Алгоритмы и Алгоритмические Языки

Семинар #9:

- Динамическое выделение памяти;
- Наивная реализация расширяющегося стека;
- Анализ схемы изменения размера;
- Более эффективная и надёжная реализация;

07.11.2023

Динамическое выделение памяти

Структура памяти программы

Стек (`RW-`) – для локальных переменных и стека вызовов, управляется компилятором.

Куча (`RW-`) – управляется программистом.

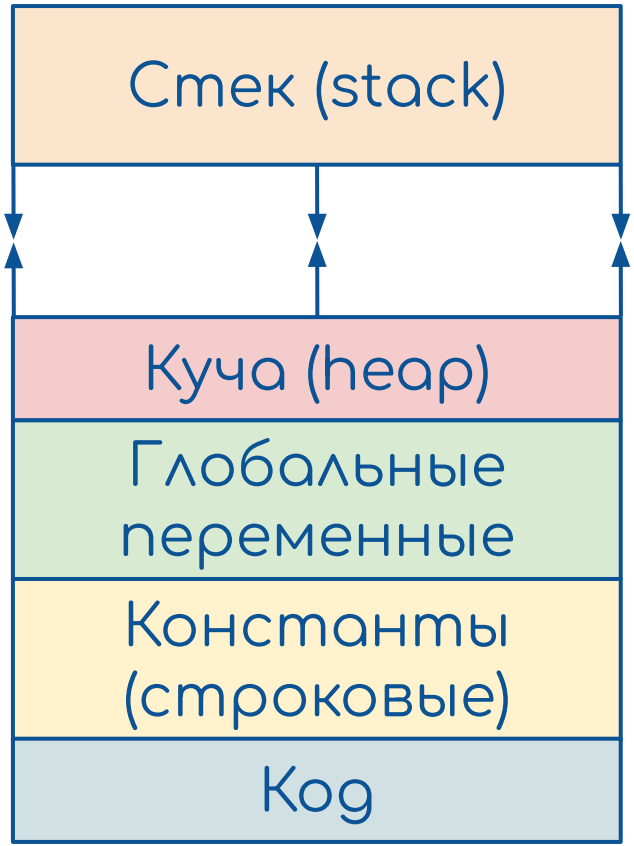
Глобальные переменные (`RW-`);

Константы и строковые литералы (`R--`);

Исполняемый код (`R-X`);

Динамически подгружаемые библиотеки;

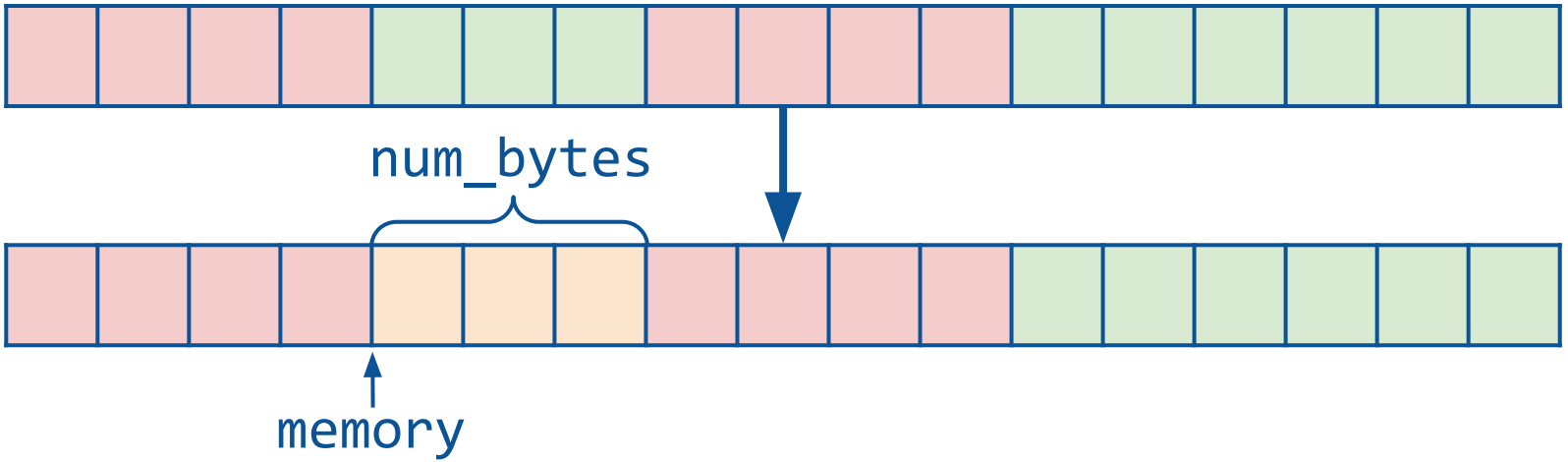
Многое другое...



