Основы программирования на C++

Занятие 7. Строки, структуры, работа с файлами

Дерево языка

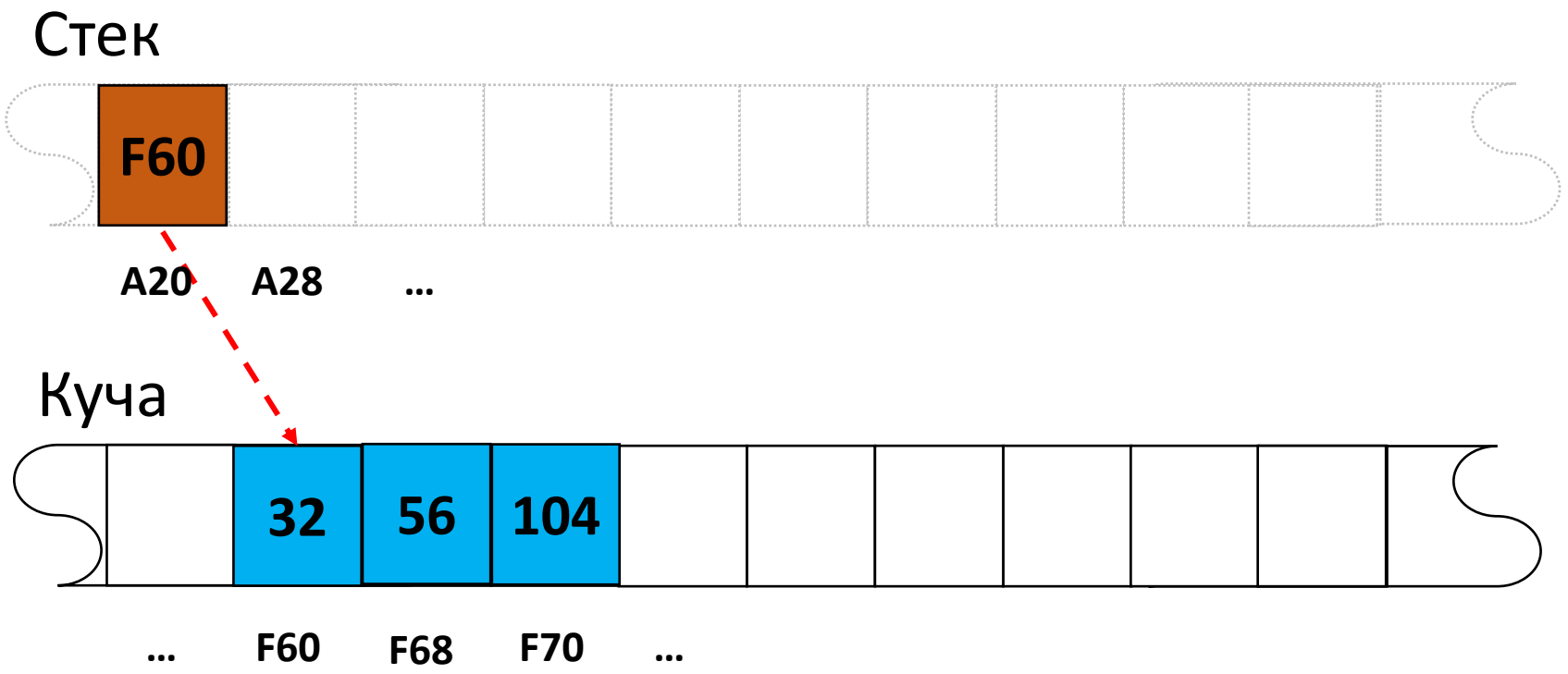
В предыдущей лекции…

Стек	**Куча**
Память распределяется по методу (LIFO)	Память распределяется в случайном порядке
Нет необходимости освобождения памяти	Необходимо освобождать память
Размер: **1 МБ** (по умолчанию Windows)	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
```

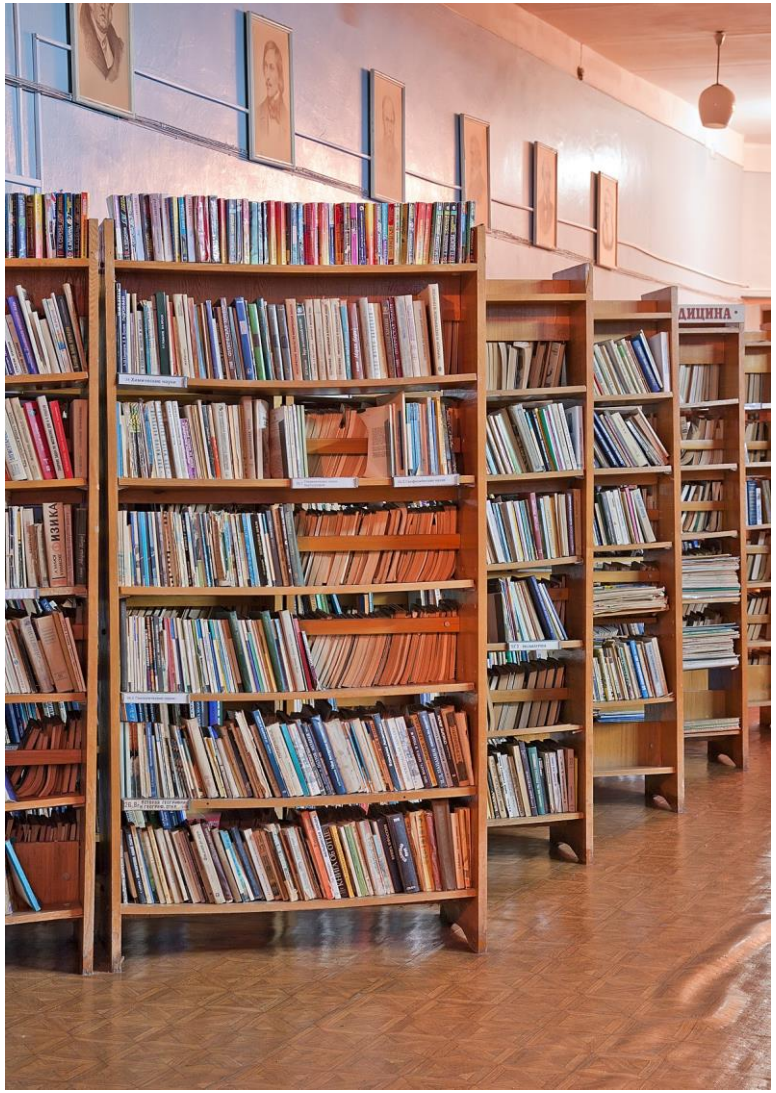


В предыдущей лекции…

```c
#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 в Си

Строки == Массив символов

```c
#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.
```

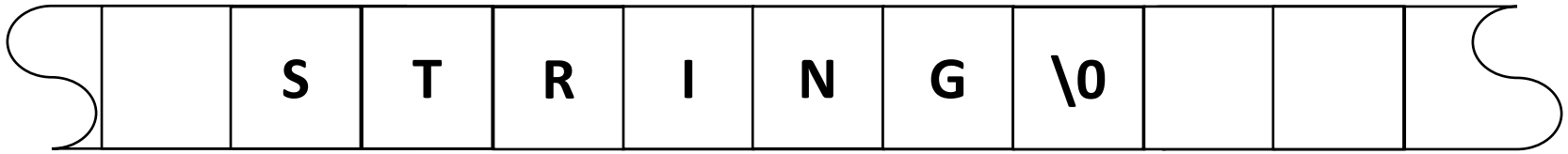
Строки в Си

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

const int N = 10;

int main()
{
    char name[N];
    printf("Enter string: ");
    scanf_s("%s", name, N);
