

# Unsafe, fixed, P/Invoke

Похабов Иван Ильич

28 ноября 2024 г.

## Зачем это нужно:

- ▶ Работа с кодом на C/C++
- ▶ Управление памятью напрямую для работы с оборудованием
- ▶ Обход проверок времени выполнения
- ▶ Производительность

## Потенциальные проблемы:

- ▶ Множество способов "прострелить себе ногу":
  - ▶ Утечки памяти
  - ▶ Выход за пределы массива
  - ▶ В общем всё, за что мы не любим C

# Unsafe

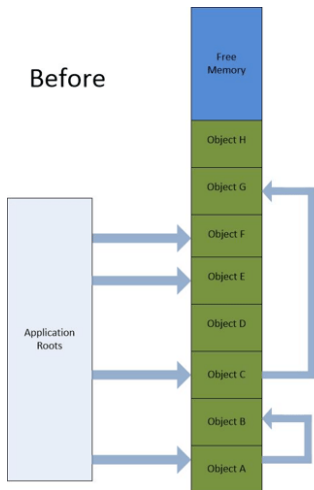
```
static unsafe void Copy(byte[] source, int sourceOffset, byte[]  
↪ target,  
    int targetOffset, int count) {  
    fixed (byte* pSource = source, pTarget = target) {  
        for (int i = 0; i < count; i++) {  
            pTarget[targetOffset + i] = pSource[sourceOffset +  
↪ i];  
        }  
    }  
}
```

# Сборщик мусора

Сборка мусора состоит из следующих этапов:

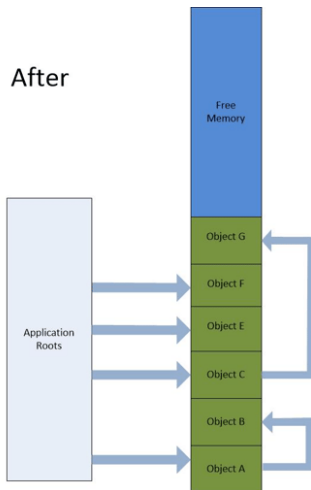
- ▶ Этап маркировки: поиск всех используемых объектов
- ▶ Этап перемещения: обновление ссылок на сжимаемые объекты
- ▶ Этап сжатия: освобождение памяти неиспользуемых объектов

Before



Garbage  
Collection

After



# Fixed

- ▶ Предотвращает перемещение объекта в ходе компактификации кучи
- ▶ Объявленный указатель можно использовать только в соответствующей `fixed` инструкции
- ▶ Объявленный указатель не может быть изменен

## Пример

```
unsafe {  
    int[] numbers = {10, 20, 30};  
    fixed (int* toFirst = &numbers[0], toLast = &numbers[^1]) {  
        Console.WriteLine(toLast - toFirst);  
    }  
}
```

- ▶ Технология для вызова функций из нативных библиотек в управляемом коде.
- ▶ Применения:
  - ▶ Доступ к нативному API
  - ▶ Интеграция с существующим кодом
  - ▶ Работа с устройствами
- ▶ Недостатки:
  - ▶ Сложность
  - ▶ Ошибки совместимости
  - ▶ Отсутствие переносимости



# Пример

```
using System;
using System.Runtime.InteropServices;

[LibraryImport("user32.dll",
StringMarshalling = StringMarshalling.Utf16,
SetLastError = true)]
private static partial int MessageBox(IntPtr hWnd,
string lpText, string lpCaption, uint uType);

public static void Main(string[] args) {
    MessageBox(IntPtr.Zero, "Message", "Attention!", 0);
}
```

# Пример

```
using System;
using System.Runtime.InteropServices;

private delegate bool EnumWC(IntPtr hwnd, IntPtr lParam);

[LibraryImport("user32.dll")]
private static partial int EnumWindows(EnumWC lpEnumFunc, IntPtr
↪ lParam);

private static bool OutputWindow(IntPtr hwnd, IntPtr lParam) {
    Console.WriteLine(hwnd.ToInt64());
    return true;
}

public static void Main(string[] args) {
    EnumWindows(OutputWindow, IntPtr.Zero);
}
```