

The background is a dark, textured surface featuring a complex network of glowing white and blue lines that resemble a circuit board or neural network. Several bright blue circular nodes are scattered throughout the network. Faint, semi-transparent binary code (0s and 1s) is visible in the lower right area.

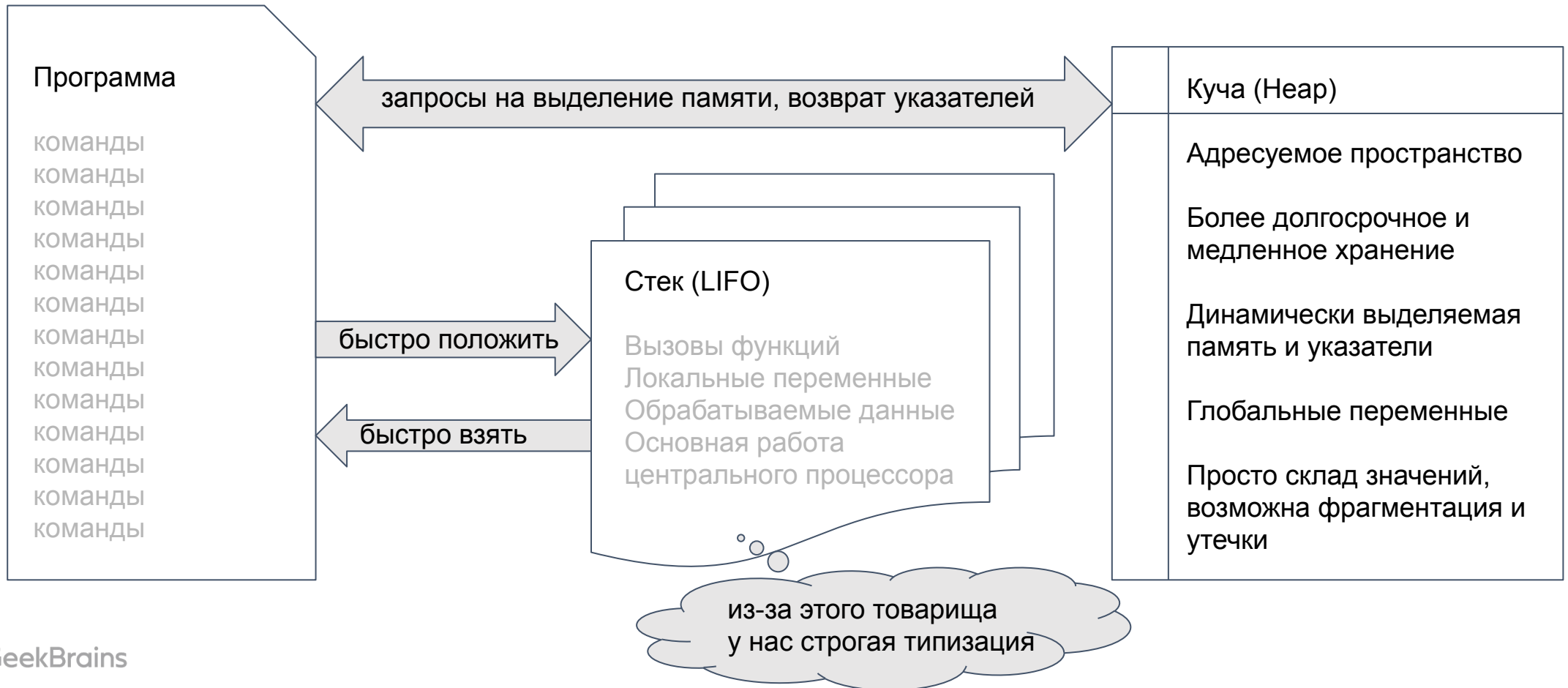
# Управление памятью

ОСНОВЫ C++

# Что будет на уроке

1. Узнаем, что представляет из себя процесс управления памятью, как выделять и освобождать динамическую память
2. Научимся не допускать утечек памяти и изучим операторы `new` и `delete`
3. Рассмотрим работу с файловой системой и потоками ввода-вывода

# Понятие стека и кучи



**void\* malloc(size)** - функция, принимающая на вход количество байт и запрашивающая сплошную никак не размеченную область памяти в куче, возвращающая неопределённый указатель. Выделенная память не очищается (нельзя достоверно знать, какие значения будут внутри).

**void\* calloc(items, size)** - функция, принимающая на вход количество значений и их размер, запрашивающая сплошную область памяти в куче, возвращающая неопределённый указатель. Выделенная память очищается (содержит нулевые значения).

**void free(void \*)** - функция, высвобождающая память, на которую указывает переданный указатель. Память возвращается под управление ОС и может быть использована для других целей.

- если память не освободить после использования - это называется **утечка памяти**
- если память не освободить - возникает **фрагментация памяти**

# **void\* realloc(pointer, size)**

Попытка изменить размер выделенной области;

Если не удалось, вернёт нулевой указатель;

Может выделить дополнительную память рядом с переданной областью (начальный адрес не изменится);

А может и нет (начальный адрес изменится);

Разные трансляторы реализуют эту функцию по-разному.

# Аргументы вызова программы

Параметры функции **main(int argc, char\*\* argv)**

**argc** - это количество переданных параметров

**argc** - всегда больше нуля

**argv** - это массив строк (аргументы разделены пробелом)

**argv[0]** - всегда имя программы

можно работать как с массивом и как с vararg

**Оператор new** - это с одной стороны упрощённая запись функций `calloc` и `malloc`, с другой стороны более удобный способ выделения памяти для объектов C++, для которого обладает всеми свойствами оператора

**Оператор delete** - это с одной стороны упрощённая запись функции `free`, с другой стороны - более удобный способ высвобождения памяти в C++. Является не функцией, а оператором языка



# Потоки ввода-вывода

- **stdin** - стандартный поток ввода информации в программу
- **stdout** - стандартный поток вывода из программы
- **stderr** - стандартный поток вывода информации об ошибках

Стандартные потоки ввода-вывода подключены к средствам ввода-вывода операционной системы (клавиатура, командная строка) и открываются и закрываются автоматически при старте и завершении программы соответственно.

# Базовые манипуляции с файлами

## Открытие

осуществляется по имени файла, имя файла можно задать как относительное (от расположения программы) так и абсолютное (от корня файловой системы)

## Запись

осуществляется аналогично использованию других потоков вывода и может быть как текстовая так и бинарная (зависит от режима открытия файла)

## Чтение

осуществляется аналогично использованию потоков ввода и обычно требует дополнительной обработки данных, из-за разницы форматов записи и кодировок

## Заккрытие

необходимо для того, чтобы вернуть файл под управление операционной системой, в противном случае другие программы не смогут его открыть