    printf("String is %s.\n", name);
    printf("String length is %zu\n", strlen(name));
    printf("\nSymbol\tCode\n");
    for (int i = 0; i < N; i++)
        printf("%c\t0x%x\n", name[i], name[i]);
    return 0;
}
```

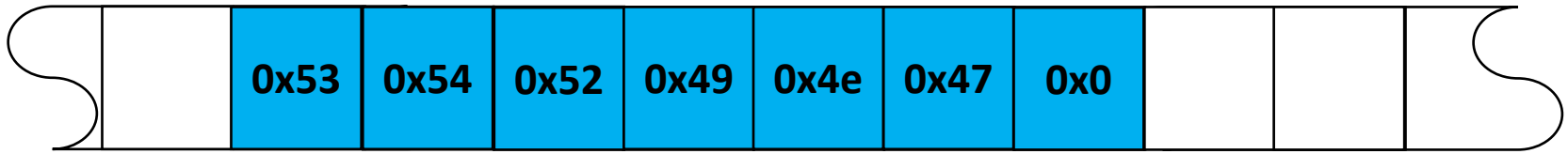
```
Enter string: SAB
String is SAB.
String length is 3

Symbol  Code
S       0x53
A       0x41
B       0x42
        0x0
■       0xfffffffe
■       0xfffffffe
■       0xfffffffe
■       0xfffffffe
■       0xfffffffe
■       0xfffffffe
■       0xfffffffe
```

Как хранится строка в памяти

Символы



Коды символов



Полезные функции

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

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

Сравнение строк

```c
#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

```c
#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];
```

