

После Оберона

Кушнир П.М.

Москва, 2016 год

BEGIN

Нулевой километр

- Проект «Лунь»
 - герметичность
 - x86win/*nix кроссплатформенность без перекомпиляции
 - cli/gui/service/daemon режимы простой заменой исполняемого файла
- Уход из большого Оберона
- Go почти смог (без шуток)
- java, javascript, вы хоть понимаете, что вы натворили?

Троичный слайд

- TRISC: первый эксперимент с компилятором Oberon
- Отдельный доклад (День Оберона 2014)
- Отдельная публикация о версии троичной машины в браузере
 - <https://habrahabr.ru/post/254261/>

Компилировали-компилировали

- “Герметизация” компилятора
- “Сменить платформу легко”, говорили они...
- Подсистема Хев, ast-генератор для компилятора Blackbox (DevCompiler)
 - Очень непростая объектная структура
- От компилятора к интерпретатору за 80 дней, проект Framework (fw) на go lang
- Отдельная публикация по результатам
 - <https://habrahabr.ru/post/252677/>

ЭТО ПЕАР!

- Ряд публикаций на “ведущем IT-ресурсе Европы” aka “мицелий” (ц)
 - <https://habrahabr.ru/post/258727/>
 - <https://habrahabr.ru/post/257953/>
 - <https://habrahabr.ru/post/258391/>
- Сгорая сам, свети другим...
 - <https://habrahabr.ru/post/258917/>
 - <https://habrahabr.ru/post/259755/>
- Рафик, где мой трафик?!

Проект LEAF

- Слезайте с плеч титанов и садитесь за руль велосипеда
- Императивный язык, как дедушка Вирт учил... не только на бумаге
- Кое-что своё
- Компилятор почти Оберона на golang
- Рантайм почти slim-binaries на golang

Проект LEAF

- Пример кода

```
1  MODULE TestFact
2      IMPORT Log
3
4      VAR max, res INTEGER
5
6      PROCEDURE Fact
7          VAR x-, res+ INTEGER
8          INFIX res x
9      BEGIN
10         IF x > 0 THEN res := x * \Fact x - 1
11         ELSE res := 1 END
12     END Fact
```


Проект LEAF

- Наследие
 - императивный
 - модульный
 - «по Дейкстре»
- Отсебятина
 - только процедуры, только именованные параметры
 - инфиксный вызов
 - только базовые типы (INTEGER, BOOLEAN, CHAR, STRING)
 - встроенные динамические коллекции (MAP, LIST, SET)
 - вычищена лишняя пунктуация, вместо неё пробелы и переводы строк (не так, как в Python :)
 - тип АТОМ
 - встроенная шина сообщений, через неё API для host-части
- Всё гораздо проще, чем кажется

Проект LEAF

- Каноничный компилятор Oberon-2013 на `golang`, с полной юникодизацией
- Подробный AST, пригодный к прямой интерпретации
- Управляемая среда исполнения
- Ссылки
 - <https://github.com/kpmy/leaf/wiki>

Проект LOMO

- LoLa - королева HDL рок-н-ролла
- Соответствие
 - Деталь-Модуль-Агент
 - Вход-Параметр-Перцептор
 - Выход-Вызов с параметром-Активность
 - Сигнал-Значение-Сообщение
- Язык объектных сообщений
 - декларативный язык

Проект LOMO

- Пример кода (шутка про ломографию)

```
UNIT Top
  REG r, x INTEGER
  VAR max INTEGER
PROCESS
  10 -> max
  x >= 0 ? r + x : r -> r
  x >= 0 ? x < max ? x + 1 : 1 : 1 -> x
END Top
```

Проект LOMO

- Наследие
 - единый модуль-объект
 - единое присвоение
 - тернарный оператор
 - нативный параллелизм
 - регистры
 - чисто входные и чисто выходные параметры
- Отсебятина
 - лёгкий синтаксис в стиле LEAF
 - все базовые типы данных LEAF
 - модуль как тип для сложных вычислений

Проект LOMO

- Каноничный компилятор LoLa на golang
- Параллельная среда исполнения на golang
- Взаимодействие с LEAF
 - был план интеграции обоих языков
- Ссылки
 - <https://github.com/kpmy/lomo/wiki>

Проект ОТ

- Язык разметки, сериализации данных и объектных шаблонов
 - модульность (как в Обероне)
 - типы данных (как в JSON)
 - иерархичность и классы (как в XML)
 - ссылки (как в Blackbox Stores)
 - синтаксис (как в Oberon и YAML)
 - шаблоны и контексты (как в JSP)

Проект OT

- Пример кода

o.t. is when this

```
<note status="saved">  
  <to>Tove</to>  
  <from>Jani</from>  
  <heading>Reminder</heading>  
  <body>Don't forget me this weekend!</body>  
</note>
```

becomes this

```
note:  
  status :: saved;  
  to: "Tove";  
  from: "Jani";  
  heading: 'Reminder';  
  body: `Don't forget me this weekend!`;  
;
```


Проект ОТ

- Проверенный парсер от LEAF
- Двухступенчатый разбор для текстового и бинарного представлений (оптимизация накладных расходов, пока не реализован)
- Ссылки
 - <https://github.com/kpmy/ot/wiki>
 - <https://habrahabr.ru/post/305168/>

Проект OWN

- Переосмысление LEAF для платформы NodeJS
 - модульность хоста (require)
 - настоящая кодогенерация javascript
 - попытки войти в специфичную область “онтологии” как пользовательские типы (не получилось)
- <https://github.com/kpmy/own>

Итого

- Мировоззрение «by Oberon»
- Собственный Стек Технологий - это просто:
 - учиться у других без поклонения
 - не брать чужое без осмысления
 - делать своё без тщеславия
 - не ждать милости от природы
 - видеть не только сходства, но и различия
 - не верить пафосным фразам евангелистов, всё проверять самому 😊
- Денег нет, но вы держитесь...

END;