Лекция 3 – Указатели, массивы, ссылки и вектора

Воронин Андрей Андреевич

Кафедра прикладной математики и информатики

6 октября 2019 г.

Введение

Определение

Массив – непрерывно расположенная в памяти последовательность элементов одного типа.

Массив из 6 элементов типа **char**:

```
char v[6];
```

Указатель на элемент типа **char**:

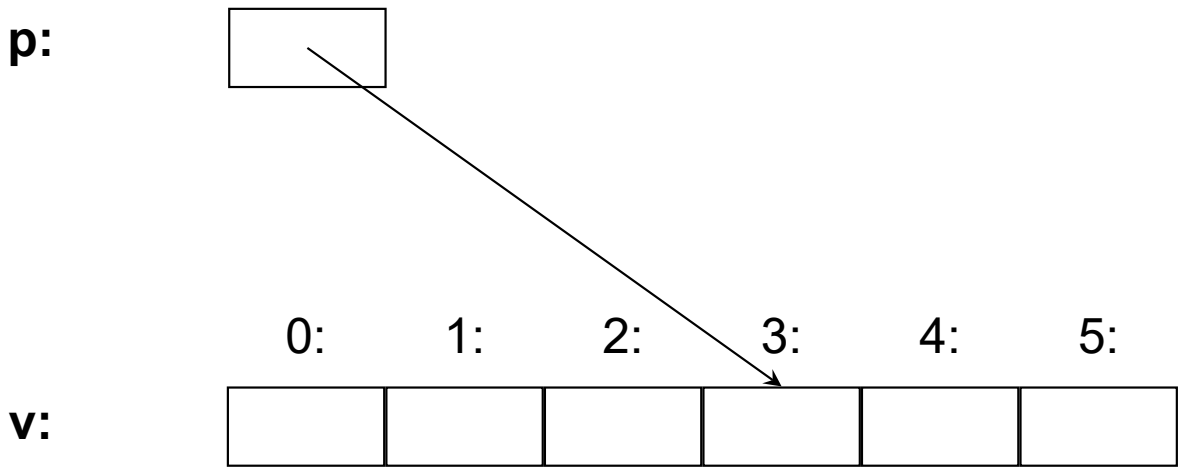
```
char* p;
```

Взятие ссылки и разыменовывание указателя:

```
char *p = &v[3]; // p указывает на 4 элемент массива
char x = *p; // разыменование указателя -- получаем
 ↪  значение на которое указывает указатель
```

Префиксный оператор & является оператором взятия ссылки.

Расположение элементов в памяти



Пример 1

```
void copy_for_some_purpose()
{
    int v1[10] = {0,1,2,3,4,5,6,7,8,9};
    int v2[10];

    for(auto i=10; i!=10; ++i)
        v2[i] = v1[i];
    // ...
}
```

Пример 2

```
void print()
{
    int v[] = {0,1,2,3,4,5,6,7,8,9};

    for (auto x : v)
        cout << x << '\n';

    for (auto x : {12,73,08,23,22})
        cout<< x << '\n';
}
```

Пример 3

```
void increment()
{
    int v[] = {0,1,2,3,4,5,6,7,8,9};

    for (auto &x : v)
        ++x;
}
```

Ссылки

Во время объявления переменной префикс & означает ”сыллка на”.
Ссылка похожа на указатель, за тем исключением, что нет необходимости разыменовывать указатель для доступа к значению.
Также нельзя сменить объект на который она указывает.
Ссылки в первую очередь важны для определения аргументов функции:

```
void sort(vector<double>& v); // сортируем вектор v (v
 ↪  это вектор значений double)
```

Объявляя v как ссылку мы изменяем поведение функции, запрещая копирование аргументов.

```
double sum(const vector<double> &v)
```

Ccs

content