

Cyborg Organizer (Cyborgaizer) [*ru*]

Dmitry Ponyatov <dponyatov@gmail.com>

11 августа 2015 г.

Оглавление

О системе	4
Реализация на языке bl	4
I Язык bl	5
Комментарии	7
1 Система типов	8
1.1 Прimitives	8
1.1.1 int : целые	8
1.1.2 float : дробные	8
1.1.3 string : строки	9
1.1.4 ptr : указатели	9
1.1.5 bit : битовые строки	9
1.1.6 error : исключение	9
1.2 Составные типы и контейнеры	10
1.2.1 rec : записи	10
1.2.2 pair : пары	10

1.2.3	list : списки	10
1.2.4	dict : хэш-массивы	10
1.2.5	graph : деревья и графы	10
1.2.6	queue : очередь	10
1.2.7	stack : FIFO	10
1.3	Математические	10
1.3.1	<code>complex</code>	10
1.3.2	<code>vector</code>	10
1.3.3	<code>matrix</code>	10
1.4	Финансовые	11
1.4.1	<code>currency</code>	11
1.4.2	<code>tax</code>	11
1.5	Файловые и сетевые типы	11
1.5.1	bin : двоичный файл	11
1.5.2	txt : plain text текстовый файл	11
1.5.3	log : отладочный лог	11
1.5.4	<code>file</code> : генерализованные файлы	11
1.5.5	<code>socket</code> : платформенные сокеты	12
1.5.6	<code>codec</code> : потоковый конвертер форматов данных	12
1.5.7	<code>protocol</code> : реализация сетевого протокола	12
1.6	Типы обеспечения параллелизма	13
1.6.1	<code>thread</code> : нить	13
1.6.2	<code>mutex</code>	13
1.6.3	<code>semaphore</code>	13
1.6.4	<code>node</code> : вычислительный узел (ядро процессора)	13
1.6.5	<code>group</code> : группа узлов	13
1.6.6	<code>msg</code> : сообщение	13

1.6.7	prio: приоритет	13
1.6.8	pipe: межпроцессное соединение	13
1.6.9	share: шара (общая структура данных)	13
1.6.10	task: задача	13
1.6.11	sched: планировщик	13
1.7	Пользовательский интерфейс	13
1.7.1	display: дисплей	14
1.7.2	hid: контроллер ввода	14
1.7.3	console: текстовая консоль	14
1.7.4	form: графическая форма, виджет	14
1.7.5	window: окно	14
1.7.6	text: текст	14
1.7.7	canvas: область графического ввода/вывода	15
1.7.8	selector: выбор элементов (меню, радиокнопки)	15
1.7.9	tree: графическое дерево	15
1.7.10	table: таблица	15
1.7.11	plot: график	15
1.7.12	font: шрифт	15
1.7.13	status: статус-элемент (строка состояния, градусник, лампа)	15
1.7.14	placer: оконный менеджер или расстановщик элементов GUI	16
1.8	Геометрические и САПР типы	16
1.8.1	layer	16
1.8.2	point	16
1.8.3	line: линия	16
1.8.4	arc: дуга	16
1.8.5	nurbs: кривая Безье	16
1.8.6	color: цвет	17

1.8.7	facet: полигон	17
1.8.8	body: тело	17
1.8.9	dim: геометрический размер	17
1.8.10	mat: материал	17
1.8.11	camera	17
1.8.12	tool: инструмент	17
1.8.13	machine: станок	18
1.8.14	detail	18
1.8.15	assembly	18
1.8.16	technology	18
1.8.17	sheet	18
1.8.18	view	18

О системе

Это прототип расширяемой пользователем облачной системы-органайзера:

- управление задачами
- тайм-менеджмент
- персональная бухгалтерия
- организация групповой работы

Реализация на языке Ы

В порядке эксперимента система реализуется на языке динамического прототипирования Ы.

Описание на языке Ы транслируется в комплект файлов исходных текстов на C_+^+ и набор документации в формате \LaTeX .

Часть I

Язык ы

- динамический
в LISP-смысле: программа сама является структурой данных, и может модифицировать другие программы или сама себя, создавать и удалять программы в процессе выполнения. Так как очень часто приходится работать с данными в текстовых форматах, в ядре языка предусмотрен функционал создания парсеров, оптимизаторов, трансляторов и компиляторов любых других языков.
- объектный
- параллельный
при разработке языка большое внимание уделяется обеспечению параллелизма в самом широком смысле: микропоток, использование потоков runtime-системы и /или ОС, управление выполнением программ на гетерогенных вычислительных кластерах, облачных сервисах и p2p распределенных сетях, средства платформенно-независимой сериализации, поддержка персистентности и резервирования, синтаксическая поддержка языком параллельного программирования.
- run-time спецификация
объектов в процессе выполнения программы не предусматривает предварительное определение объектов, при попытке использования несуществующего объекта открывается интерактивная отладочная сессия
- интерактивная отладка
в стиле SmallTalk позволяет программисту создавать программу в диалоговом режиме в процессе отладки и прогона на тестовом стенде или наборе юнит-тестов.
- компилируемый через трансляцию
результатирующая система может быть оттранслирована¹ в исходный код на C₊ для обеспечения переносимости программ для систем, для которых не реализована BL-машина, недостаточно аппаратных

¹ ограниченно

ресурсов², или предъявляются жесткие требования по надежности³.

