

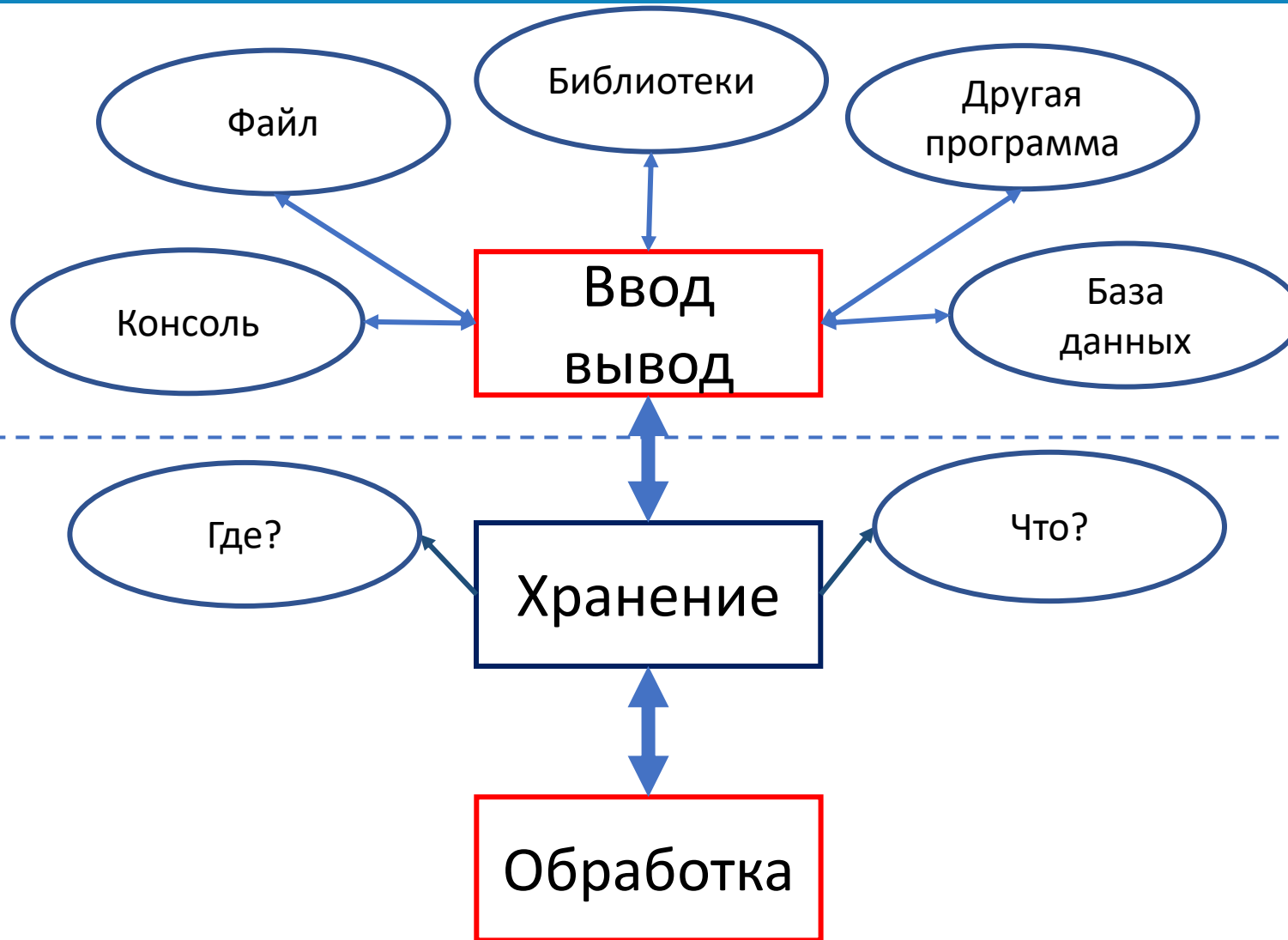


Основы программирования на C++

Занятие 7. Строки, структуры, работа с файлами



Дерево языка





В предыдущей лекции...

