

aForth. Словари. Слова.

Шамин Алексей Алексеевич

21.07.08

1 Описание слов словаря *stddict*

1.1 Слово: /

Назначение

Операция деления 1го элемента стека на 0й. Со стека снимаются 2 элемента, на стек кладётся их частное. Требует наличия на стеке двух аргументов.

Пример использования

```
<целое1> <целое0> /  
<вещественное1> <вещественное0> /  
<вещественное1> <целое0> /  
<целое1> <вещественное0> /
```

При делении целого и вещественного тип приводится к вещественному.

1.2 Слово: -

Назначение

Операция вычитания 0го элемента стека из 1го. Требует наличия на стеке двух аргументов. Со стека снимаются 2 элемента, на стек кладётся их разность.

Пример использования

```
<целое1> <целое0> -  
<вещественное1> <вещественное0> -  
<вещественное1> <целое0> -  
<целое1> <вещественное0> -
```

При вычитании целого и вещественного тип приводится к вещественному.

1.3 Слово: *

Назначение

Операция умножения 0го элемента стека на 1й. Требует наличия на стеке двух аргументов. Со стека снимаются 2 элемента, на стек кладётся их произведение.

Пример использования

```
<целое1> <целое0> *  
<вещественное1> <вещественное0> *  
<вещественное1> <целое0> *  
<целое1> <вещественное0> *
```

При умножении целого и вещественного тип приводится к вещественному.

1.4 Слово: +

Назначение

Операция сложения. Требует наличия на стеке двух аргументов. Со стека снимаются 2 элемента, на стек кладётся их сумма.

Пример использования

```
<целое1> <целое0> +  
<строка1> <строка0> +  
<вещественное1> <вещественное0> +  
<вещественное1> <целое0> +  
<целое1> <вещественное0> +
```

При сложении целого и вещественного тип приводится к вещественному.

1.5 Слово: and

Назначение

Побитовое И двух верхних операндов стека. Требует наличия на стеке двух целых аргументов.

Пример использования

```
<целое1> <целое0> and
```

1.6 Слово: lshift

Назначение

Сдвигает <целое1> на число битов, указанное в <целое0> влево. Требует наличия на стеке двух целых аргументов.

Пример использования

<целое1> <целое0> lshift

1.7 Слово: not

Назначение

Побитовое НЕ (инверсия всех бит) вершины стека. Требует наличия на стеке одного целого аргумента.

Пример использования

<целое0> not

1.8 Слово: or

Назначение

Побитовое ИЛИ двух верхних операндов стека. Требует наличия на стеке двух целых аргументов.

Пример использования

<целое1> <целое0> or

1.9 Слово: rshift

Назначение

Сдвигает <целое1> на число битов, указанное в <целое0> вправо. Требует наличия на стеке двух целых аргументов.

Пример использования

<целое1> <целое0> rshift

1.10 Слово: xor

Назначение

Побитовое ИСКЛЮЧАЮЩЕЕ ИЛИ двух верхних операндов стека. Требует наличия на стеке двух целых аргументов.

Пример использования

<целое1> <целое0> xor

1.11 Слово: =

Назначение

Операция сравнения двух верхних эл-тов стека. Со стека снимаются 2 элемента, на стек кладётся результат их сравнения. Если они равны, то на вершину ложится 1, если не равны, то 0. Требуется наличия на стеке двух аргументов.

Пример использования

```
<целое1> <целое0> =
<вещественное1> <вещественное0> =
<вещественное> <целое> =
<целое> <вещественное> =
При сравнении целого и вещественного тип приводится к вещественному.
```

1.12 Слово: >

Назначение

Операция сравнения двух верхних эл-тов стека. Со стека снимаются 2 элемента, на стек кладётся результат их сравнения. Если 1й элемент стека больше 0го, то на вершину ложится 1, иначе 0. Требуется наличия на стеке двух аргументов.

Пример использования

```
<целое1> <целое0> >
<вещественное1> <вещественное0> >
<вещественное> <целое> >
<целое> <вещественное> >
При сравнении целого и вещественного тип приводится к вещественному.
```

1.13 Слово: <

Назначение

Операция сравнения двух верхних эл-тов стека. Со стека снимаются 2 элемента, на стек кладётся результат их сравнения. Если 1й элемент стека меньше 0го, то на вершину ложится 1, иначе 0. Требуется наличия на стеке двух аргументов.

