Dynamic Memory Allocation

ESC108A Elements of Computer Science and Engineering

B. Tech. 2017

Course Leaders:

Roopa G.

Ami Rai E.

Chaitra S.

Objectives

- At the end of this lecture, student will be able to
  - explain dynamic memory allocation in C programming language

Contents

- Dynamic Memory Allocation
- Library functions for dynamic memory allocation
- Command line arguments

Three kinds of Memory

1. Fixed memory
   – Executable code, Global variables, Static variables ,etc.,

2. Stack memory
   – Local variables for functions, whose size can be determined at call time, Information saved at function call and restored at function return, etc.,

3. Heap memory
   – Structures whose size varies dynamically (e.g. variable length arrays or strings), Structures that are allocated dynamically (e.g. records in a linked list),etc.,

Types of Allocation of Memory

1. Static Allocation
   – Allocation of memory space at compile time

2. Dynamic Allocation
   – Allocation of memory space at run time

Dynamic Memory Allocation

- The ability for a program to
  - obtain more memory space at execution time to hold new nodes
  - release space no longer needed

- In C, there are 4 library functions under **"stdlib.h"** for dynamic memory allocation

1. malloc()
2. calloc()
3. realloc()
4. free ()

malloc()

- Allocates requested size of bytes and returns a pointer first byte of allocated space
- returns null pointer if it couldn’t able to allocate requested amount of memory
- does not initialize the memory allocated during execution
- It carries garbage value
- Example:

  int *p;

  p = (int *) malloc (n * sizeof(int)); //returns the sizeof an integer on the machine, multiply by n and malloc that many bytes

calloc()

- allocates multiple blocks of memory each of same size
  - malloc() allocates single block of memory
- initialize every byte to zero
- return pointer to the block (NULL if unable to allocate block)

- Example:

  ptr=(float*) calloc (25,sizeof(float)); //allocates contiguous space in memory for an array of 25 elements each of size of float, i.e, 4 bytes

Memory Re-Allocation

realloc()

- For growing/shrinking the allocated memory
- change the block size to new_size
- return pointer to resized block
  - If block size is increased, contents of old block may be copied to a completely different region

- Example:

```
int *p;
p = (int *) malloc (n*sizeof(int));
p = (int *) realloc (p, m*sizeof(int)); or realloc(p,m);
```

Freeing the Memory

free()

- Always free all dynamically allocated memory after use
- frees the allocated memory by malloc (), calloc (), realloc () functions and returns the memory to the system

- Example:

```
int *p;
p = (int *) malloc ( n*sizeof(int));
free(p);
```

Using Command-Line Arguments

- It is possible to pass arguments to main from a command line by including parameters int argc and char *argv[] in the parameter list of main

```
int main( int argc, char *argv[] )
```

- Parameter argc
  - Receives the number of command-line arguments
- Parameter argv
  - An array of strings in which the actual command-line arguments are stored

Using Command-Line Arguments contd.

- Common uses of command-line arguments
  - passing options to a program
  - passing filenames to a program

Example:

```
$ myProgram a 22
```

argc: 3

argv[0]: “myProgram"

argv[1]: “a"

argv[2]: “22"

Summary

- Dynamic Allocation is the allocation of memory space at run time
- In C, there are 4 library functions under **"stdlib.h"** for dynamic memory allocation
  1. malloc()
  2. calloc()
  3. realloc()
  4. free ()
- Dynamic allocation of arrays require use of pointers
- It is possible to pass arguments to main from a command line by including parameters int argc and char *argv[] in the parameter list of main

Further Reading

Kernighan, B. W. and Richie, D. (1992) *The C Programming Language.* 2nd ed., New Delhi:PHI.