

Dynamic Memory Allocation

C

- Stack destroys on function return
data in heap is shared through dynamic libraries.
- Runtime memory allocation
 - Decision at runtime
 - Wastage of memory is restricted

Why dynamic MA in C.

- Changing the size of an array is not allowed.
- Memory can be allocated or deallocated at run time.

Function for DMA in C

- 1) `malloc()` - Most used, Memory allocation.
- 2) `calloc()`
- 3) `free()`
- 4) `realloc()`

1) `malloc()`

It takes number of bytes to be allocated as an input and returns a pointer of type void.

`int *ptr;`

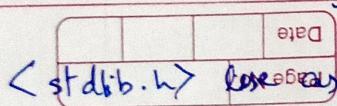
Format : `ptr = (int *) malloc (30 * sizeof (int))`

Type cast

(space)

(4)

of int



<stdlib.h> header file.

- It can be used as array. ($\text{ptr}[i]$)

2) `calloc` :- Continuous allocation.

It initializes every memory block with a default value of 0.

Syntax : $\text{ptr} = (\text{float}^*) \text{calloc}(30, \text{sizeof}(\text{float}))$;

Allocates contiguous

- Space in memory for 30 blocks floats, initialized by 0 at starting.

If something goes wrong during memory allocation, then a NULL pointer is returned.

3) `free` function :- To deallocate memory.

The memory allocated by `malloc` & `calloc` is not deallocated automatically.

4) `realloc()` function :- To allocate memory of new size using the previous pointer and size.

Syntax :-

$\text{ptr} = \text{realloc}(\text{ptr}, \text{newsize})$;