

# ХАЯГ, ХАЯГАН ХУВЬСАГЧ БА ЗААЛТАН ХУВЬСАГЧ (ЛАБОРАТОРИ №2)

Э.Уранчимэг

ХШИУС, МКУТ, Программ хангамжийн III түвшний оюутан

## 1. ОРШИЛ

Энэ тайланд хаяг, хаяган хувьсагч, new operator болон санах ойн цоорхойн талаар дэлгэрэнгүй тайлбарласан. Мөн хаяган хувьсагч, заалтан хувьсагчууд ашиглан бичсэн хэрэглэгчдийн функцийг хэрэгжүүлэлтийг харуулсан. Хавсралт хэсгээс дэлгэрэнгүй кодыг унших боломжтой.

## 2. ЗОРИЛГО

Хаяган хувьсагчийг ашиглан хүснэгтийн элементүүдэд хандаж, функцад параметруудыг хаягаар нь дамжуулж түүнийг функц нь хүлээн авахдаа хаяган хувьсагчид хадгалан хэрэглэгчийн функцийг хэрэгжүүлсэн. Үүний тулд дараах зорилтуудыг тавьж ажилласан:

1. Санах ойн хаяг, хаяган хувьсагчийг тодорхойлох,
2. Хаяган хувьсагчийн санах ойд эзлэх хэмжээг тодорхойлох,
3. Хаягийн арифметикийг хэрхэн ашигласныг тайлбарлах,
4. Хаяган хувьсагч ашиглан функц бичих,
5. Заалтан хувьсагчийг тодорхойлох,
6. Заалтан параметр ашиглах функц бичих,
7. Ойн цоорхой хэрхэн үүсдэгийг тодорхойлох

## 3. ОНОЛЫН СУДАЛГАА

### 3.1 САНАХ ОЙН ХАЯГ

Хувьсагч нь зарласан төрлөөсөө хамаарч санах ойн үүрэнд өөр өөр багтаамж эзлэн байрлана. Санах ойн 1 үүр нь 8 бит мэдээлэл агуулах боломжтой. Үүрийн дугаарыг санах ойн хаяг гэнэ. Тооцооллын хувьд санах ойн хаяг нь програм хангамж, техник хангамжийн янз бүрийн түвшинд ашиглагддаг компьютерийн санах ой дахь тодорхой байршлыг заадаг. Санах ойн хаягууд нь тэмдэглэгээгүй бүхэл

Computer		Programmers		
Address	Content	Name	Type	Value
90000000	00	sum	int (4 bytes)	000000FF (255 <sub>10</sub> )
90000001	00			
90000002	00			
90000003	FF			
90000004	FF	age	short (2 bytes)	FFFF (-1 <sub>10</sub> )
90000005	FF			
90000006	1F	average	double (8 bytes)	1FFFFFFFFFFFFFFF (4.45015E-308 <sub>10</sub> )
90000007	FF			
90000008	FF			
90000009	FF			
9000000A	FF			
9000000B	FF			
9000000C	FF			
9000000D	FF			
9000000E	90	ptrSum	int* (4 bytes)	90000000
9000000F	00			
90000010	00			
90000011	00			

Note: All numbers in hexadecimal

тоогоор (unsigned int) ердийн байдлаар харагдуулдаг, тогтмол урттай цифрүүдийн дараалал байдаг. Ийм тоон семантик нь CPU-ийн онцлог (зааврын заагч, өсөн нэмэгдэж буй хаягийн бүртгэл гэх мэт), мөн янз бүрийн програмчлалын хэлээр батлагдсан массив шиг санах ойн ашиглалт дээр суурилдаг.

### 3.2 ХАЯГАН ХУВЬСАГЧ

Санах ойн хаяг хадгалдаг хувьсагчийг хаяган хувьсагч (pointer) гэнэ. Санах ойн хаяг агуулах хувьсагчийг хаяган хувьсагч (pointer) гэнэ. Хаяган хувьсагчийн утга нь санах ойн хаяг байна. Хаяган хувьсагчийг зарлахдаа нэрний өмнө(\*) тавьдаг. Мөн хаяган хувьсагчийн зааж байгаа утганд хандахдаа (\*) хэрэглэдэг. C/C++ хэлүүд дээр заагчийн хэмжээ компьютерийн үйлдлийн систем, CPU архитектур зэргээсээ шалтгаалаад 32 битийн компьютерийн хувьд заагчийн хэмжээ 4 байт байх бол 64 битийн компьютер дээр 8 байт байна.