Пример использования

```
<целое1> <целое0> <
<вещественное1> <вещественное0> <
<вещественное> <целое> <
<целое> <вещественное> <
При сравнении целого и вещественного тип приводится к вещественному.
```

1.14 Слово: {

Назначение

Начинает определение массива. Массив может включать в себя любые типы данных. Строки и вещественные сохраняются как последовательность байт памяти им соответствующая.

Целые в массиве могут интерпретироваться как - целые (int), короткие (short) или байты (byte). По умолчанию целые сохраняются как int.

Интерпретация содержимого массива - целиком удел программиста.

Слова int, byte и short служат модификаторами, показывающими, как интерпретировать следующие за ними целые значения.

Пример использования

```
{ 1 2 3 4 ( сохраняются в массиве как целые - по умолчанию )
short 1 2 3 4 ( сохраняются в массиве как короткие целые )
byte 1 2 3 4 ( сохраняются в массиве как короткие байты )
"Строка в массиве" ( Строка сохраняется целиком, с завершающим ну-
лём )
1.2 3.4 0.0 ( Сохраняются как вещественные )
}
```

1.15 Слово: }

Назначение

Заканчивает определение массива. Описание - см. слово {.

1.16 Слово: : (двоеточие)

Назначение

Начинает определение нового слова (его компиляцию).

Пример использования

```
: <новое слово> <слова> ;
```

1.17 Слово: ; (точка с запятой)

Назначение

Заканчивает определение нового слова (его компиляцию).

Пример использования

```
: <новое слово> <слова> ;
```

1.18 Слово: doword

Назначение

Исполняет строку как строку программы.

Пример использования

<строка> doword

1.19 Слово: include

Назначение

Загружает и исполняет файл.

Пример использования

<имя файла> include

1.20 Слово: if

Назначение

Условие if else endif

Пример использования

Форма 1 - краткая

<условие>

if

<операторы, выполняются, если условие не 0 (истинно)>

endif

Форма 2 - полная

<условие>

if

<операторы, выполняются, если условие не 0 (истинно)>

else

<операторы, выполняются, если условие 0 (ложно)>

endif

1.21 Слово: until

Назначение

Цикл begin-until. Цикл выполняется хотябы один раз и повторяется при ЛОЖНОМ (0) условии

Пример использования

```
begin
<Операторы, выполняются хотябы 1 раз>
<Условие<>0 - выход, 0- повтор цикла>
until
```

1.22 Слово: while

Назначение

Цикл begin-while-repeat. Цикл может не выполниться ни разу и повторяется при ИСТИННОМ (не 0) условии.

Пример использования

```
begin
<условие, если <>0, то цикл выполняется>
while
<тело цикла>
repeat
```

1.23 Слово: close

Назначение

Закрывает файл, дескриптор которого на стеке. Возвращает на стек - 0 или отрицательное значение при ошибке.

Пример использования

```
<целое-дескриптор> close
```

1.24 Слово: creat

Назначение

Создать файл, путь к которому - строка на стеке. Если файл существует - он усекается до нулевой длины. Возвращает на стек - дескриптор файла или отрицательное значение при ошибке.

Пример использования

```
<строка-путь-к-файлу> creat
```

1.25 Слово: getc

Назначение

Считывает символ из файла, дескриптор которого находится на стеке и помещает этот символ на стек. Если символов нет или ошибка - возвращает -1.

Пример использования

<целое-дескриптор> getc

1.26 Слово: open

Назначение

Открыть файл, путь к которому - строка на стеке. Если файл не удаётся открыть - он создаётся. Возвращает на стек - дескриптор файла или отрицательное значение при ошибке.

Пример использования

<строка-путь-к-файлу> open

1.27 Слово: putchar

Назначение

Записывает символ в файл. Если ошибка - возвращает -1, иначе - код записанного символа - 0..255.

Пример использования

<целое-дескриптор> <целое-символ> putchar

1.28 Слово: save

Назначение

Выгружает текущий словарь в файл, имя которого на стеке. Если файл существует - перезаписывает его.

Пример использования

<имя файла> save

1.29 Слово: land

Назначение

Логическое И двух верхних операндов стека (0 - ложь, не 0 - истина). Требует наличия на стеке двух целых аргументов.