Динамическое выделение памяти

Функция `malloc` выделяет свободную память на куче:

```
void* memory = malloc(num_bytes);
```

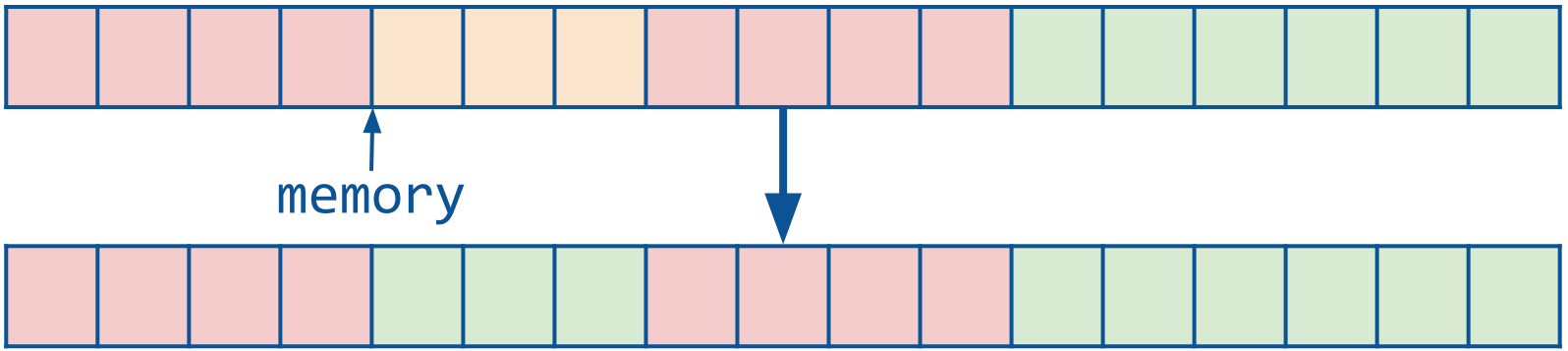


Как malloc находит свободную память?

Динамическое выделение памяти

Функция free освобождает занятую память на будущее:

```
free(memory);
```

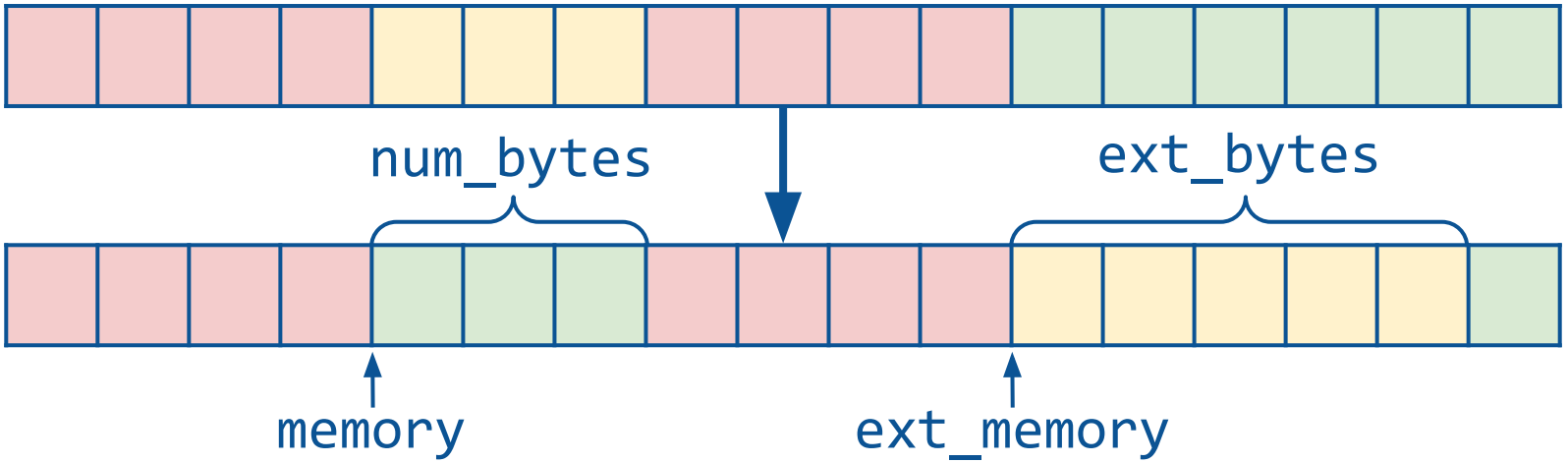


Как free узнаёт, сколько памяти освобождать?