Стек

Память распределяется по методу
(LIFO)

Нет необходимости
освобождения памяти

Размер: **1 МБ** (по
умолчанию Windows)

Куча

Память распределяется в
случайном порядке

Необходимо освободить
память

Размер: **4 ГБ и более**



В предыдущей лекции...

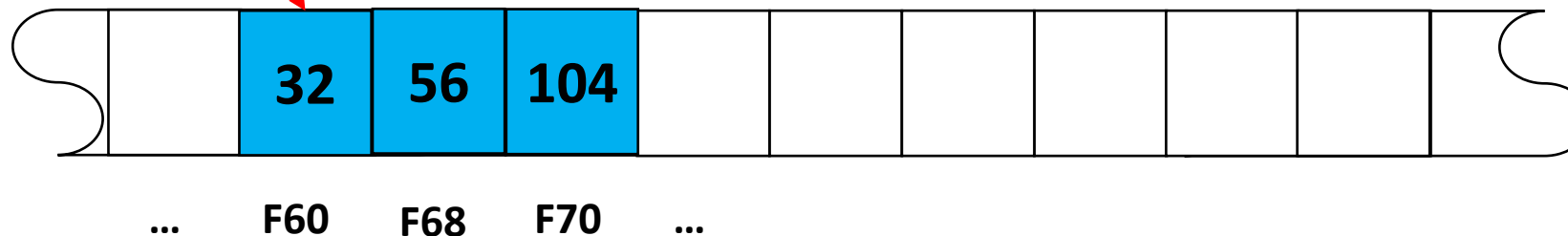
```
double* arr = (double*)malloc(N * sizeof(double));
```

```
&arr =          00000037405CFA20  
&arr[0] =       0000023344359F60  
&arr[1] =       0000023344359F68  
&arr[2] =       0000023344359F70
```

Стек



Куча





В предыдущей лекции...

```
#include <stdio.h>
#include <stdlib.h>

const int N = 10;

void input_array(double *x)
{
    int arrMax = 100, arrMin = 0;
    for (int i = 0; i < N; i++)
        x[i] = arrMin + (arrMax - arrMin) * ((double) rand() / RAND_MAX);
}

void print_array(double* x)
{
    for (int i = 0; i < N; i++)
        printf("%lf ", x[i]);
    printf("\n ");
}

int main()
{
    double* array_heap = (double*)malloc(N * sizeof(double));
    input_array(array_heap);
    print_array(array_heap);
    free(array_heap);
    return 0;
}
```



Проблема

Необходимо разработать программу для хранения и обработки домашней библиотеки.





Вопросы

1. Как хранить названия книг?
2. Как лучше хранить список книг?
3. Как сохранять список книг?



Строки в Си

Строк в Си НЕТ



Строки == Массив символов

```
#include <stdio.h>
```

```
const int N = 10;
```

```
int main()
```

```
{
```

```
    char name[N];
```

```
    printf("Enter name: ");
```

```
    scanf_s("%s", name, N);
```

```
    printf("Your name is %s.", name);
```

```
    return 0;
```

```
}
```

```
Enter name: SAB  
Your name is SAB.
```



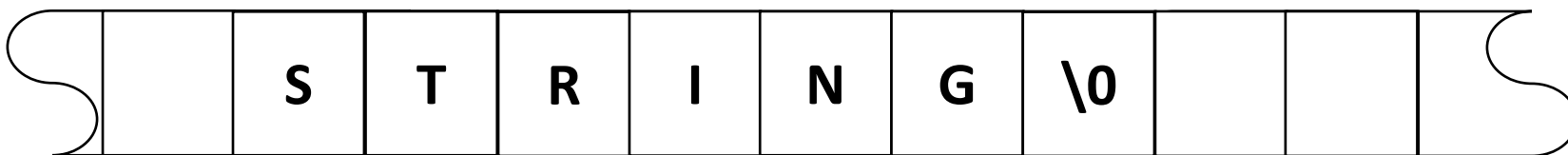
```
Enter string: SAB
String is SAB.
String length is 3

Symbol      Code
S            0x53
A            0x41
B            0x42
             0x0
█            0xfffffffffe
█            0xfffffffffe
█            0xfffffffffe
█            0xfffffffffe
█            0xfffffffffe
█            0xfffffffffe
```

