

Simple static memory manager

Manage a static allocated buffer as a set of fixed-size memory chunks, giving the possibility to alloc and free a certain number of them. Allocated chunks are contiguous. Example usage:

```
char buffer[10240];

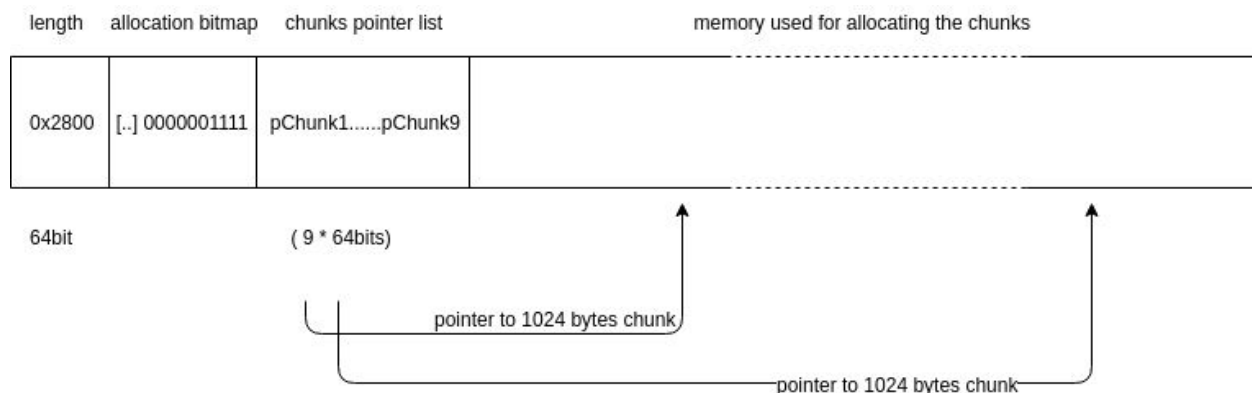
memInit((void*)buffer, size);

uint32_t* p1 = 0;
memAlloc((void*)buffer, 4, &p1);
//p1 is now a pointer to a memory 4 * 1024Bs
memFree((void*)buffer, &p1);
```

Allocation scheme

For the example above, the given buffer can hold up to 9 chunks (each 1024B large). Internally, a certain number of words are used for tracking information about which chunks is free or not, the list of pointers to each chunk, and the length of each chunk.

The first 64bits are the size of the buffer. Then, there is the allocation bitmap (1 == free, 0 == used). Then there is the chunks pointed list and finally the area reserved for each chunk. In the above example, a one-word allocation bitmap is used, since the total number of alloc-able chunks is 9 and one word is enough to store 9 bits. The chunks pointer list is made of 9 words. The rest of the words are used for the chunks.



Given a backing buffer, *sizeInByte* big, and the chunk size of 1024 bytes, (considering a 64 bits architecture), the number of possible chunks is:

$$1024x + 8x + x/64 + 8 = \text{sizeInByte} \quad (x/64 \text{ as ceiling}(x/64))$$

$$1032x + x/64 = \text{sizeInByte} - 8$$

$$1032 \cdot 64x + x = 64 * (\text{sizeInByte} - 8)$$

$$x = 64 * (\text{sizeInByte} - 8) / (1032 * 64 + 1);$$

Given a buffer of 10240 bytes for instance, 9 chunks of 1024 bytes can be allocated.

$$\text{sizeInByte} = 10240$$

$$1024x + 8x + x/64 + 8 = 10240$$

$$x = 64 * (10240 - 8) / (1032 * 64 + 1);$$

$$x = 9$$

Performance

Roughly, about 10% of space is wasted for the chunks management.

In terms of time, the time for finding *m* chunks space, out of *n* possible, *m* * *n* comparison must be done hence $O(n^2)$ for visiting the allocation bitmap.

Return codes

The following is the error codes list:

0, success

1, the backing handle is null.

2, the backing buffer is smaller than the chunk size, or the backing buffer size is not multiple of the chunk size.

3, if either size (requested number of chunks) is 0, or the size is bigger than the max num of possible allocations.

4, if ptr is unknowns (is not managed) by this manager.

