Багшаас асуу: Node? In C, we can represent a node using structures

Double pointer:

When you pass in a pointer to a function, the address value is being copied over to the function parameter. Due to the function's scope, that copy will vanish once it returns.

By using a double pointer, you will be able to update the original pointer's value. The double pointer will still be copied by value, but that doesn't matter. All you really care is modifying the original pointer, thereby bypassing the function's scope or stack.

Linked list

Массивтай адилаар LinkedList нь шугаман өгөгдлийн бүтэц юм. Харин массив санах ойд дарааллаж байрладаг бол LinkedList нь дарааллаж байрладаггүй ба элементүүд нь заагч ашиглаж хоорондоо холбогддог.

Яагаад Linked List гэж?

Массивийг нэгэн төрлийн өгөгдлийг байршуулахад ашиглаж болно, гэхдээ дараах хязгаарлалтуудтай байдаг.

1. Массивийн урт тогтмол байдаг: Бид дээд талдаа хэдэн элемент ашиглагдахыг мэдэж байх ёстой. Мөн нөөцөлсөн ой маань хэрэглээнээс хамааран байнга нөөцөлсөн дээд хязгаартай тэнцүү байх ёстой.
2. Массивт шинэ утга нэмэх нь их удаан. Шинэ элементэд шинэ өрөө бий болгохын тулд оршиж байгаа элементүүд байраа болих хэрэгтэй болдог.

Жишээ нь: доорх массив байлаа

id[] = [1000, 1010, 1050, 2000, 2040].

бид 1005 элементийг дарааллийн дагуу оруулахыг хүслээ. Ингэхийн тулд 1000-с хойших бүх элемент хойшоо нэг нүүх хэрэгтэй болно. Устгах үйлдэл ч мөн адил.

Массивтай харьцуулахад листийн давуу тал.

1. Динамик хэмжээ
2. Элемент нэмэх, устгах нь хэцүү

Линкэд лист нь:

1. Дурын индекст шууд хандаж чадахгүй. Эхний элементээс эхэлж давтаж хандах хэрэгтэй болдог.
2. Элемент бүрт шаардагдаж байгаа заагчид нэмэл санах ой ашиглана
3. Not cache friendly. Since array elements are contiguous locations, there is locality of reference which is not there in case of linked lists

Линкэд лист нь листийн эхний элементийг зааж буй заагчаар илэрхийлэгдэнэ. Эхний элементийг толгой гэнэ. Толгойн утга NULL байвал линкэд листийн хоосон байна.

Лист дахь бүх элемент багадаа 2 хэсгээс бүрдэнэ.

1. Өгөгдөл
2. Заагч(Заалт) даараагийн элементийг заах

#include <stdio.h>

int main () {

int var;

int \*ptr;

int \*\*pptr;

var = 3000;

/\* take the address of var \*/

ptr = &var;

/\* take the address of ptr using address of operator & \*/

pptr = &ptr;

/\* take the value using pptr \*/

printf("Value of var = %d\n", var );

printf("Value available at \*ptr = %d\n", \*ptr );

printf("Value available at \*\*pptr = %d\n", \*\*pptr);

return 0;

}