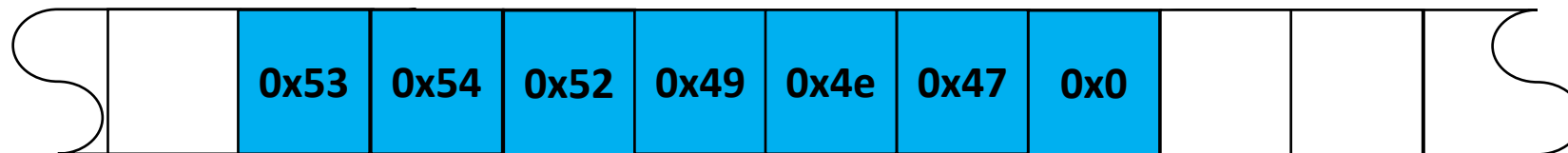


Как хранится строка в памяти

Символы



Коды символов





Полезные функции

```
#include <string.h>
```

	Синтаксис	Описание
1	<code>size_t *strlen (const char *str);</code>	Функция <code>strlen</code> вычисляет количество символов в строке до первого вхождения символа конца строки.
2	<code>char *strcpy (char *destination, const char *source);</code>	Функция <code>strcpy</code> копирует данные из строки, на которую указывает аргумент <code>source</code> , в строку, на которую указывает аргумент <code>destination</code> , пока не встретится символ конца строки (нулевой символ).
3	<code>int strcmp (const char *str1, const char *str2);</code>	Функция сравнения строк. Возвращает 0 – если сравниваемые строки идентичны.



Сравнение строк

```
#include <stdio.h>
#include <string.h>

const int N = 10;

int main()
{
    char str1[N];
    char str2[N];
    scanf_s("%s", str1, N);
    scanf_s("%s", str2, N);
    printf("String is %s\n", str1);
    printf("String is %s\n", str2);
    int res = strcmp(str1, str2); // strings compare
    printf("Res = %d\n", res);
    return 0;
}
```

str1 < str2 res = -1

str1 > str2 res = 1

str1 = str2 res = 0



Копирование строк

```
#include <stdio.h>
#include <string.h>

const int N = 10;

int main()
{
    char str1[N];
    char str2[N];
    scanf_s("%s", str1, N);
    printf("String 1 is %s\n", str1);
    strcpy_s(str2, str1); // strings copy str1 to str2
    printf("String 2 is %s\n", str2);
    return 0;
}
```



Структуры

```
const int N = 10;
const int N_students = 2;

char name[N_students][N];
char surname[N_students][N];
int grade[N_students];

for (int i = 0; i < N_students; ++i){
    printf("Name: ");
    scanf_s("%s", name[i], N);
    printf("Surname: ");
    scanf_s("%s", surname[i], N);
    printf("Grade: ");
    scanf_s("%d", &grade[i]);
}
for (int i = 0; i < N_students; ++i){
    printf("Name = %s\n", name[i]);
    printf("Surname = %s\n", surname[i]);
    printf("Grade = %d\n", grade[i]);
}
```

Как лучше хранить список объектов состоящие из нескольких частей?

Полный код см. в заметках к слайду



Структуры

```
struct Student
```

```
{
```

```
    char name[N];
```

```
    char surname[N];
```

```
    int grade;
```

```
};
```



