Memori Dinamis

Memori dinamis atau Dynamic Memory adalah suatu teknik alokasi memori yang memungkinkan memesan memori di saat program berjalan atau eksekusi program bukan di saat pembuatan atau penulisan program. kita juga dimungkinkan untuk menghapus memori saat tidak dibutuhkan lagi.

Misalnya, program berjalan pada awalnya memiliki sedikit memori data yang sudah kita pesan dan tulis saat pembuatan program ( Statis ). Kemudian suatu saat seorang pengguna program tersebut membutuhkan penyimpanan data lebih yang tak terduga, otomatis dia membutuhkan memori lagi untuk menyimpan data tersebut. Dengan teknik memori dinamis kita bisa memungkinkan pengguna tersebut mendapatkan memori lebih. Memungkinkan kita untuk membuat program yang dapat meminta memori tambahan sebanyak yang diperlukan untuk penyimpanan data kepada sistem. Bukan hanya memesan memori lebih kita juga dimungkinkan menghapus dan membebaskan memori yang telah di pesan saat tidak membutuhkanya lagi. Memori yang bebas dan tidak terpakai akan tersedia untuk pemesanan atau alokasi memori berikutnya. dan Satu-satunya cara untuk mengakses variabel dinamis adalah dengan menggunakan pointer.

Pada contoh program yang penulis berikan pada artikel sebelumnya, mereka masih bersifat memori statis atau variabel statis adalah variabel yang dideklarasikan dan dinamai pada saat penulisan program (identifier). Memori dipakai oleh variabel ini akan tetap ada selama program itu berjalan. Mereka tidak aktif dalam alokasi memori saat program di jalankan, dengan arti pengguna hanya akan memanfaatkan memori yang sudah di siapkan pada program tersebut. Kekurangan variabel statis tidak dapat dibuat ataupun dihapus pada saat program berjalan.

memori di dalam C++ dibagi menjadi dua yaitu :

Stack / Tumpukan : Semua variabel yang dideklarasikan di dalam function akan mengambil memori dari stack.

Heap : adalah memori yang tidak dipakai dalam program dan bisa digunakan untuk mengalokasi memori secara dinamis di saat program jalan.

Pada bahasa pemrograman C Kita bisa menggunakan malloc, calloc, realloc, dan free. dan fungsi tersebut juga tersedia pada bahasa pemrograman C++ dengan new dan delete, menyediakan fungsi yang sama dan disarankan oleh programmer bahasa C++.

Macam-macam Operator Memori

Malloc

Malloc adalah fungsi standar untuk mengalokasi memori, dalam bahasa C, juga tersedia pada C++, Malloc terdapat pada <stdlib.h> / <cstdlib>. Malloc berfungsi untuk mengatur alokasi memori pada heap sebanyak byte yang dibutuhkan, pengaksesan blok memory dilakukan melalui sebuah pointer dan pada saat pemesanan memori tidak dilakukan inisialisasi.

Bentuk umum penulisan

void\* malloc (size\_t size) ;

Size = besar memori tipe data.

Contoh Penulisan

varFoo = (char\*) malloc (sizeof(char)) ;

Calloc

Calloc adalah fungsi standar untuk mengalokasikan memori dan blok untuk penyimpanan (array), dalam bahasa C, juga tersedia pada C++, Calloc terdapat pada <stdlib.h> / <cstdlib>. Calloc berfungsi untuk mengalokasi beberapa blok atau elemen memori secara diamis dari heap dan setiap elemen yang berukuran size. Bedanya calloc dan malloc adalah calloc secara otomatis akan meng-inisialisasi semua elemen dengan NOL.

Bentuk umum penulisan

void\* calloc (size\_t num, size\_t size) ;

Num = jumlah block yang akan dipesan.

Size = besar memori tipe data.

Contoh Penulisan

varFoo = (int\*) calloc (5, sizeof(int)) ;

Realloc

Realloc adalah fungsi standar untuk menambah atau mengurangi ukuran blok, dalam bahasa C, juga tersedia pada C++, realloc terdapat pada <stdlib.h> / <cstdlib>. Realloc berfungsi untuk menambah atau mengurangi blok atau elemen memori yang telah dialokasikan secara dinamis dari heap memory dan setiap elemen yang berukuran pada [size].

Bentuk umum penulisan

void\* realloc (void\* ptr, size\_t size) ;

Ptr = pointer ke suatu alokasi memori yang sizenya/block akan diubah.

Size = besar memori tipe data.

Contoh penulisan

varFoo = (int\*) realloc (varPertama, sizeof(int)) ;

Free

Free adalah fungsi standar untuk menghapus, melepaskan atau membebaskan memori, dalam bahasa C, juga tersedia pada C++, free terdapat pada <stdlib.h> / <cstdlib>. Free berfungsi untuk menghapus, melepaskan atau membebaskan memori yang telah terpakai dan tidak digunakan lagi, memori tersebut akan di kembalikan ke sistem dan tersedia untuk alokasi/pemesanan memori berikutnya.

Bentuk umum penulisan

void free (void\* ptr) ;

ptr = target pointer untuk dbebaskan.

Contoh penulisan

free (targetVar) ;

New

New adalah fungsi standar untuk memesan atau alokasi memori, berasal dari bahasa C++, fungsi ini disarankan untuk pengguna C++. Fungsi new sama seperti malloc pada bahasa C. new berfungsi untuk mengatur alokasi memori secara dinamis pada heap sebanyak byte yang sebanyak tipe data yang disebutkan.

Bentuk umum penulisan

varPointer = new typeData ;

varPointer = new typeData [index] ; //array

Contoh Penulisan

varPointer = new int;

varPointer = new int [10]; //array

Delete

Delete adalah fungsi standar untuk menghapus atau membebaskan memori yang telah di pesan, berasal dari bahasa C++, fungsi ini disarankan untuk pengguna C++, fungsi delete sama seperti free pada bahasa C. delete berfungsi untuk menghapus, membebaskan atau melepaskan memori yang telah terpakai dan tidak digunakan lagi, memori tersebut akan di kembalikan ke sistem dan tersedia untuk alokasi/pemesanan memori berikutnya.

Bentuk umum penulisan

delete pointer;

delete []pointer; //array

Penting : Setelah alokasi memori sangat disarankan untuk menghapus atau membebaskan jika sudah tidak dibutuhkan lagi, hal itu untuk mencegah kebocoran memori atau memory leak.