Realloc (Re-Allocation Memory) adalah salah satu fungsi untuk mengubah ukuran blok memori yang disebutkan, tersedia pada bahasa pemrograman C dan juga pada bahasa pemrograman C++. Realloc tersedia pada header <stdlib.h> / <cstdlib>.

RE-ALLOCation atau dalam bahasa Indonesia adalah mengalokasikan kembali, Realloc berfungsi untuk mengubah ukuran blok memori yang disebutkan dari memori yang sudah dialokasikan sebelumnya menggunakanan fungsi alokasi seperti malloc, calloc, new dan lain-lain. Fungsi akan memindahkan isi nilai sesuai dengan isi alamat memori yang lama dan dipindahkan ke lokasi memori baru yang telah di pesan dan diperbaharui sesuai kebutuhan yang telah disebutkan. Dan memori yang lama akan di lepaskan atau dihapus. Untuk mengakses kedua memori (lama dan baru) kita bisa menggunakan pointer, dari pointer memori lama akan dipindahkan ke pointer memori baru. jika berhasil melakukan alokasi dan pemindah memori maka akan memberikan nilai berbentuk alamat kepada variabel pointer bersangkutan, Jika tidak berhasil melakukan alokasi dan pemindahan maka pointer memori baru akan bernilai null dan pointer memori lama tidak akan dihapus atau dilepaskan.

Tapi jika pointer memori lama atau pointer yang disebut untuk di realloc adalah pointer NULL atau belum menunjuk ke memori. Maka fungsi realloc akan bekerja seperti fungsi malloc. jika berhasil melakukan alokasi memori maka akan memberikan nilai berbentuk alamat kepada variabel pointer bersangkutan, Jika tidak berhasil melakukan alokasi maka akan memberi nilai “NULL” pada pointer.

Dinamis memori adalah suatu teknik alokasi memori yang memungkinkan memesan atau alokasi memori di saat program berjalan atau eksekusi program, bukan saat pembuatan atau penulisan program. Contoh kasus adalah, ketika pengguna program membutuhkan memori lebih untuk menyimpan data bukan dengan hanya menggunakan memori statis yang telah kita siapkan sebelumnya pada pembuatan program, maka teknik dinamis memori bisa kita manfaatkan pada kasus tersebut.

Bentuk umum penulisan / Syntax :

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

Parameter :

Ptr = pointer yang menunjuk ke memori.

Size = ukuran byte baru.

Contoh Penulisan :

varPtrBaru = (int\*) realloc (varPtrLama, 5+sizeof(int)) ; //5+4=9

Return Value :

Akan memberikan alamat memori yang memiliki nilai sama seperti memori lama, tetapi memori tersebut memiliki jumlah byte yang berbeda. Alamat memori lama akan dibebaskan dan akan tersedia untuk alokasi lainya. jika gagal dalam penugasan maka akan memberikan nilai NULL pada pointer baru dan akan memberikan memori denga nilai sama pada pointer lama (memori lama tidak di bebeaskan).

Jika Pointer Lama adalah NULL maka realloc akan bekerja seperti malloc. Jika fungsi realloc berhasil maka realloc akan memberikan nilai berbentuk alamat memori kepada pointer dan jika tidak (gagal dalam memesan memori) akan memberikan nilai NULL pada pointer.

Contoh Program :

#include <stdlib.h>

#include <iostream>

#define NEWLINE cout<<endl; //Konstanta

using namespace std;

int main (){

int \*pointer = nullptr, \*newPointer = nullptr ; //Null Pointer

cout<<"Pointer = "<<pointer<<endl; //Null

cout<<"newPointer = "<<newPointer<<endl; //NULL

NEWLINE

pointer = (int \*) malloc(sizeof(int));

cout<<"pointer = "<<pointer<<" = "<<\*pointer<<endl;

cout<<"newPointer = "<<newPointer<<endl; //NULL

NEWLINE

\*pointer = 20000;

cout<<"pointer = "<<pointer<<" = "<<\*pointer<<endl;

cout<<"newPointer = "<<newPointer<<endl; //NULL

NEWLINE

newPointer = (int \*) realloc(pointer, 10); //Fungsi Calloc

cout<<"pointer = "<<pointer<<" = "<<\*pointer<<endl;

cout<<"newPointer = "<<newPointer<<" = "<<\*newPointer<<endl;

NEWLINE

free (pointer); //Hapus

free (newPointer); //Hapus

cout<<"pointer = "<<pointer<<" = "<<\*pointer<<endl; //Kembali pada Heap

cout<<"newPointer = "<<newPointer<<" = "<<\*newPointer<<endl; //Kembali pada Heap

return 0;

}

Diatas adalah contoh yang menunjukan contoh penggunaan realloc. Contoh di atas akan menampilkan tahap demi tahap perubahan dari kedua memori yang di tunjuk oleh pointer, pernyataan keluaran pertama akan menampilkan pointer yang berisi NULL. Selanjutnya akan dilakukan alokasi menggunakan fungsi malloc, pada variabel “pointer”, dan selanjutnya akan ditampilkan perubahannya. Setelah itu variabel pointer kita beri operasi penugasan “\*pointer = 20000;”. Anda akan melihat perubahanya pada keluaran berikutnya. Dan selanjutnya merupakan contoh penggunaan fungsi realloc, dari “pointer” ke “newPointer” sebesar 10 byte, perubahan bisa kita lihat pada tampilan berikutnya.

Pada variabel “pointer” akan menunjuk alamat yang sama tetapi akan mempunyai nilai seperti nilai awal sebelum kita memberikan operasi penugasan pada pointer, dengan arti bahwa memori tersebut sudah pada heap dan isi memori tersebut sudah di pindahkan ke memori yang di tunjuk “newPointer”.