### 3.3 NEW ОПЕРАТОР

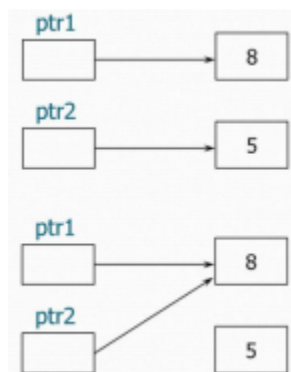
Шинэ оператор нь RAM руу санах ой хуваарилах хүсэлтийг илэрхийлнэ. Хэрэв хангалттай санах ой байгаа бол шинэ оператор санах ойг эхлүүлж, шинээр хуваарилагдсан болон эхлүүлсэн санах ойн хаягийг заагч хувьсагч руу буцаана. Энгийн массив зарлах, new ашиглан санах ойн блок хуваарилах хоёрын хооронд ялгаа бий. Хамгийн чухал ялгаа нь энгийн массивуудыг хөрвүүлэгч нь салгах явдал юм (Хэрэв массив нь локал бол функц буцаж ирэх эсвэл дуусах үед хуваарилагдана). Гэсэн хэдий ч динамикаар хуваарилагдсан массивууд нь программист хуваарилагдах эсвэл програм дуусах хүртэл үргэлж тэнд хэвээр үлдэнэ. Динамикаар хуваарилагдсан санах ойг чөлөөлөх нь програмистын үүрэг тул програмистуудад C++ хэл дээр delete операторыг өгдөг.

### 3.4 САНАХ ОЙН ЦООРХОЙ

New оператораар санах ойг нөөцлөн авч болдог. New - ээр нөөцөлсөн санах ойд delete оператор ашиглахгүй л бол хэзээ ч чөлөөлөгдөхгүй. Ашиглаж дууссан үедээ Delete p; гэж чөлөөлнө. Delete - ээр чөлөөлөөгүй программыг санах ойн цоорхой гэнэ. Санах ойг дүүргэхийн тулд ийм төрлийн вирус хийж болдог.

Жишээ:

```
int main() {  
    int *ptr1 = new int;  
    int *ptr2 = new int;  
    *ptr1 = 8;  
    *ptr2 = 5;  
    ptr1 = ptr2;  
    cout << ptr2 << " " << ptr1;  
}
```



Дээрх ptr1, ptr2 заагчууд ptr1-ийн нөөцөлсөн санах ойд байрлах 8-ийг заана. Харин ptr2-ийн нөөцөлсөн санах ой эзэнгүйдэж байна. Үр дүн нь яг адилхан утга хэвлэнэ.

```
PS D:\OOP\lab2> ./a
0x10219f0 0x10219f0
```

### 3.5 SIZEOF() КОММАНД

C++ хэл дээрх sizeof() оператор нь эмхэтгэх үед өгөгдлийн төрөл, хувьсагч болон тогтмолуудын хэмжээг тооцдог нэгдмэл оператор юм. sizeof() оператороос буцаасан утга нь size\_t төрлийн байна. sizeof() операторын операндуудыг өгөгдлийн төрөл, илэрхийлэлд ашиглаж болно.

sizeof оператор нь операндын төрлийн объектыг хадгалахад шаардагдах хадгалах сангийн хэмжээг байтаар илэрхийлдэг. Энэ оператор нь программдаа машинаас хамааралтай өгөгдлийн хэмжээг зааж өгөхөөс зайлсхийх боломжийг олгодог.

### 3.5 ЗААЛТАН ХУВЬСАГЧ

Заалтан хувьсагч нь зарлагдсан хувьсагчид шинэ нэр өгдөг. Заагчтай адил заалт нь объектын хаягийг хадгалах замаар хэрэгждэг. Заалтан хувьсагчийг функцийн параметрээр хэрэглэхэд тохиромжтой. Ингэснээр ой хэмнэхээс гадна эх өгөгдөл рүү нь шууд хандах боломжтой болдог.

