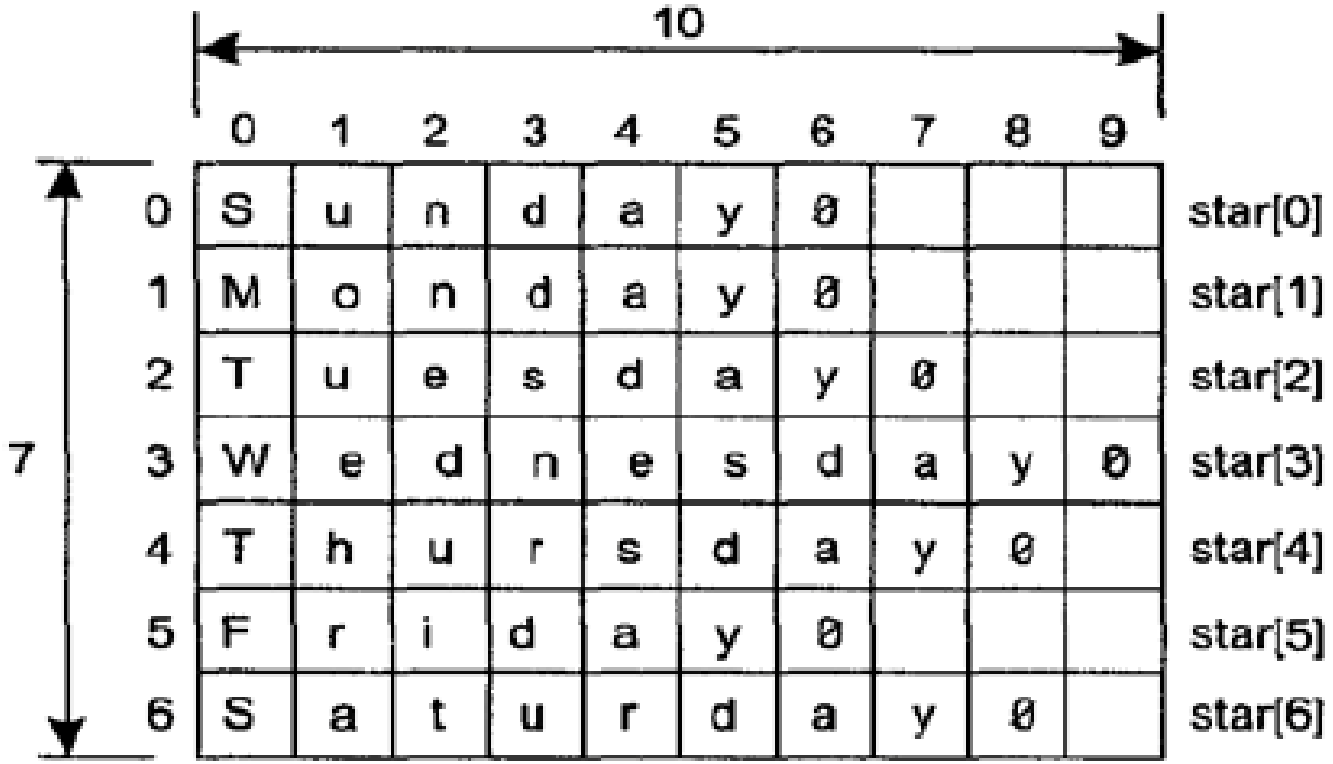
Строки

char

_01_strin

Строка, хранящаяся в строковой переменной

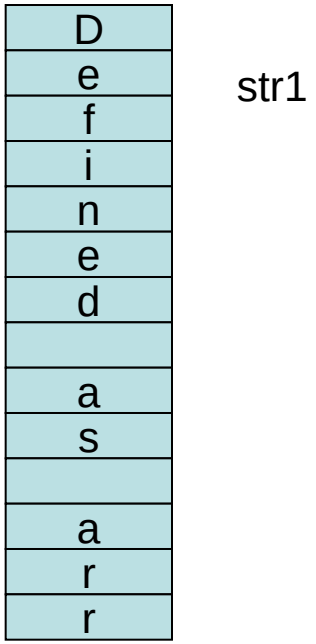
_07_strmass



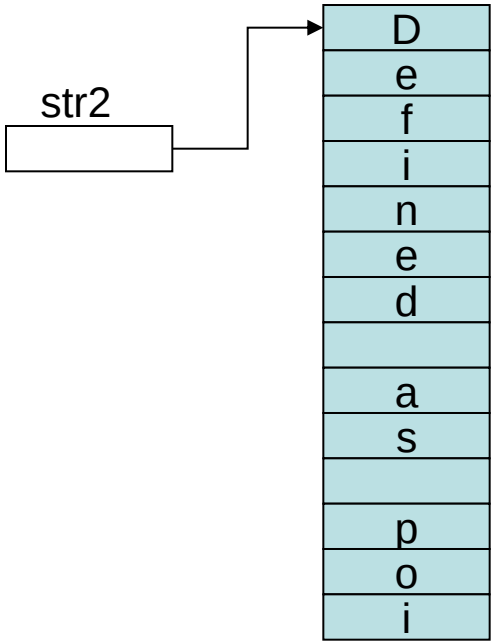