Динамическое выделение памяти

Функция realloc изменяет размер выделенного блока памяти:

```
void* ext_memory = realloc(memory, ext_bytes);
```



Чем **realloc** может быть лучше **malloc+memcpy+free**?

Наивная реализация расширяющегося стека

Реализация структуры данных “Стек”

Реализации в репозитории курса: `09_resizing_stack`

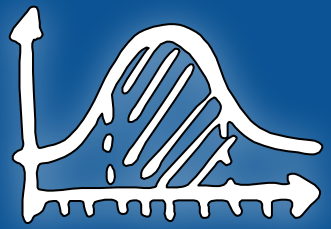
- Наивная реализация (`evil_stack.h`);

Особые моменты реализации:

- Подключение заголовочных файлов;
- Псевдо-полиморфизм структуры данных;
- Работа с динамической памятью:
  выделение, изменение размера, освобождение памяти;
- Операции со стеком: `push`/`pop`;
- Защита от `pop-from-empty-stack`.

Проблемы реализации?

Анализ схемы изменения размера



Увеличение ёмкости стека

Расширение ёмкости стека:

```
// Resize to provide space for new element:
if (stack->size == stack->capacity)
{
    stack_resize(stack, stack->capacity + 1U);
}
```

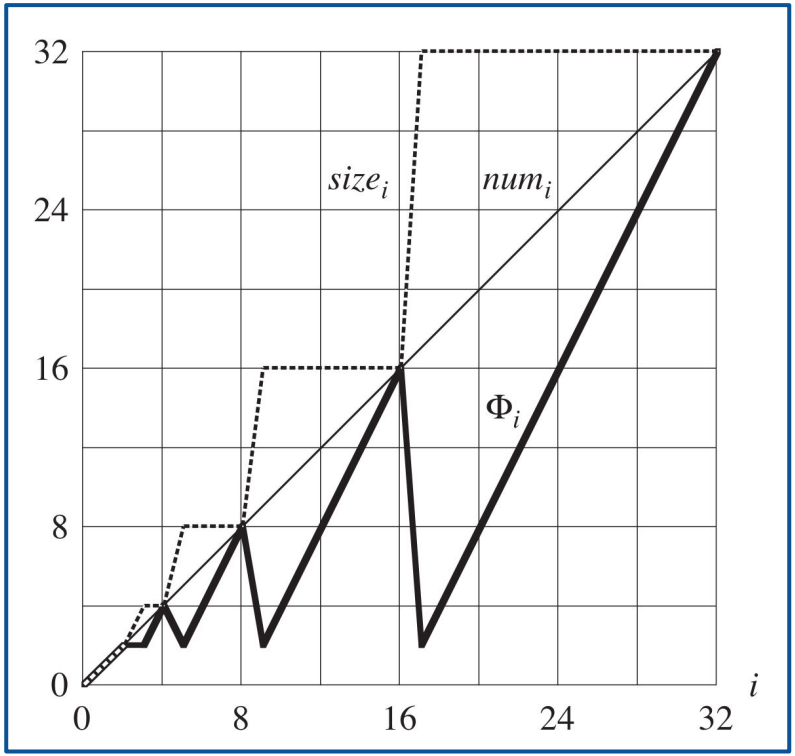
В худшем случае один `realloc` копирует всё содержимое стека.

В худшем случае: `n` `push`-ей за $O(n^2)$ операций с памятью!

Возможное решение?