## 4. ХЭРЭГЖҮҮЛЭЛТ

I. Доорх кодыг туршиж үр дүнг тайлабарлан бич.

```
int main() {
    char *p1;
    int *p2;
    double *p3;
    cout<<sizeof(p1)<<endl<<sizeof(p2)<<endl<<sizeof(p3);
}
```

Хаяган хувьсагч нь хаяг буюу дугаарласан натурал int хэлбэрийн тоонуудыг хадгалдаг тул урд бичигдсэн төрлөөсөө үл хамааран бүгд int төрөл хадгалдаг. Тиймээс хэвлэсэн үр дүнгүүд нь бүгд адилхан 4 байт байна.

Үр дүн:

```
PS D:\OOP\lab2> g++ 1.cpp
PS D:\OOP\lab2> ./a
4
4
4
```

## II. Доорх кодыг туршиж мөр бүрийн үр дүнг тайлбарлан бич.

```
int main() {
    int a=125; // int төрлийн а хувьсагчид 125 гэсэн утга өгсөн.
    int *p = &a; // а хувьсагчийн хаягийг р хаяган хувьсагчид утга болгон
    онооно.
    cout<<p<<endl; // р заагчийн санах ойд зааж байгаа а-ийн хаягийг
    хэвлэнэ
    cout<<*p<<endl; // р заагчийн санах ойд зааж байгаа а-ийн утгыг
    хэвлэнэ
    p++; /* хаяган төрлийг нэгээр нэмэгдүүлэхэд дараагийн тоог заах буюу
    заагчийн
        зааж буй төрлийн хэмжээгээр утга нь нэмэгдэнэ. */
    cout<<p<<endl; // int төрлийн хувьсагчийн хаяг 4 байтаар нэмэгдэнэ.
    cout<<*p<<endl; // р заагчийн санах ойд зааж байгаа нэмэгдүүлсэн утгыг
    хэвлэнэ
}
```

Үр дүн:

```
PS D:\OOP\lab2> g++ 2.cpp
PS D:\OOP\lab2> ./a
0x61ff08
125
0x61ff0c
6422284
```

## III. Доорх кодын мөр бүрийг тайлбарла

```
int main() {

    int numbers[5]; // бүхэл тоон төрөлтэй 5 утга хадгалах хүснэгт үүсгэж
    байна.
    int * p; // бүхэл тоо заах хаяган хувьсагч зарлаж байна.
    p = numbers; *p = 10; /* хаяган хувьсагч numbers хүснэгтийг зааж
    байна. Заагч
        нь мөн хүснэгтийн эхний элементийн хаягийг заадаг. */
    p++; *p = 20; /* заагчийн утга 1-р тооны хаягийг утга болгон авна.
    Заагч
        1-р тоог зааж байгаа учир 1-р тооны утгыг 20 болгож байна. */
    p = &numbers[2]; *p = 30; /*хүснэгтийн 2-р элементийн хаягийг р
    заагчид олгоод
        хаягаар нь дамжуулан 2-р элементийн утгыг 30 болгосон. */
}
```

```

    p = numbers + 3; *p = 40; /* numbers бол хҮснэгтийн эхлэл хаяг ҮҮнийг
гурваар
    нэмэгдүүлснээр хҮснэгтийн гуравдугаар элементийн хаягтай ижил утгатай
болно.
    Тиймээс заагч гурвадугаар элементийг зааж, утгыг нь 40 болгож байна.
*/
    p = numbers; *(p+4) = 50; /* p хаяган хувьсагч нь хҮснэгтийн эхлэл
хаягийг авна.
    Хаягийн арифметик хэрэглэн 4-р элементийг зааж numbers[4]-ийн утгыг 50
болгож
    байна. */
    for (int n=0; n<5; n++)
        cout << numbers[n] << ", "; //ХҮснэгтийн утгуудыг хэвлэх үйлдэл хийнэ.
}

```

Үр дүн:

```

PS D:\00P\lab2> g++ 3.cpp
PS D:\00P\lab2> ./a
10, 20, 30, 40, 50,

```

IV. Хаяган хувьсагч ашиглан (функцын параметер нь хаяган хувьсагч байна) хоёр хувьсагчийн утгыг солих хэрэглэгчийн функц бич.

```
void swap(int *a, int *b)
```

Параметр нь хаяган хувьсагч байх swap функц зарлаад дотор нь хувьсагчуудынхаа утгыг солих үйлдлүүдийг бичнэ.