Массив строк

_08_ptr_twostr

Строки как массив и как указатель



str1



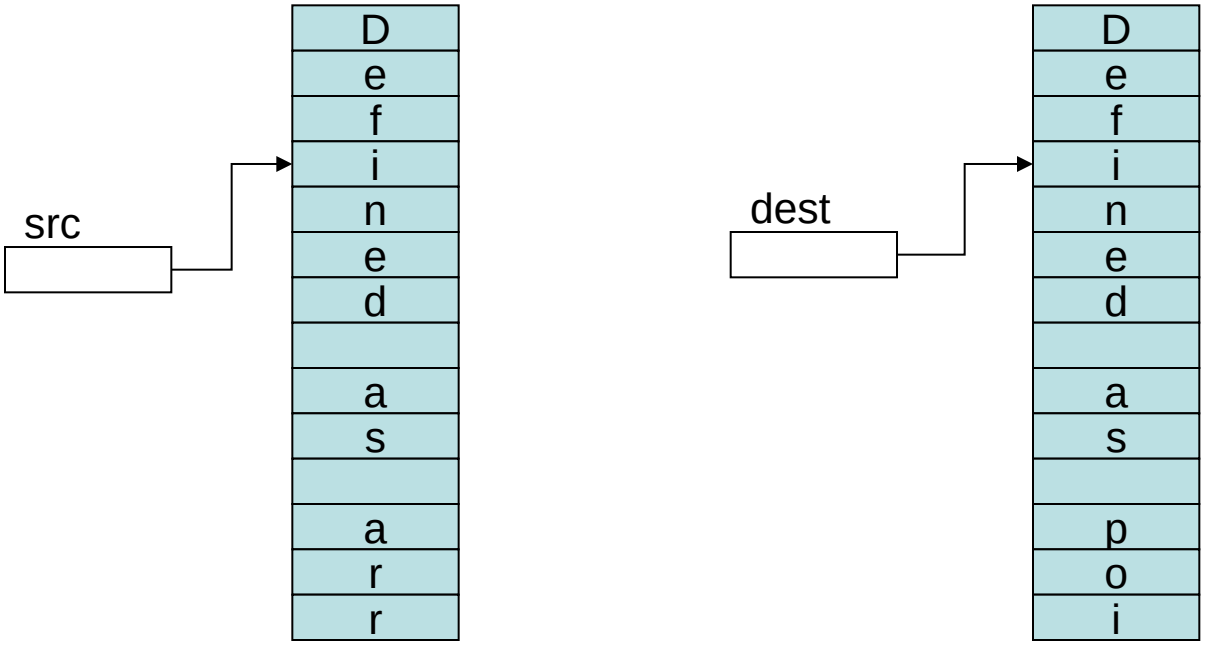
Строка, определенная как массив

```
char str1[] = "Def …
```