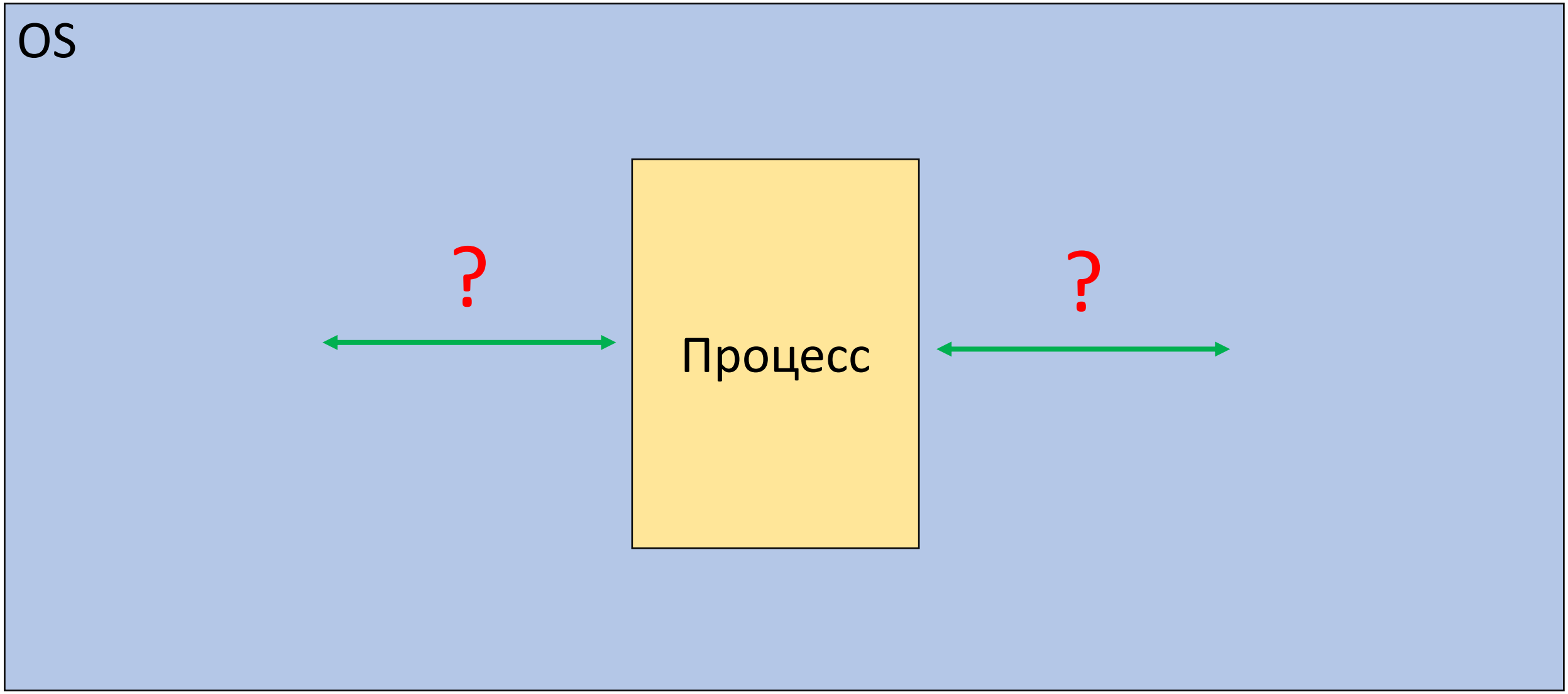
Объявление

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

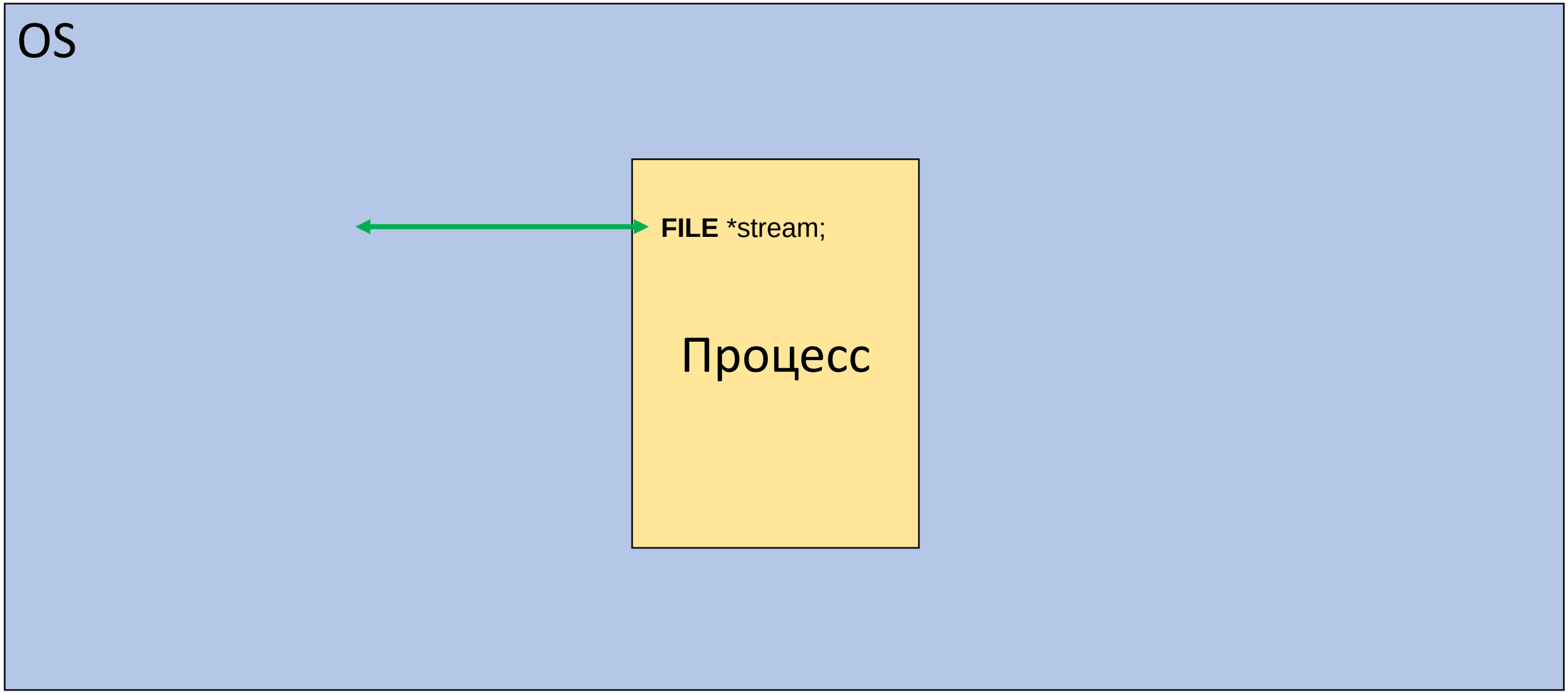
Обращение к полю

Полный код см. в заметках к слайду

Работа с файлами



Работа с файлами

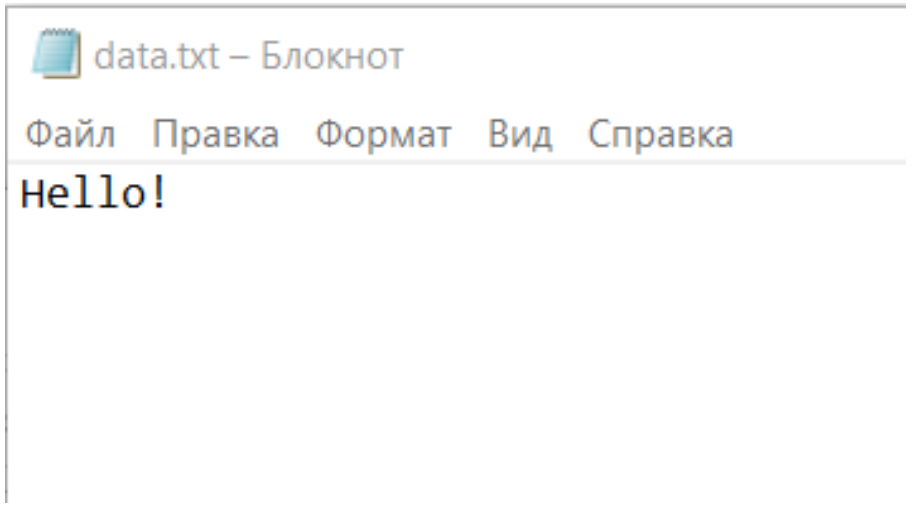


Основные функции работы с файлами

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

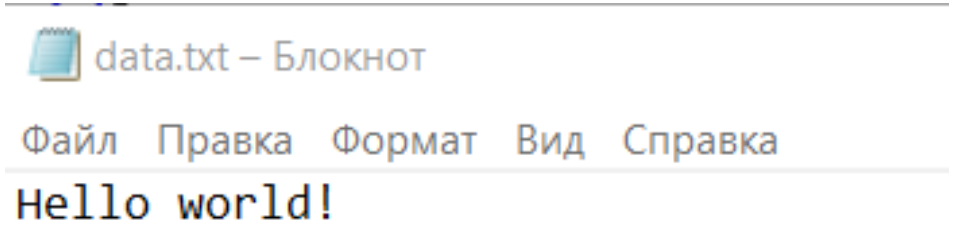
Работа с файлами

```c
#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

```c
#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;
}
```




```
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
```