

Руководство по системе WEB

к. т. н. Жуков И. Б. © 2013
ibzh@yandex.ru

(Предварительный вариант)
<http://ibzh.eko3.ru/tex/webdescr.htm>

Введение

WEB-файл является обычным текстовым файлом, в котором вперемешку находятся команды, собственно, языка WEB, кусочки программы на языке PASCAL и их описания на языке TEX. Чтобы получить из WEB-файла текст программы на языке PASCAL, используется программа TANGLE. Она устраняет весь текст TEX и, используя команды WEB, объединяет разрозненные кусочки текста на языке PASCAL в одну программу, которая записывается на диск. Далее, полученный pas-файл можно скомпилировать в исполняемый exe-файл.

Чтобы получить описание программы в формате plainTEX служит программа WEAVE. Она копирует весь текст TEX с некоторыми изменениями в итоговый tex-файл, форматирует кусочки программы, создаёт указатель идентификаторов и пр. Из этого tex-файла можно получить соответствующими программами документ в форматах DVI, PS или PDF.

Состав WEB-файла

Пример простейшего WEB-файла приведён в листинге 1. Итоговый pas-файл — в листинге 2. Итоговый TEX-файл — в приложении.

WEB-файл состоит из нескольких разделов, начинающихся с команды @*, каждый из которых, в свою очередь, состоит из трёх частей: описания на языке TEX, макроопределения на языке WEB и кусочка программы на языке PASCAL. Любая из частей может отсутствовать, но порядок должен сохраняться: описание (если есть) должно быть первым, а кусочек программы (если присутствует) — последним. Из названий разделов в итоговом TEX-файле создаётся оглавление. В каждом разделе могут находиться несколько подразделов, объявляемых командой @_, которые также состоят из трёх частей (описания, макроопределений и кусочка программы, следующих в указанном порядке). Подразделы в оглавление уже не включаются. Перед первым разделом (до первой команды @*) может находиться необязательная часть т. н. *Limbo*, которую программа WEAVE полностью копирует в преамбулу TEX-файла, а TANGLE пропускает.

Итоговые файлы WEB TEX- и pas-файлы

TEX-файл, создаваемый WEAVE, начинается со строки `\input webmac`. За которой следует текст *Limbo*, в указатель вставляются все идентификаторы, встретившиеся в кусочках программы и макроопределениях, а также в описании кусочков между знаками `|...|`. Исключения составляют зарезервированные идентификаторы (`program`, `array` и т. п.) и идентификаторы, состоящие из одного символа.

Перечень WEB-команд

Все команды WEB состоят из двух символов, первый из которых — @ («коммерческое at»), а второй — буква или иной символ ASCII. Строчные и прописные буквы не различаются (например, @p и @P рассматриваются одинаково). Если нужно использовать символ @ в качестве обычного символа, а не как обозначение начала команды, нужно напечатать @@.

В приведённом ниже перечне команд заключённые в квадратные скобки буквы означают, что команда

L — может появляться в <i>Limbo</i>	\bar{L} — завершает <i>Limbo</i>
T — может появляться в описании на языке \TeX	\bar{T} — завершает описание на языке \TeX
P — может появляться в кусочке программы на языке PASCAL	\bar{P} — завершает кусочек программы
M — может появляться в имени модуля	\bar{M} — завершает имя модуля
C — может появляться в комментарии	\bar{C} — завершает комментарий
S — может появляться в строке	\bar{S} — завершает строку

◊ Как указывалось выше, $\@@ [C, L, M, P, S, T]$ обозначает символ $\@$, который просто заносится в выходной файл.

◊ Начало раздела (starred module) задаётся командой $\@* [\bar{L}, \bar{P}, \bar{T} \rightarrow M]$. Имя раздела должно обязательно заканчиваться точкой и размещаться в той же строке, что и команда $\@*$. В \TeX -файле название печатается жирным шрифтом и заносится в оглавление. Кроме того, номер раздела помещается в рас-файл в качестве комментария. Следует избегать использования сложных \TeX -команд в названии раздела, поскольку это может привести к конфликтам с макросами. Когда во время обработки WEB-файлов **TANGLE** и **WEAVE** встречаются $\@*$ они печатают на терминал звёздочку со следующим за ней номером модуля.