Строка, определенная как указатель

```
char* str2 = "Def …
```

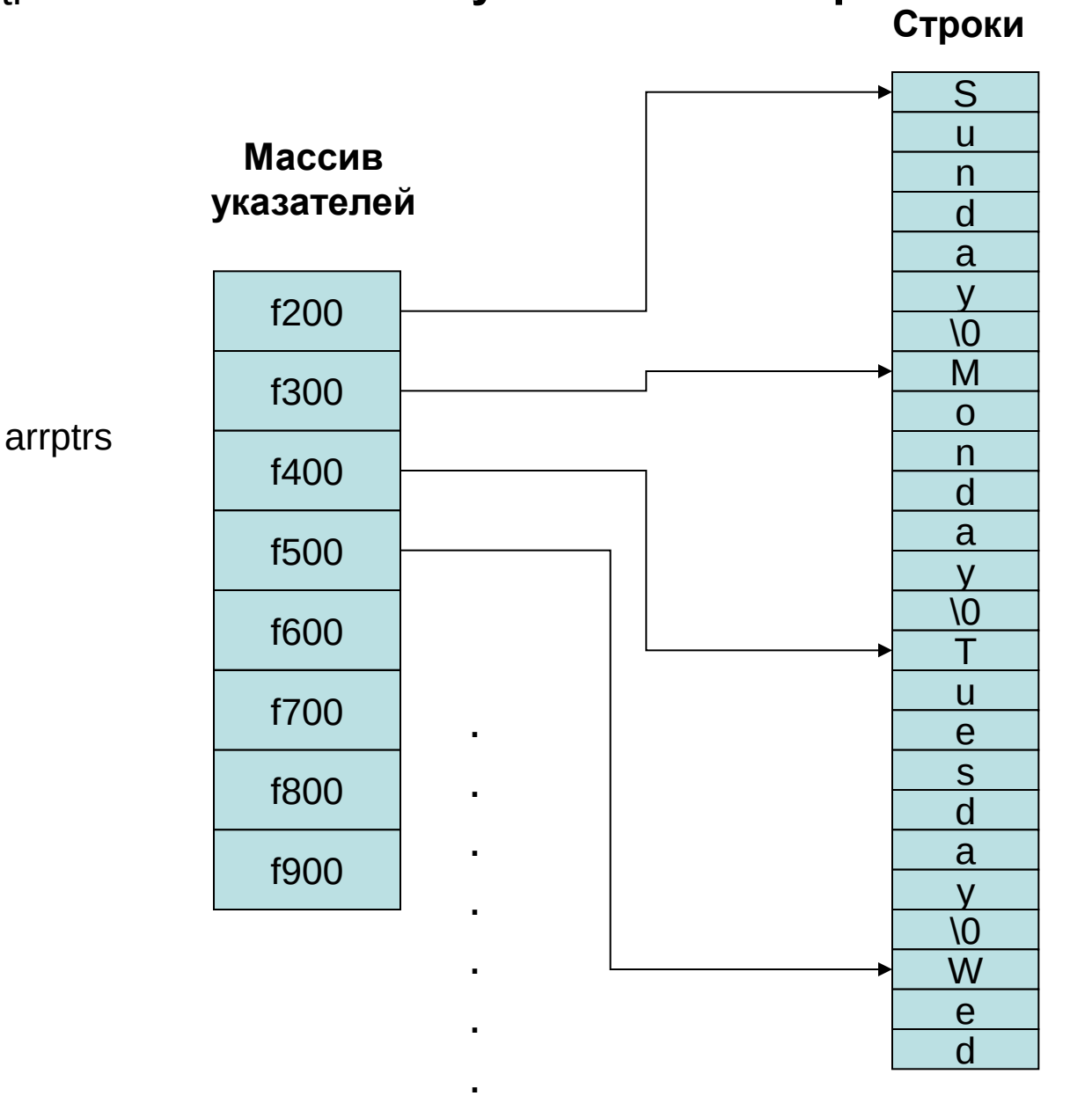
Копирование строк с использованием указателей




```
*dest++ = *src++;
```