Амортизация стоимости копирований

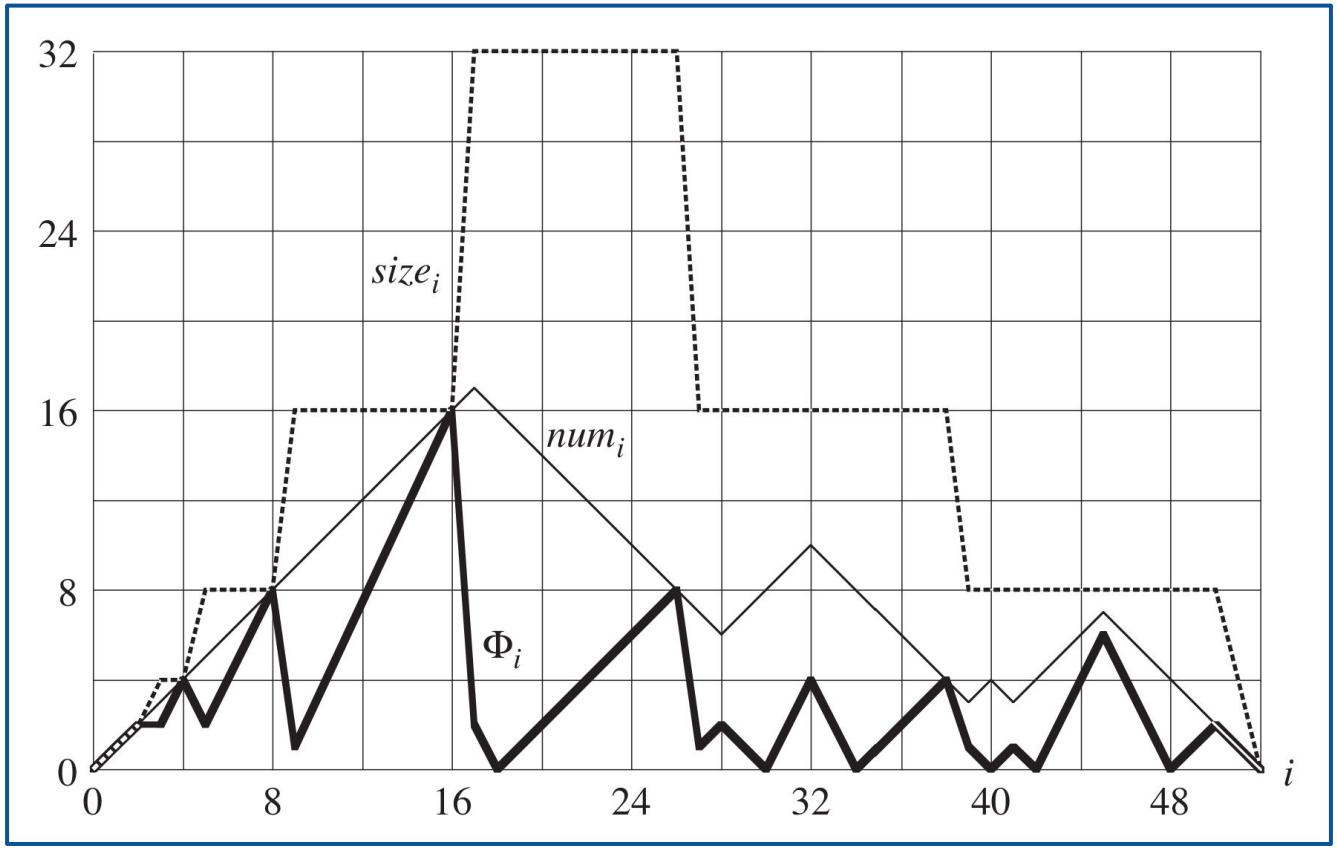
```
// Resize to provide space for new element:
if (stack->size == stack->capacity)
{
    size_t new_capacity = 2U * stack->capacity;
    if (stack->size == 0U)
    {
        new_capacity = 1U;
    }

    stack_resize(stack, new_capacity);
}
```



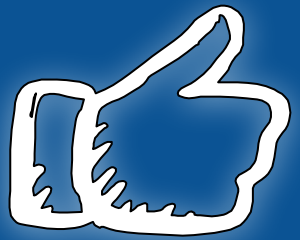
Полное изложение схемы:

- CLRS, изд. 3, глава 17.4;

Уменьшение ёмкости стека



Более эффективная и надёжная реализация стека



Реализация структуры данных “Стек”

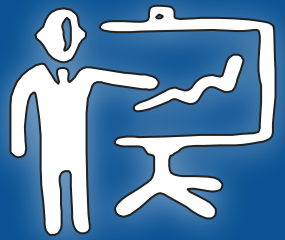
Реализации в репозитории курса: `09_resizing_stack`

- Чуть более эффективная и надёжная реализация (`stack.h`);

Особые моменты реализации:

- Схема изменения размера массива “с гистерезисом”;
- Защита стека от `use-after-free`;

Вопросы?



Красивые иконки взяты с сайта handdrawngoods.com