◊ Подразделы начинаются командой $\@_$. Вместо пробела за знаком $\@$ может стоять знак табуляции (с ASCII-кодом 9) или возврат каретки (с ASCII-кодом 13). Подразделы не должны появляться перед первым разделом.

После начала раздела (подраздела) следует обычный текст \TeX . В этот текст можно включать кусочки, форматированные в стиле **PASCAL** при помощи конструкций вида

| *pas-текст* |.

Этот *pas-текст* пропускается **TANGLE** и в рас-файл не попадает. **WEAVE** вставляет встретившиеся в *pas-тексте* идентификаторы в указатель. Примечания и имена модулей в *pas-тексте* не допускаются. Используйте его только для коротких текстов, иначе могут возникнуть сложности с форматированием.

Если нужно форматировать идентификатор, но при этом, чтобы он не попал в указатель, можно воспользоваться макросом $\{\{ \text{идентификатор} \}$. (Идентификаторы обозначаются курсивом, в отличие от ключевых слов, которые обозначаются прямым жирным шрифтом.)

Если нужно напечатать текст шрифтом пишущей машинки, используется макрос $\{ \{ \text{имя} \}$. Если в этот текст нужно вставить пробел (невидимый), то должна использоваться конструкция $\{ \}$. Если, нужен видимый пробел, то $_$.

Замечание. Макросы $_$ и $_.$ (в отличие от $| \dots |$) не являются частью **WEAVE** или **TANGLE**, они определены в файле `webmac.tex`, ссылка на который вставляется в начале `tex`-файла.

◊ Кусочки программы начинаются с команды $\@p [\bar{P}, \bar{T} \rightarrow P]$ (безымянные модули, *unnamed modules*). **TANGLE** помещает в рас-файл безымянные модули в том порядке, в котором они встречаются в `web`-файле. В `web`-файле должен быть по крайней мере один «безымянный модуль».

◊ Внутри безымянных модулей могут находиться ссылки на именованные модули, задаваемые командами $\@< \text{название модуля} \@> [P, \bar{T} \rightarrow P]$. Название модуля может занимать несколько строк. Длинные названия модулей писать неудобно, поэтому достаточно один раз записать полное название модуля, а в дальнейшем пользоваться сокращениями, например:

```

@ @<Длинное имя модуля @>=
  begin i:=1;

@ @<Длинное им...@>=
  end;

@ @p
@<Длинное...@>;

```

Многоточие перед знаками $\@>$ заменяет часть имени перед модуля.

В названиях модулей допускаются встроенные тексты **ПАСКАЛЯ** $| \dots |$. Префикс, разумеется, должен быть достаточно большим, чтобы подходить только к одному имени модуля, иначе **TANGLE** и **WEAVE** выдадут сообщение об ошибке.

◊ Обычно, TANGLE не помещает комментарии в рас-файл. Однако, в некоторых случаях это бывает нужно (например, директивы компилирования помещаются в скобках { и }). Для этой цели служат команды @{ и @} [P], текст между которыми помещается в рас-файл и заключается в фигурные скобки {...}.

◊ Команда @= *рас-текст* @> [P] задают текст, который целиком без изменений помещается в рас-файл (включая пробелы, символы { и пр.). WEAVE этот текст окружает прямоугольной рамкой.

◊ Команда @t *TEX-текст* @> [P] помещает *TEX-текст* в \hbox и форматирует как часть программы. TANGLE пропускает *TEX-текст*. Команды WEB в этом тексте не допускаются. Если нужно написать символ @, то печатается команда \AT! (она определена в файле webmac.tex).

Пример использования: @t \hskip 1cm @>

Это поместит горизонтальный промежуток размером в 1 см.

◊ Ссылки помещаются в автоматически создаваемый указатель с помощью одной из команд

- @^ *текст ссылки* @> [P,T] — для печати *текста ссылки* прямым шрифтом;
- @. *текст ссылки* @> [P,T] — для печати *текста ссылки* моноширинным шрифтом;
- @: *текст ссылки* @> [P,T] — для печати *текста ссылки* в формате, задаваемом макросом \9, который определяется пользователем.