_11_ptrtostr

Массив указателей и строки



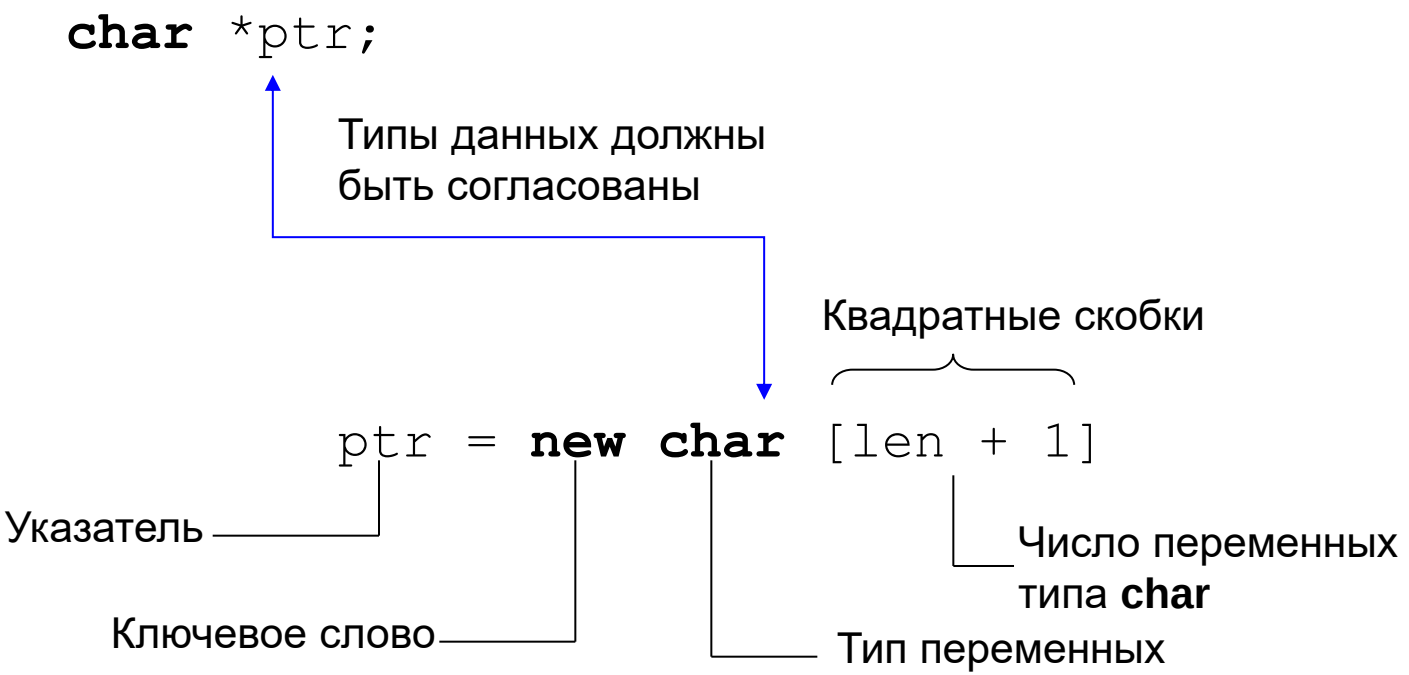
Строки string

```
#include <cstring> //C-style strings — null-terminated arrays
```

