

Семинар #14: Безопасность.

Проверка на ошибки. Переменная `errno` и функция `perror`.

```
#include <stdio.h>
int main()
{
    FILE* file = fopen("input.txt", "r");
    if (file == NULL)
    {
        perror("Error");
        exit(1);
    }
    // Работаем с файлом file
}
```

Опасность функций семейства `scanf` при считывании строк.

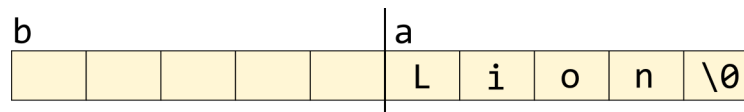
Функции `scanf`, `fscanf` и `sscanf` имеют одну неприятную особенность при считывании строк. Спецификатор `%s` делает следующее: считывает строку и записывает её всю по передаваемому ей адресу. Это может привести к серьёзным ошибкам если то место, куда мы записываем строку будет меньше, чем записываемая строка. Рассмотрим, например, следующую простую программу:

```
#include <stdio.h>
int main()
{
    char a[5] = "Lion";
    char b[5];

    scanf("%s", b);

    printf("a = %s\n", a);
    printf("b = %s\n", b);
}
```

- Что напечатает эта программа, если на вход передать строку `Cat`?
- Что напечатает эта программа, если на вход передать строку `Zebra`?
- Что напечатает эта программа, если на вход передать строку `Elephant`?
- Как исправить ошибки?



Функция `scanf` выходит за границы массива `b` и переписывает другую строку. Если бы на месте строки `a` была бы переменная другого типа, то `scanf` испортил бы и её.

Функции `ftell` и `fseek`