TANGLE пропускает все *тексты ссылок*. В *тексте ссылки* не допускается использовать ни каких команд WEB, в том числе и @@. Если нужно вставить символ @, то используется команда «\AT!» (которая определена в стилевом файле webmac.tex). *Текст ссылки* вместе с открывающей и закрывающей командами должен размещаться на одной строке.

Одно из применений @: ... @> состоит в следующем. Если нужно, например, ввести текст \TeX, чтобы в указателе появилось слово TEX, то из-за косой черты, WEAVE его не поставит там, где размещаются остальные термины, начинающиеся на T. Для этого задаётся макрос

```
\def\9#1{}
```

тогда, введя @:TeX}{\TeX@>, получим следующее. WEAVE обработает WEB-файл и поместит в указатель вызов макроса \9{TeX}{\TeX}, который, в силу своего определения вставит только макрос TEX.

◊ Команды @'число и @"число [P,T] задают, соответственно, восьмеричное и шестнадцатеричное число. Числа @'100 и @"100 будут преобразованы TANGLE в 64 и 256, а WEAVE — в «'100» и «"100». Эти команды должны использоваться только для представления положительных чисел.

◊ TANGLE на месте команды @\ [P] разрывает строку в рас-файле.

◊ Контрольная сумма пула строк вводится командой @\$ [P].

На этом месте следует остановиться подробнее. Исходно система WEB создавалась для разработки TEX и иже с ним. TEX направлен на обработку строк, поэтому в WEB были введены средства для работы со строками. WEB поддерживает два вида строк: заключаемые в одинарные кавычки (которые распознаёт PASCAL), и заключаемые в двойные (которые PASCAL считает неправильными). Строки в одинарных кавычках так в программу и передаются, а строки в двойных кавычках помещаются в особый файл (строковый пул), а в PASCAL помещаются номера этих строк.

Первые 256 номеров (коды от 0 до 255) держатся для односимвольных строк (например, 65 для "A 90 для "Z" и т. д.), а номера, начиная с 256, используются для строк длиной от двух и более символов. Односимвольные строки в пул строк не заносятся, т. к. они представлены своими ASCII-кодами, а остальные заносятся следующим образом: длина строки (две десятичных цифры, первая из которых 0, если длина строки меньше 10) за которыми сразу следует текст строки без кавычек. Заканчивается строка символом возврата каретки. Последняя строка начинается со знака *, за которым следует контрольная сумма (девять десятичных цифр).

Контрольная сумма позволяет определять, соответствует ли данный пул строк программе или нет. Для того, чтобы в программе её можно было использовать и применяется символ @\$.

Если в WEB-файле мы укажем z1:="z"; z2="zz"; то на выходе рас-файле мы можем получить, например, z1:=122; z2= 256;, при условии, что "zz" у нас первая строка из нескольких букв.

◊ Команда @d [\bar{P} , \bar{T}] служит для определения макросов. Есть три варианта использования этой команды:

- @d *имя_константы* = *число*
- @d *имя_макроса* == *команды*
- @d *имя_макроса_с_параметром*(#) == *команды*

◊ Команда @f [\bar{P} , \bar{T}] служит для задания форматирования идентификатора. Она пропускается TANGLE. Если она используется для одного и того же идентификатора несколько раз, то последняя команда задаёт форматирование для *всего* документа.

Форматами могут выступать следующие ключевые слова

- function, procedure, program;
- const, label, type;
- var;
- array, file, set;
- record;
- begin;
- end;
- case;
- if;
- do, then, of;
- else;
- for, while, with;
- downto, to;
- repeat;
- until;
- xclause;
- goto, packed;
- nil;
- div, mod;
- and;
- in;
- not;
- or;

Слова, указанные в одной строке, форматируются одинаково.

◊ Команда @& [P] соединяет часть строки, находящуюся справа с частью строки, находящейся слева. Пробелы и разрывы строк не будут разделять эти две подстроки. Элемент, находящийся слева не должен быть точкой с запятой, т. к. на ней может происходить разрыв строк.