Пример использования

<целое1> <целое0> land

1.30 Слово: lnot

Назначение

Логическое НЕ вершины стека (0 - ложь, не 0 - истина). Требует наличия на стеке одного целого аргумента.

Пример использования

<целое0> lnot

1.31 Слово: lor

Назначение

Логическое ИЛИ двух верхних операндов стека (0 - ложь, не 0 - истина). Требует наличия на стеке двух целых аргументов.

Пример использования

<целое1> <целое0> lor

1.32 Слово: aresize

Назначение

Изменение размера массива в байтах. Изменяет размер массива на заданный. Если новый размер больше прежнего, то добавляемое место забивается нулями. Если новый размер меньше прежнего, то массив усекается.

Пример использования

<новый размер массива> <массив> aresize

1.33 Слово: smove

Назначение

Копировать содержимое из одной области памяти в другую. Области памяти могут пересекаться.

Пример использования

<адр.источника> <адр.приёмника> <размер в байтах> smove

1.34 Слово: free

Назначение

Освободить память, начальный адрес которой, находится на стеке. Память должна быть ранее выделена словом malloc.

Пример использования

<адрес памяти> free

1.35 Слово: &

Назначение

Требуется один аргумент. Получить адрес переменной. Возвращает на стек адрес переменной или массива, лежащей на его вершине.

Пример использования

<переменная или массив> &

1.36 Слово: getmchar

Назначение

Получить байт по указанному адресу, преобразовать в целое и поместить на стек.

Пример использования

<адрес> getmchar

1.37 Слово: getmint

Назначение

Получить целое по указанному адресу и поместить на стек.

Пример использования

<адрес> getmint

1.38 Слово: getmreal

Назначение

Получить вещественное по указанному адресу и поместить на стек.

Пример использования

<адрес> getmreal

1.39 Слово: getmshort

Назначение

Получить короткое целое по указанному адресу и поместить на стек.

Пример использования

<адрес> getmshort

1.40 Слово: getmstr

Назначение

Получить строку по указанному адресу и поместить на стек.

Пример использования

<адрес> getmstr

1.41 Слово: malloc

Назначение

Требует один аргумент. Выделить память размером в N байт и вернуть начальный адрес на стеке. Если память выделить не удалось - возвращается 0.

Пример использования

<N> malloc

1.42 Слово: realloc

Назначение

Изменить размер памяти, выделенной с помощью слова malloc. Возвращает на стек адрес начала блока памяти или 0, если изменение размера не удалось.

Пример использования

<новый размер в байтах> <адрес> realloc

1.43 Слово: setmchar

Назначение

Запись байта по адресу. Байт и адрес на стеке представлены целыми числами.

Пример использования

<байт> <адрес> setmchar

1.44 Слово: setmint

Назначение

Запись целого по адресу.

Пример использования

<целое32> <адрес> setmint

1.45 Слово: setmreal

Назначение

Запись вещественного по адресу.

Пример использования

<вещественное> <адрес> setmreal

1.46 Слово: setmshort

Назначение

Запись короткого целого по адресу.

Пример использования

<короткое целое> <адрес> setmshort

1.47 Слово: setmstr

Назначение

Запись строки по адресу.

Пример использования

<строка> <адрес> setmstr

1.48 Слово: isword

Назначение

Проверяет - есть ли в словаре слово с заданным именем. Если есть - ложит на стек TRUE (1), если нет - FALSE(0)

Пример использования

```
"dup" isword . CR ( Напечатает 1 )  
"новое слово" isword . CR ( Напечатает 0 )
```

1.49 Слово: mktime

Назначение

Поместить на стек время сборки - строку в формате дд.мм.гггг-чч.мм.сс.

Пример использования

```
mktime
```

1.50 Слово: subversion

Назначение

Поместить на стек подверию версию aForth - строку.

Пример использования

```
subversion
```

1.51 Слово: version

Назначение

Поместить на стек версию aForth - строку, содержащую несколько чисел, разделённых точками.

Пример использования

```
version
```

1.52 Слова: . .x .X .C

Назначение

Печать значения с вершины стека.

. - Целые печатаются в 10м виде.

.x - Целые печатаются в 16м виде. Буквы, обозначающие 16е цифры - строчные.

.X - Целые печатаются в 16м виде. Буквы, обозначающие 16е цифры - прописные.

.C - Печать символа, код которого находится на стеке в виде целого.

Пример использования

10 . " десятичная печать " .