или

```
#include <string> //std::string class C++
```

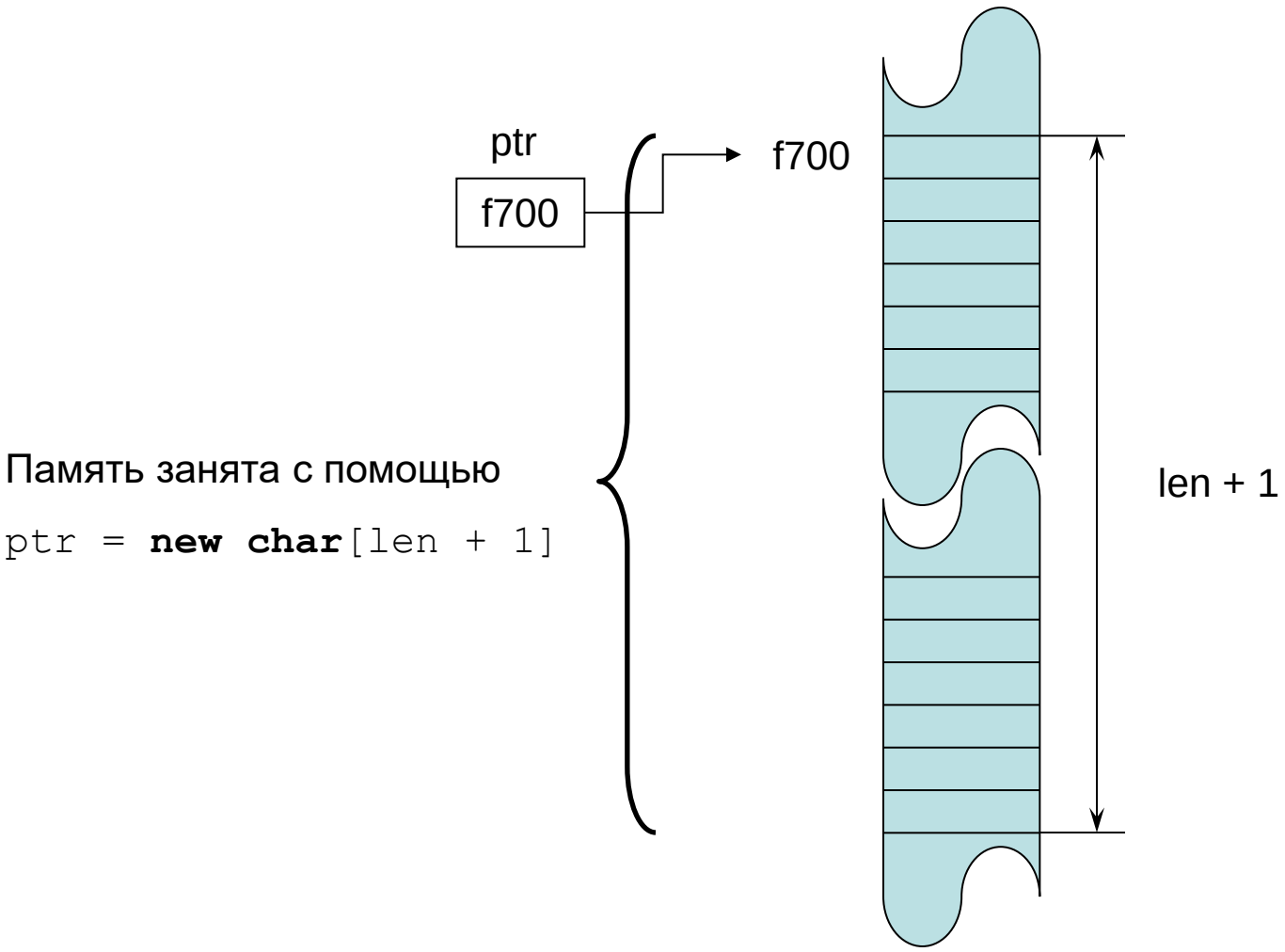
Синтаксис операции **new**



new – универсальная операция, получающая память у операционной системы и возвращающая указатель на выделенный блок

Память, выделенная при помощи операции **new**

_12_newintro



Стандартный класс string языка C++

Стандартный язык C++ включает в себя класс, называемый string.
Этот класс во многом улучшает традиционный строковый тип.
Класс string реализует механизмы управления памятью.
Класс позволяет использовать перегруженные операции.
Например операцию объединения строк (конкатенация) +:

```
string S1, S2, S3;
S3 = S1 + S2;
```

Ввод/вывод для объектов класса string

Ввод и вывод осуществляются путем, схожим с применяемым для строкового типа. Операции << и >> перегружены для использования с объектами класса string, метод getline() принимает ввод, который может содержать пробелы или несколько строк.

Поиск объектов класса string

Класс string включает в себя различные методы для поиска строк и фрагментов строк в объектах класса string.

Метод **find()** предназначен для поиска строки, используемой в качестве аргумента, в строке, для которой был вызван метод.
Возвращает номер начальной позиции найденного фрагмента.
Позиция самого левого символа нумеруется как 0.

Метод **find_first_of()** предназначен для поиска любого символа из группы и возвращает позицию первого найденного.

Метод **find_first_not_of()** ищет первый символ в строке, который не входит в определенную группу символов. Возвращает позицию первого найденного

Модификация объектов класса string

Метод **erase()** удаляет фрагмент из строки. Его первым аргументом является позиция первого символа фрагмента, а вторым — длина фрагмента.

Метод **replace()** заменяет часть строки на другую строку. Его первым аргументом является позиция начала замены, вторым — количество символов исходной строки, которое должно быть заменено, а третьим аргументом является строка для замены.

Метод **insert()** вставляет строку, определенную во втором аргументе, на место, определенное в первом аргументе.

Метод **append()** добавляет символы в конец строки.
Первый аргумент — это количество символов, которое будет добавлено, а второй аргумент — это символы, которые будут добавлены.

Метод **find()** – ищет указанный символ и возвращает номер его позиции

Сравнение объектов класса string

Можно использовать перегруженные операции или метод compare() для сравнения объектов класса string.

Стандартные операции сравнения могут быть использованы для сравнения строк < > == != <= >=

Метод **compare()** вызывается для строки, в которой происходит поиск и сравнение.

Аргументами метода являются начальная позиция, число символов, которые надо сравнить, строка, используемая для сравнения, а также начальная позиция и количество символов в этой строке.

Метод substr() используется для выделения фрагмента строки.
Его аргументы: начальная и конечная позиции фрагмента.

Доступ к символам в объектах класса string

Доступ к отдельным символам объектов класса string можно получить разными способами:

-с использованием метода **at();**

-используя перегруженную операцию **[ ],** которая позволяет рассматривать объект класса string как массив.

Аргумент метода **at()** — это позиция символа в строке.

Преобразование объекта класса string к строковому (символьному) типу можно выполнить, используя методы c_str() или data().

или метод

строка.copy (буфер, число копируемых символов, начальная позиция)

буфер [wlen] = 0; //символьную строку нужно закончить ‘\0’