Структура



Поля структуры

```
struct Student student[N_students];
```



Объявление

```
student[i].name
```

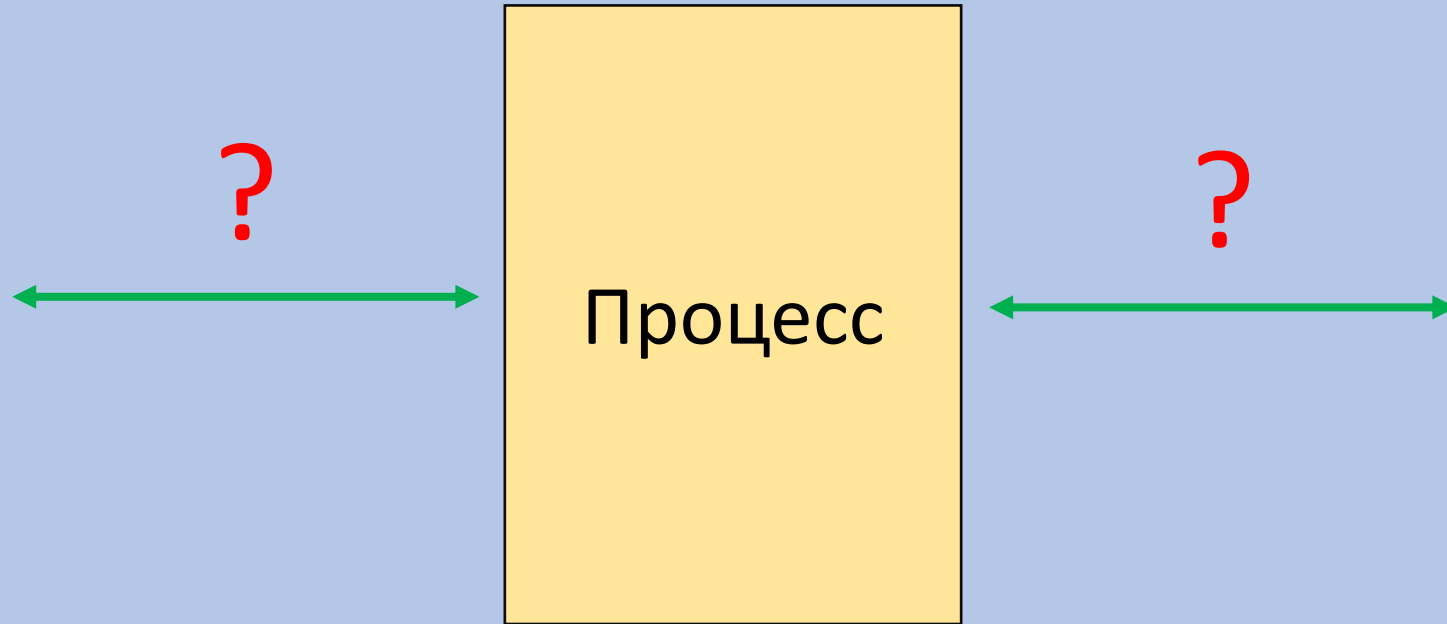


Обращение к полю



Работа с файлами

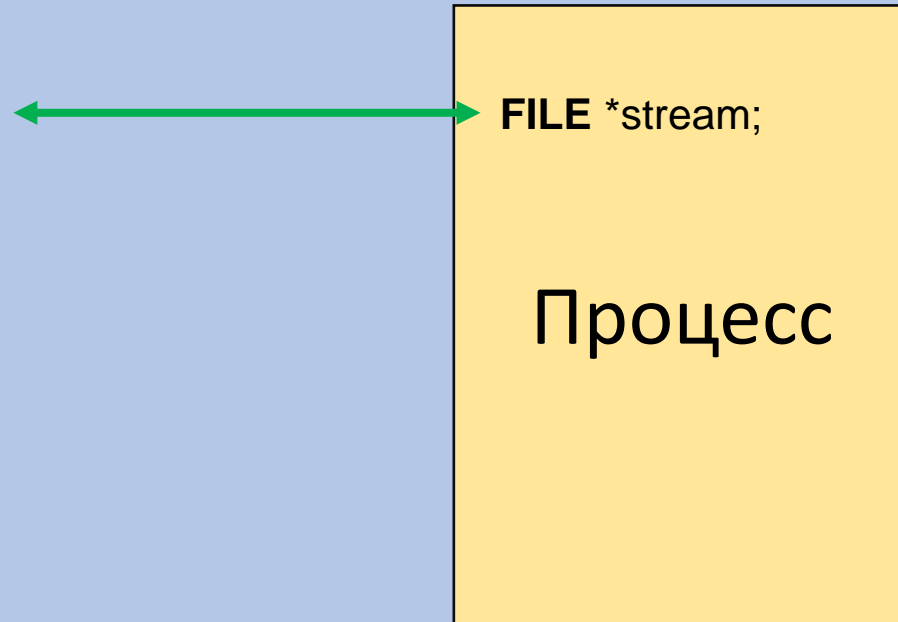
OS





Работа с файлами

OS





Основные функции работы с файлами

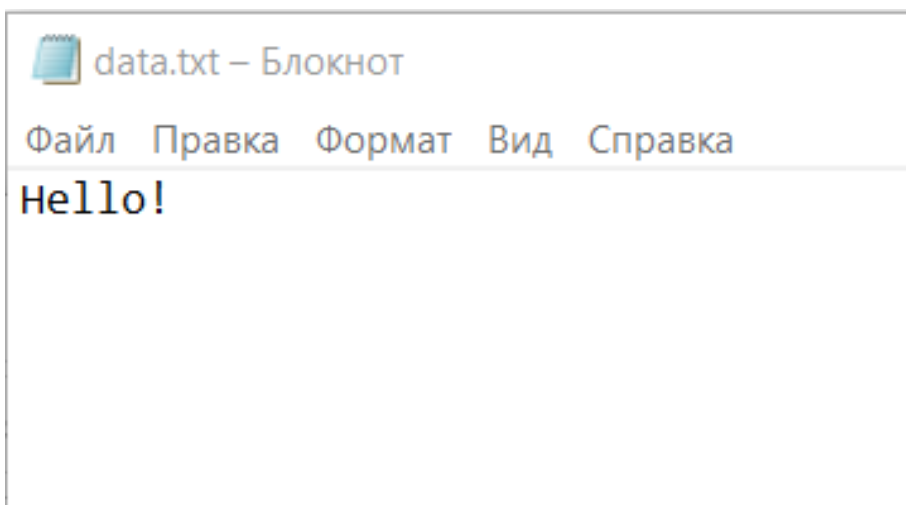
Описание	Синтаксис	Комментарий
Открытие файла	<pre>errno_t fopen_s(FILE** pFile, const char* filename, const char* mode);</pre>	Модификаторы: "r" Открывает для чтения "w" Открывает для перезаписи "a" Открывает для записи в конец файла
Запись в файл	<pre>int fprintf(FILE *stream, const char *format [, argument]...);</pre>	Если stream= stdout , то вывод будет производиться на экран
Чтение из файла	<pre>char *fgets(char *str, int numChars, FILE *stream);</pre>	Считывание возможно проводить в цикле до момента, когда функция вернет NULL <i>while ((fgets(c, 3, fp)) != NULL)</i> Считывает numChars – 1 символ
Закрытие файла	<pre>int fclose(FILE *stream);</pre>	



Работа с файлами

```
#include <stdio.h>

int main(void)
{
    FILE* fp;
    fopen_s(&fp, "data.txt", "w");
    fprintf(fp, "%s", "Hello!");
    fclose(fp);
    return 0;
}
```





Работа с файлами

```
#include <stdio.h>
```

```
int main(void)
{
