

Отчет по лабораторной работе № 1 по курсу «Функциональное программирование»

Студент группы М8О-307-19 МАИ *Инютин Максим Андреевич*, №10 по списку

Контакты: mainyutin@gmail.com

Работа выполнена: 08.03.2022

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806

Отчет сдан:

Итоговая оценка:

Подпись преподавателя:

1. Тема работы

Примитивные функции и особые операторы Common Lisp.

2. Цель работы

Научиться вводить S-выражения в Lisp-систему, определять переменные и функции, работать с условными операторами, работать с числами, используя схему линейной и древовидной рекурсии.

3. Задание (вариант № 1.48)

Реализовать на языке Common Lisp программу для умножения двух целых чисел за логарифмическое число шагов. Можно использовать функции сложения, вычитания, умножения и деления числа на 2, но нельзя умножать произвольные числа.

4. Оборудование студента

Процессор Intel Core i7-9750H (12) @ 4.5GHz, память: 32 Gb, разрядность системы: 64.

5. Программное обеспечение

ОС Ubuntu 20.04.4 LTS, компилятор GNU CLISP 2.49.92, текстовый редактор Atom 1.58.0

6. Идея, метод, алгоритм

Рассмотрим умножение двух чисел a и b на примере:

$$7 \cdot 13 = 7 \cdot (1 + 4 + 8) = (7 \cdot 1) + (7 \cdot 4) + (7 \cdot 8) = (7 \cdot 2^0) + (7 \cdot 2^2) + (7 \cdot 2^3).$$

Так как $13_{10} = 1101_2$, то можно умножать 7 на два 2, получая степени двойки, и прибавлять к результату, если на соответствующем бите стоит 1.

Мы рассмотрим все биты числа, которых не более $\log_2 b$. Умножение числа a на два будет выполнено не более $\log_2 b$ раз. По итогу будет совершенно порядка логарифма операций.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

8.1. Исходный код

```
; 1.48
```

```
(defun odd (x) (logand x 1))
```

```
(defun mul2 (x) (ash x 1))
```

```
(defun div2 (x) (ash x -1))
```

```
(defun fast* (x y)
```

```
(cond
```

```
  ((< y 0) (- (fast* x (- y))))
```

```
  ((= y 0) 0)
```

```
  ((= (odd y) 1) (+ x (fast* (mul2 x) (div2 y)))))
```

```
  ((fast* (mul2 x) (div2 y)))))
```

8.2. Результаты работы

```
[1]> (load "lab1.lisp")
;; Loading file lab1.lisp ...
;; Loaded file lab1.lisp
#P"/home/engineerx1/Study/6
term/functional-programming/lab1/lab1.lisp"
[2]> (fast* 5 0)
0
[3]> (fast* 0 4)
0
[4]> (fast* 1 3)
3
[5]> (fast* 9 1)
9
[6]> (fast* 3 4)
12
[7]> (fast* 4 3)
12
[8]> (fast* 88123 765)
67414095
[9]> (fast* 765 88123)
67414095
[10]> (fast* 8192 8192)
67108864
[11]> (fast* 1 1)
1
[12]> (fast* 0 0)
0
[13]> (fast* 123456789 987654321)
121932631112635269
[14]> (fast* -81 3)
-243
[15]> (fast* 81 -3)
-243
[16]> (fast* -14 -2)
28
```

9. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание
------	---------	-------------------------	------------

10. Замечания автора по существу работы

На практике гораздо проще просто перемножить два числа, однако в учебных целях считаю полезным реализовывать привычные операции с ограничениями.

11. Выводы

Я познакомился с синтаксисом языка Common Lisp. Было непривычно и сложно правильно расставить скобки, что и было основной трудностью.

Составленная программа работает за логарифмическое время. Подобный подход используется для быстрого возведения в степень чисел.