

Семинар 0

Обзор 1-го семестра

Бирюков Владимир

МФТИ

1. Язык C

1.1 Синтаксис языка C.

1.2 Память и указатели.

1.3 Сегменты памяти процесса. Стек и куча.

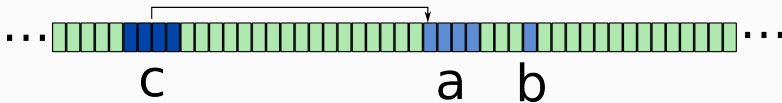
2. Алгоритмы и структуры данных

2.1 Сложность алгоритмов. $O(n)$ нотация.

2.2 Основные структуры данных: массив, список, дерево.

Язык С

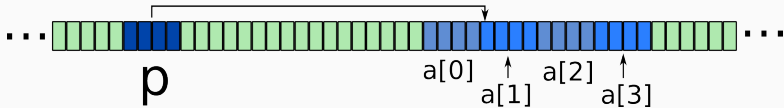
```
int a = 1;  
int * c = &a;  
*c = 5;
```



```
int a[4] = {1, 2, 3, 4};
```

```
int * p = &a[0];
```

```
p = p + 1;
```



Сегменты памяти процесса

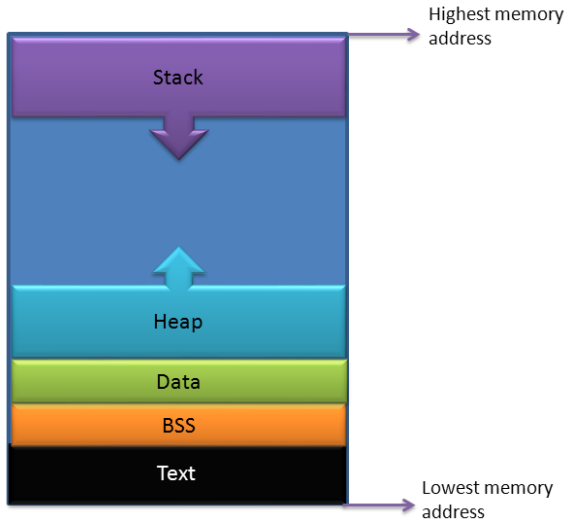
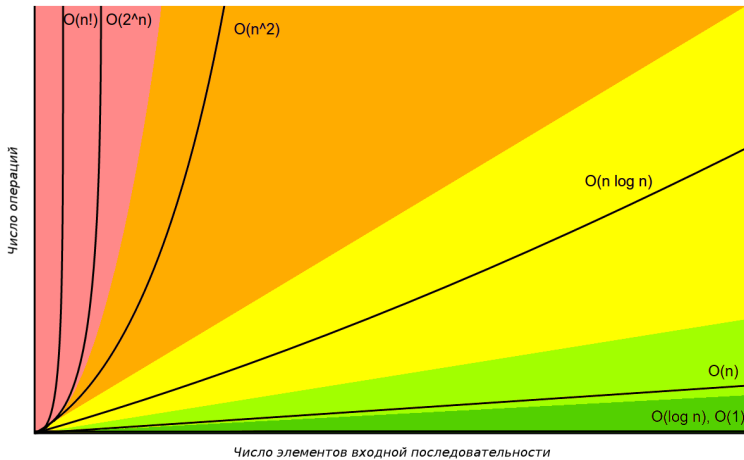


Figure : Process memory organization

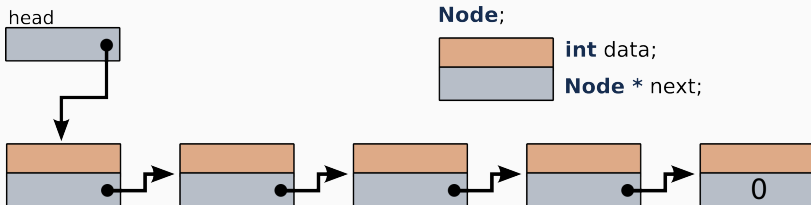
Алгоритмы и структуры данных



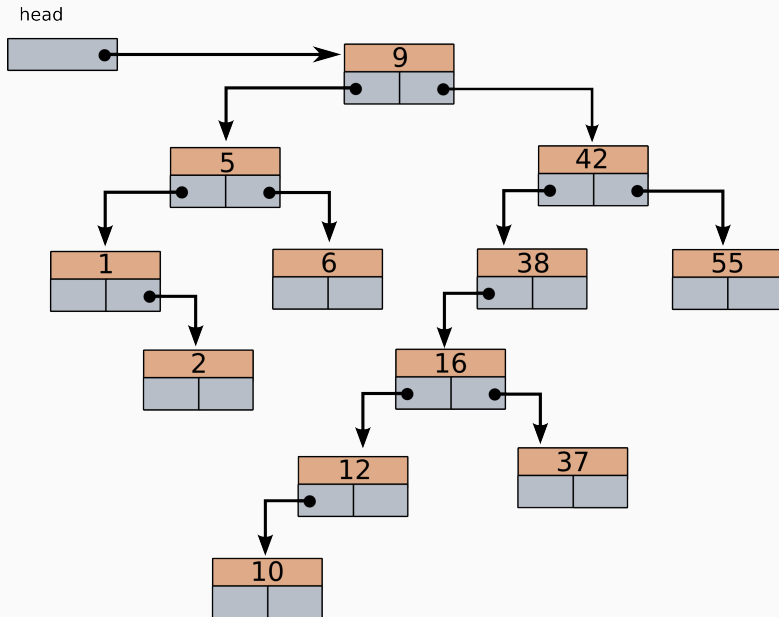
Алгоритмы сортировки

Алгоритм сортировки	Сложность(в среднем)
Сортировка вставками	$O(N^2)$
Сортировка пузырьком	$O(N^2)$
Сортировка выбором	$O(N^2)$
Сортировка слиянием	$O(N \log(N))$
Быстрая сортировка	$O(N \log(N))$
Цифровая сортировка	$O(kN)$

Структуры данных. Связный список



Структуры данных. Бинарное дерево



Операции со структурами данных

	Массив	Список	Бинарное дерево
index	$O(1)$	$O(N)$	$O(\log(N))$
find	$O(N)$	$O(N)$	$O(\log(N))$
insert	$O(N)$	$O(1)^*$	$O(\log(N))$
remove	$O(N)$	$O(1)^*$	$O(\log(N))$

* если известны указатели на данный и предыдущий элементы.

Новая тема: Раздельная компиляция

Язык C является компилируемым языком программирования
Компиляция – преобразование текста программы(исходного кода) в машинный код.

Зачем разбивать программу на файлы?

1. С небольшими файлами удобнее работать
2. Ускорения повторной компиляции при небольших изменениях
3. Структурирование кода

Общая схема сборки программы