- литературное программирование⁴ и автоматическая генерация документации

Комментарии

// строчный комментарий

/* блочный комментарий */

² встраиваемые системы

³ ↑

⁴ literate programming

Глава 1

Система типов

1.1 Примитивные

1.1.1 `int`: целые

```
a = 123  
b = int("-567")
```

1.1.2 `float`: дробные

```
a = 12.34  
b = float("-5.6e-7")
```

1.1.3 **string**: строки

a = 'одиначные "кавычки"'

b = "двойные 'кавычки'"

c = "используйте \"\t для кавитирования \" и спец.символов\x07\n"

1.1.4 **ptr**: указатели

1.1.5 **bit**: битовые строки

Были введены для работы с низкоуровневыми машинными целыми и упакованными двоичными данными, в т.ч. использования в качестве бинарных масок.

1.1.6 **error**: исключение

1.2 Составные типы и контейнеры

1.2.1 **rec**: записи

1.2.2 **pair**: пары

1.2.3 **list**: списки

1.2.4 **dict**: хэш-массивы

1.2.5 **graph**: деревья и графы

1.2.6 **queue**: очередь

1.2.7 **stack**: FIFO

1.3 Математические

1.3.1 **complex**

1.3.2 **vector**

1.3.3 **matrix**

1.4 Финансовые

1.4.1 currency

1.4.2 tax

1.5 Файловые и сетевые типы

1.5.1 **bin**: двоичный файл

1.5.2 **txt**: plain text текстовый файл

1.5.3 **log**: отладочный лог

log обеспечивает запись отладочной информации в процессе выполнения программ, поддерживаются текстовые файлы, сетевое логирование и передача данных в интерактивный отладчик.

1.5.4 **file**: генерализованные файлы

Универсальный файл с набором кодеков, лексическим/синтаксическим анализатором, динамической структурой, обработчиками событий,...

1.5.5 socket: платформенные сокет

1.5.6 codec: потоковый конвертер форматов данных

1.5.7 protocol: реализация сетевого протокола

1.6 Типы обеспечения параллелизма

1.6.1 thread: нить

1.6.2 mutex

1.6.3 semaphore

1.6.4 node: вычислительный узел (ядро процессора)

1.6.5 group: группа узлов

1.6.6 msg: сообщение

1.6.7 prio: приоритет

1.6.8 pipe: межпроцессное соединение

1.6.9 share: шара (общая структура данных)

1.6.10 task: задача

1.6.11 sched: планировщик

1.7 Пользовательский интерфейс

1.7.1 **dislay**: дисплей

.

1.7.2 **hid**: контроллер ввода

HID: Human Inteface Device

- клавиатура
- мышь
- тач
- джойстик
- MIDI

1.7.3 **console**: текстовая консоль

.

1.7.4 **form**: графическая форма, виджет

.

1.7.5 **window**: окно

.

1.7.6 **text**: текст

.

1.7.7 **canvas**: область графического ввода/вывода

.

1.7.8 **selector**: выбор элементов (меню, радиокнопки)

.

1.7.9 **tree**: графическое дерево

.

1.7.10 **table**: таблица

.

1.7.11 **plot**: график

.

1.7.12 **font**: шрифт

.

1.7.13 **status**: статус-элемент (строка состояния, градусник, лампа)

.

1.7.14 **placer**: оконный менеджер или расстановщик элементов GUI

1.8 Геометрические и САПР типы

1.8.1 **layer**

1.8.2 **point**

1.8.3 **line**: линия

1.8.4 **arc**: дуга

1.8.5 **nurbs**: кривая Безье

1.8.6 color: цвет

.

1.8.7 facet: полигон

.

1.8.8 body: тело

.

1.8.9 dim: геометрический размер

.

1.8.10 mat: материал

.

1.8.11 camera

.

1.8.12 tool: инструмент

.

1.8.13 machine: станок

.

1.8.14 detail

.

1.8.15 assembly

.

1.8.16 technology

.

1.8.17 sheet

.

1.8.18 view

.