

8. Dynamic memory allocation

Introduction:

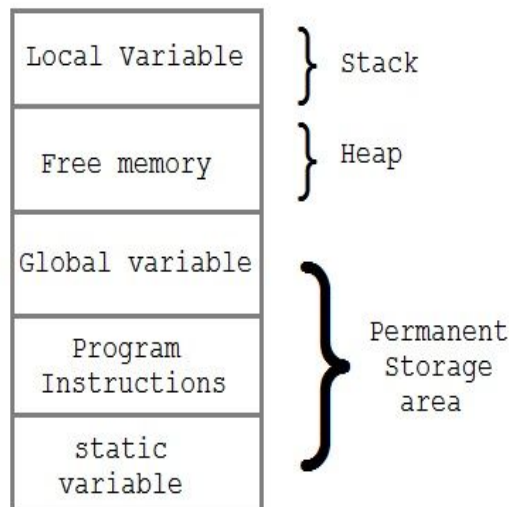
In C, the exact size of array is unknown until compile time, i.e., the time when a compiler compiles your code into a computer understandable language. So, sometimes the size of the array can be insufficient or more than required.

Dynamic memory management techniques allows us to allocate additional memory space or to release unwanted space at run time, thus, optimizing the use of storage space.

Dynamic memory allocation :

Dynamic memory allocation allows your program to obtain more memory space while running, or to release it if it's not required. In simple terms, Dynamic memory allocation allows you to manually handle memory space for your program.

Memory allocation process



Global variables, static variables and program instructions get their memory in permanent storage area whereas local variables are stored in area called Stack. The memory space between these two region is known as Heap area. This region is used for dynamic memory allocation during execution of the program. The size of heap keep changing.

Memory allocation functions :

Function	Task
malloc	Allocates requested size of bytes and returns a pointer first byte of allocated space
calloc	Allocates space for an array elements, initializes to zero and then returns a pointer to memory
free	deallocate the previously allocated space
realloc	Change the size of previously allocated space

Functions used in Dynamic Memory Allocation :**malloc():**

malloc()) is used to allocate a certain amount bytes of memory during the execution of a program. malloc() allocates size_in_bytes bytes of memory from heap, if the allocation succeeds, a pointer to the block of memory y is returned else NULL is returned.

malloc()) returns an uninitialized memory for you to use.

Malloc()) can be used to allocate space for complex data types such as structures.

Syntax:

```
ptr_var = (cast_type *)malloc(size_in_bytes);
```

Example :

```
#include<stdio.h>
int main()
{
    int *p ;
    p = (int *)malloc(size of(int));
    *p =25;
    printf(“%d”,*p) ;
    free(P);
}
```

Calloc():

calloc() is used to allocate a block of memory during the execution of a program, e.g. for an array.

calloc() allocates a region of memory large enough to hold no_of_blocks of size size_of_block each, if the allocation succeeds then a pointer to the block of memory is returned else NULL is returned.

Syntax:

```
ptr_var=(cast_type *)calloc(no_of_blocks ,size_of_block);
```

Example :

```
#include<stdio.h>
int main ()
{
int i,n;
int *p;
printf ("Enter how many numbers:");
scanf ("%d",&n);
p = (int*) calloc (n, size of(int));
for (i=0; i<n; i++)
{
scanf ("%d",p);
p++;
}
}
```

Questions :

Static Memory Allocation	Dynamic Memory Allocation
If memory is allocated to variables before execution of program starts then it is called static memory allocation .	If memory is allocated at run time (during execution of program)then it is called dynamic memory allocation.
It is fast and saves running time.	It is bit slow.
It allocates memory from stack.	It allocate es memory from heap.
Allocated memory stays from start to end of program.	Memory can be allocated at any time and can be release ed at any time.
e.g. int i; float j;	e.g. p = malloc(size of(int));