Следующие команды предназначены для управления разрывами строк. Они пропускаются TANGLE и используются только WEAVE.

◊ @/ [P] — принудительный разрыв строки внутри программы, форматируемой WEAVE. Эта команда должна использоваться только после операторов или предложений [clause?].

◊ @! [P] — то же вызывает разрыв строки внутри программы, но предназначена для использования в середине выражения, например, в правой части длинного оператора присваивания или в длинном условии оператора **if-then-else**.

◊ @# [P] — подобна @/, но вставляет небольшой дополнительный пробел между строками. Может использоваться, например, между определениями процедур или между группами макроопределений, которые хотя и находятся в одном модуле, но логически разделены.

◊ @+ [P] — отменяет разрыв строки, который иначе был бы вставлен WEAVE, т. е., например, перед оператором **else**, если нужно поместить на одной строке конструкцию **if-then-else**.

◊ Команда @, [P] вставляет тонкий пробел;

◊ Команда @; [P] рассматривается WEAVE также, как и знак «;», но только он является невидимым.

Команды @/, @!, @#, @+, @, и @; пропускаются программой TANGLE и на pas-файл никакого влияния не оказывают. TANGLE не заменяет «@;» на «;»!

◊ Последовательность @= *текст* @> [P] помещает *текст* в программу с точностью до символа.

◊ Команда @\ [P] вставляет разрыв строки в программу, создаваемую TANGLE. WEAVE, встречая её, так и пишет в выходной файл @\.

◊ Если указать @! [P, T] непосредственно перед идентификатором, то в указателе номер модуля будет подчёркнут. Это полезно, чтобы указать место, где идентификатор определяется или сообщаются важные подробности о нём. WEAVE неявно вставляет @! сразу за зарезервированными словами **function**, **procedure**, **program** и **var**, а также за командами @d и @f.

◊ Команда @? [P, T] используется, чтобы отменить явную или неявную команду @!.

Замечание. ПАСКАЛЬ не различает регистр символов в именах, но определения, создаваемые Web по командам @d, чувствительны. Вы можете написать @d xxx==что-нибудь. Тогда, встретив строку xxx, tangle подставит вместо неё что-нибудь, но, встретив xXx, так и вставит в файл xXx.

Замечание. Для задания размера листов в TeX-файле используются следующие команды:

- \pagewidth=6,5in — ширина основного текста страницы;
- \pageheight=8,7in — высота основного текста страницы;
- \fullpageheight=9in — высота страницы с учётом дополнительного места для колонтитула и номера страницы.

После изменения любого из этих размеров следует сразу вызвать \setpage. Эти команды должны указываться в части Лимбо.

Дополнительные особенности. Пара символов (* и *) заменяются на @{ и @}. Пара символов (. и .) — на [и].

Файл изменений

Файл изменений имеет следующий вид:

```
Необязательные примечания
@x Необязательное примечание
    Старые строки
@y Необязательное примечание
    Новые строки
@z Ещё одно примечание
```

```
Необязательные примечания
@x
    Старые строки
@y
    Новые строки
@z
```

TANGLE и WEAVE пропускают в файле изменений все строки до первого @x. Затем, ищут в WEB-файле последовательность строк, которая полностью совпадает с последовательностью строк между найденным @x и следующим @y. Соответствие должно быть с точностью до пробела. Пробелы в конце строки отбрасываются, Регистры учитываются. Далее эта последовательность строк заменяется строками, расположенными между @y и @z.

После этого в файле изменений пропускаются все строки до следующего @x, и процесс повторяется. Просмотр WEB-файла идёт последовательно. Заменяемые части в CHG-файле должны идти в том порядке, в каком они расположены в WEB-файле. Остаток строки, следующий за @x, @y или @z отбрасывается, и может использоваться для примечаний. Символы должны быть первыми символами в строке. Если перед ними есть пробел, то они пропускаются. Это позволяет легко временно отменять изменения, практически не меняя CHG-файл.

Регистр символов @x, @y и @z не различается.