence stride two windows. We can see that there is a suffix(of the previous window) matching in the previous window and a prefix(of the later window) matching in the later window. **The matching prefix of the later window is also the suffix of the pattern. This is the key.**
_mm_cmpistrm(pattern,windowP,_SIDD_UNIT_MASK|_SIDD_CMP_EQUAL_ORDERED) can find out the first part

and _mm_cmpistrm(windowP+1, pattern,_SIDD_UNIT_MASK|_SIDD_CMP_EQUAL_ORDERED) can find out the second part . we can use the two value to check if we can make up the pattern.