

блoхбук

Dmitry Ponyatov <dponyatov@gmail.com>

1 мaртa 2016 г.

# Оглавление

Об этом сборнике . . . . .	2
<b>I    Язык <i>bI</i></b>	<b>3</b>
<b>AST</b> : Абстрактный Символьный Тип / <b>Sym</b> / . . . . .	4
 <b>II    emLinux для встраиваемых систем</b>	<b>6</b>
<b>1    clock</b> : коридорные электронные часы = контроллер дурдома	<b>7</b>
<b>2    gambox</b> : игровая приставка	<b>8</b>

III	Микроконтроллеры Cortex-Mx	9
IV	Технологии	10
V	Сетевое обучения	11
VI	Прочее	12
	Книги	13

## Об этом сборнике

В этот сборник я пишу отдельные статьи, сортированные только по общей тематике, и добавляю их, когда у меня в очередной раз зачесется L<sup>A</sup>T<sub>E</sub>X.

Это сборник черновых материалов, которые мне лень компоновать в отдельные книги, и которые пишутся просто по желанию “чтобы было”. Заказчиков на подготовку учебных материалов подобного типа нет, большая часть только на этапе освоения в мной самим, просто хочется иметь некое слабоупорядоченное хранилище наработок, на которые можно дать кому-то ссылку.

Часть I

Язык *bl*

# AST: Абстрактный Символьный Тип /Sym/

```
1 // ===== ABSTRACT SYMBOLIC TYPE (AST)
2 struct Sym {
```

тип (класс) и значение элемента данных

```
1 // -----
2     string tag;           // data type / class
3     string val;          // symbol value
```

конструкторы (токен используется в лексере)

```
1 // ----- constructors
2     Sym( string , string );    // <T:V>
3     Sym( string );            // token
```

AST может хранить (и обрабатывать) вложенные элементы

```
1 // ----- nest[]ed elements
2     vector<Sym*> nest;
3     void push(Sym*);
4     void pop();
```

параметры (и поля класса)

```
1 // ----- par{}ameters
2     map<string ,Sym*> pars;
3     void par(Sym*);           // add parameter
```

## Вывод дампа объекта в текстовом формате

```
1 // ----- dumping
2 virtual string dump(int depth=0);    // dump symbol object as text
3 virtual string tagval();              // <T:V> header string
4 string tagstr();                     // <T: 'V'> Str-like header string
5 string pad(int);                     // padding with tree decorators
```

## Вычисление объекта

```
1 // ----- compute (evaluate)
2 virtual Sym* eval();
```

## Операторы

```
1 // ----- operators
2 virtual Sym* str();                  // str(A) string representation
3 virtual Sym* eq(Sym*);                // A = B assignment
4 virtual Sym* inher(Sym*);             // A : B inheritance
5 virtual Sym* member(Sym*);            // A % B,C named member (class slot)
6 virtual Sym* at(Sym*);                 // A @ B apply
7 virtual Sym* add(Sym*);                // A + B add
8 virtual Sym* div(Sym*);                // A / B div
9 virtual Sym* ins(Sym*);                // A += B insert
10 };
```

## Часть II

em*Linux* для встраиваемых систем

# Глава 1

clock: коридорные электронные часы =  
контроллер дурдома



## Глава 2

gambox: игровая приставка

## Часть III

# Микроконтроллеры Cortex-Mx

# Часть IV

## Технологии

# Часть V

## Сетевое обучения

# Часть VI

## Прочее

# Литература

[1] **Dragon Book: Компиляторы**

Ахо, Сети, Ульман

[2] **SICP: Структура и интерпретация компьютерных программ**