10 .x " шестнадцатеричная печать, строчные буквы-цифры " .

10 .X " шестнадцатеричная печать, прописные буквы-цифры " .

48 .C " печать символа с кодом 48 " .

1.53 Слово: intsize

Назначение

Поместить на стек размер целого (в байтах).

Пример использования

intsize

1.54 Слово: realsize

Назначение

Поместить на стек размер вещественного (в байтах).

Пример использования

realsize

1.55 Слово: shortsize

Назначение

Поместить на стек размер короткого целого (в байтах).

Пример использования

shortsize

1.56 Слово: size

Назначение

Поместить на стек размер (в байтах) переменной или константы.

Пример использования

<Переменная или константа> size

1.57 Слово: drop

Назначение

Уничтожает вершину стека. Кол-во элементов стека уменьшается на 1.

Пример использования

<константа или переменная> drop

1.58 Слово: dup

Назначение

Дублирует значение вершины стека.

Пример использования

<константа или переменная> dup

1.59 Слово: pover

Назначение

Копирование произвольного значения стека. На вершине - номер от начала стека

0 - верхнее значение и т.п.

1 0 pover - на вершине 1

3 1 1 pover - на вершине 3

0 pover то же, что и dup

1 pover то же, что и over

Пример использования

<константы или переменные> <константа или переменная> pover

1.60 Слово: nswap

Назначение

Обменять значения вершина стека и любое предыдущее.

0 nswap - аналогично swap

1 nswap - вершина стека и 2й элемент

и т.п.

Пример использования

<константа или переменная> <константа или переменная> nswap

1.61 Слово: over

Назначение

Дублирует значение, следующее за вершиной стека (1й элемент стека).

Пример использования

<константа или переменная> <константа или переменная> over

1.62 Слово: swap

Назначение

Меняет местами два верхних значения на стеке (0й и 1й элементы стека).

Пример использования

<константа или переменная> <константа или переменная> swap

1.63 Слова: istype array-type int-type real-type str-type

Назначение

Слово istype снимает со стека значение и возвращает на стек константу-код, соответствующую типу снятого значения. Каждый тип данных имеет свой уникальный код.

Слова array-type, int-type, real-type, str-type возвращают на стек код соответствующего типа.

Пример использования

```
( Проверка типов. На стеке значение неизвестного типа. )
: checktype
istype
dup array-type = if "Тип: массив" CR endif
dup int-type = if "Тип: целое" CR endif
dup real-type = if "Тип: вещественное" CR endif
str-type = if "Тип: строка" CR endif ;
```


1.64 Слова: `typecast`

Назначение

Слово принудительно приводит тип значения на вершине стека к указанному, если типы совместимы. Всегда возвращает константу указанного типа на стеке.

Пример использования

Формат команды:

`<значение> <код типа> typecast`

Примеры:

(Приведение к целому типу)

`2.345 int-type typecast . CR` (Напечатает 2 - целое число)

(Приведение к вещественному типу)

`5 real-type typecast . CR` (Напечатает 5.00000 - вещественное число)

1.65 Слово: `array`

Назначение

Создаёт массив байт заданной длины и инициализирует его нулями.

Пример использования

`<длина в байтах> array <имя массива>`

1.66 Слово: `variable`

Назначение

Создаёт переменную и инициализирует её начальным значением.

Пример использования

`<значение> variable <переменная>`

Тип переменной определяется из контекста.

`1 variable A` - Создаёт целочисленную переменную A

`1.0 variable A` - Создаёт вещественную переменную A

`"Строка" variable A` - Создаёт строковую переменную A

1.67 Слово: `!!`

Назначение

Снимает значение с вершины стека и помещает его в переменную.

Пример использования

<значение> <переменная> !!

Типы переменной и значения должны быть совместимы. Если типы переменной и значения, то значение всегда приводится к типу переменной.

2 Описание слов словаря *linux.dict*

2.1 Слово: `getchar`

Назначение

Ждёт, пока на стандартном вводе не появится символ и помещает его код на стек.

Пример использования

`getchar` . (Ждёт символ на стандартном вводе и печатает его код)

2.2 Слово: `getcharnb`

Назначение

Помещает на стек код символа из стандартного ввода, если символа нет - то помещает -1. То есть - то же, что и `getchar`, но не ждёт появления символа.

Пример использования

`getcharnb` . (печатает код символа или -1, если символа нет)