16 - POINTERS in C++

quarta-feira, 5 de fevereiro de 2025 07:57

[POINTERS in C++]

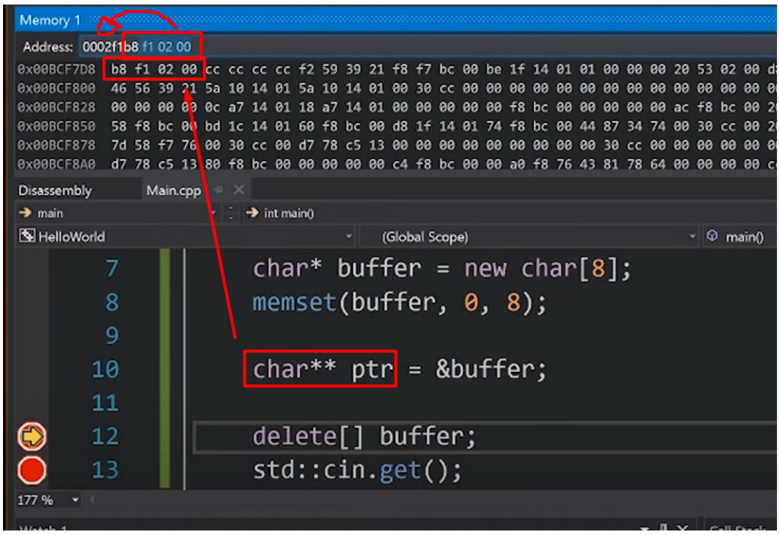
- All application is stored in memory
- pointer is na integer that stores a memory adress
- Memory is one big line of memory, like a single streat n a town
    - every house has na adress (byte of data)
    - tell where the specific byte is
- don't need to use but it's usefull
- pointer is just na adress
- Types are someking of fixture to make our life easier, it's meaninless for pointers
- void* pointer -> don't need a type, just say that the data in that adress is suppose to be from that type
    - A type don't change what a ointer s
    - we can change the type of the pointer and it's ok, will store na integer with the adress anyway
    - Used for maniulation of that memory but not to store pointers (try to set a variable there for example)

```
void* ptr = 0;
    int var = 8;
    ptr = &var;
```

- pointer = 0
    - 0 is not a valid memory adress
    - means null
    - NULL or nullptr is the same
- Every memory we create has a memory adress
    - The & gets the memory adress from a declared value
    - stores the integer adress into the pointer definition
    - with this memory value, we can search it in the memory data (used in visual studio)
- why not always use void *
    - dereference to acess the value on a certain pointer
    - get a value out of a pointer, what is in there?

```
// allocates 8 bytes of memory and return the pointer for the first one
char* buffer = new char[8];
// the a block of memory with a value.. Fill Buffer with zeros for 8 bytes
memset(buffer, 0, 8)
// need to delete the allocated memory
delete[] buffer;
```

- Pointers are just variables, that is why we can have a ointer of a pointer
    - a memory adress that pointes to a nother memory adress
    - stores 4 bytes with the adress of the original pointer (backwards)
    - the original pointer shows the actual data



Address: 0002f1b8 f1 02 00

```
0x00BCF7D8  b8 f1 02 00 cc cc cc cc f2 59 39 21 f8 f7 bc 00 be 1f 14 01 01 00 00 00 20 53 02 00
0x00BCF800  46 56 39 21 5a 10 14 01 5a 10 14 01 00 30 cc 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00BCF828  00 00 00 00 0c a7 14 01 18 a7 14 01 00 00 00 00 00 f8 bc 00 00 00 00 00 ac f8 bc 00
0x00BCF850  58 f8 bc 00 bd 1c 14 01 60 f8 bc 00 d8 1f 14 01 74 f8 bc 00 44 87 34 74 00 30 cc 00
0x00BCF878  7d 58 f7 76 00 30 cc 00 d7 78 c5 13 00 00 00 00 00 00 00 00 00 30 cc 00 00 00 00 00
0x00BCF8A0  d7 78 c5 13 80 f8 bc 00 00 00 00 00 c4 f8 bc 00 00 a0 f8 76 43 81 78 64 00 00 00 00
```

```
char* buffer = new char[8];
memset(buffer, 0, 8);

char** ptr = &buffer;

delete[] buffer;
std::cin.get();
```