Үр дүн:

```

PS D:\00P\lab2> g++ 4.cpp
PS D:\00P\lab2> ./a
10 34
34 10

```

V. Заалтан хувьсагч (функцын параметер нь заалт байна) хоёр хувьсагчийн утгыг солих хэрэглэгчийн функц бич

```
void swap(int &a, int &b)
```

Параметр нь заалтан хувьсагч байх swap функц зарлаад дотор нь хувьсагчуудынхаа утгыг солих үйлдлүүдийг бичнэ.

Үр дүн:

```

PS D:\00P\lab2> g++ 5.cpp
PS D:\00P\lab2> ./a
20 17
17 20

```

## 5. ДҮГНЭЛТ

Энэ лабораторын ажлаар санах ой, түүний хаяг, хаяган хувьсагч, төрөл, багтаамж, new оператор болон санах ойн цоорхойн талаар судаллаа. Хувьсагчийн төрөл хөрвүүлэлтийг зөв ашиглахгүй бол логикийн том алдаа гардаг. Санах ойн цоорхойг ашиглан компьютерийн вирус хийх ч боломжтой гэдгийг мэдлээ.

## 6. АШИГЛАСАН МАТЕРИАЛ

- ★ “Объект хандлагат программчлал лекцийн материал”, М.Золжаргал
- ★ <https://www.geeksforgeeks.org/new-and-delete-operators-in-cpp-for-dynamic-memory/>

## 7. ХАВСРАЛТ

7-р бодлогоны код:

```
#include <iostream>
using namespace std;

void swap(int *a, int *b){ /*Хаяган хувьсагч ашигласан хоёр хувьсагчийн
утга солих функцээ зарлана*/
    int temp = *a; /*Шинэ хувьсагч зарлан а-ийн хаягийг хадгална*/
    *a = *b; /*а-ийн хаяган хувьсагчид b-ийн хаяган утгыг онооно*/
    *b = temp; /*b-ийн утгад temp-д хадгалсан а-ийн утгыг онооно.
Ингэснээр хоёр хувьсагчийн утга солигдоно.*/
}

int main(){
    int a, b; /*Хоёр тооныхоо хувьсагчийг зарлаж санах ой нөөцлөнө*/
    cin >> a >> b; /*Гараас 2 тооныхоо утгыг уншиж нөөцөлсөн санах ойдoo
утгуудаа онооно*/
    swap(&a, &b); /*Хаяган хувьсагч ашигласан функцээ дуудаж ажиллуулна*/
    cout << a << " " << b; /*Солигдсон утгуудыг хэвлэнэ*/
}
```

8-р бодлогоны код:

```
void swap(int &a, int &b){ /*Заалтан хувьсагч ашигласан хоёр хувьсагчийн
утга солих функцээ зарлана*/
    int temp; /*Шинэ хувьсагч зарлана*/
    temp = a; /*а-ийн хаягийг хадгална*/
    a = b; /*а-ийн заалтан хувьсагчид b-ийн утгыг онооно*/
}
```

```
    b = temp; /*b-ийн утгад temp-д хадгалсан *a-ийн утгыг онооно.  
    Ингэснээр хоёр хувьсагчийн утга солигдоно.*/  
}  
  
int main(){  
    int a, b; /*Хоёр тооныхоо хувьсагчийг зарлаж санах ой нөөцлөнө*/  
    cin >> a >> b; /*Гараас 2 тооныхоо утгыг уншиж нөөцөлсөн санах ойдoo  
утгуудаа онооно*/  
    swap(a, b);/*Заалтан хувьсагч ашигласан функцээ дуудаж ажиллуулна*/  
    cout << a << " " << b; /*Солигдсон утгуудыг хэвлэнэ*/  
}
```