Align

10 September 2021 15:22

std::alignof(type) gives alignment in bytes of the specified type

alignof(fundamentalType) = sizeof(same type) .. Eg alignof(int)=4, alignof(double)=8

typedef struct { int a; double b;} S;
alignof(S) for eg gives alignment of 8 bytes i.e. aligned to max element size i.e. double

---

std::alignment_of<type>::value

#include <type_traits>
std::alignment_of<type>::value is same as alignof

---

std::align(alignment, sizetobefit, ptr, maxsize)

#include<memory>

Align in range

Returns a pointer to the first possible address of the range of *space* bytes pointed by *ptr* where it is possible to fit *size* bytes of storage aligned as specified by *alignment*.

The function updates *ptr* to point to such address, and decreases *space* by the number of bytes used for alignment, so that these arguments can be used in repeated calls to this function.

If *space* is not enough to accommodate for the requested aligned buffer, the function returns a null pointer, and alters neither *ptr* nor *space*.

Align finds in buffer pointed by ptr an address which is aligned to alignment where sizetobefit can be fit. If it does it modifies ptr to point to that address and decreases size.
If not null-pointer.
Takes ptr and maxsize by reference

---

alignas() -- IMPORTANT

Specifies the alignment requirement in BYTES of a type or an object

alignas(*expression* **)**

alignas(*type-id* **)**

alignas(*pack* **...** **)**

1) `alignas(expression)` must be an integral constant expression that evaluates to zero, or to a valid value for an alignment or extended alignment.
2) Equivalent to `alignas(alignof(type))`

alignas(32)
alignas(int) = alignas(alignof(int)) = alignas(4)

```
// every object of type struct_float will be aligned to alignof(float) boundary
// (usually 4):
struct alignas(float) struct_float
{
    // your definition here
};

// every object of type sse_t will be aligned to 32-byte boundary:
struct alignas(32) sse_t
{
    float sse_data[4];
};

// the array "cacheline" will be aligned to 64-byte boundary:
alignas(64) char cacheline[64];
```