    FILE* fp;
    char sym[10];
    fopen_s(&fp, "data.txt", "w");
    fprintf(fp, "%s", "Hello world!");
    fclose(fp);

    fopen_s(&fp, "data.txt", "r");
    while ((fgets(sym, 10, fp)) != NULL)
    {
        printf("%s\n", sym);
    }
    fclose(fp);

    return 0;
}
```



data.txt – Блокнот

Файл Правка Формат Вид Справка

Hello world!

Hello wor
ld!



Работа с файлами

```
if ((fopen_s(&fp, filename, "w")) != NULL)
{
    perror("Error");
    return 1;
}
```

При работе с файлами
рекомендуется проводить
проверку на корректное
открытие

Полный код см. в заметках к слайду



Перемещение указателя

`fseek(указатель, смещение, точка отсчета)`

```
fseek(fp, 0, SEEK_SET); //перевод указателя на начало файла  
fseek(fp, 0, SEEK_END); //перевод указателя на конец файла
```



Домашнее задание 7-8

Доработать программу калькулятор следующим образом:

1. Добавить пользователям возможность вводить текстовые команды для управления калькулятором
2. Добавить возможность сохранения результатов вычисления простых арифметических действий в файл
3. Добавить возможность расчета сложения, вычитания и вычисления длины векторов. Использовать структуру.
(http://mathprofi.ru/vektory_dlya_chainikov.html)
4. *Добавить возможность считывания матриц из файла