

СОЗДАНИЕ МАССИВОВ МАКРОКОМАНД СРЕДСТВАМИ ЛАТЭХ

Цымай Д.В.

Куценко С.А.

24 апреля 2015 г.

Аннотация

В статье предложен дополнительный набор макрокоманд для разработки тестовых заданий средствами ЛАТЭХ. Рассмотрены области их возможного применения при подготовке текстов.

Ключевые слова: верстка документов, макрокоманды, массивы, LaTeX

Введение

ЛАТЭХ— наиболее популярный набор макрорасширений системы компьютерной вёрстки ТЭХ, который облегчает подготовку сложных документов. Использование макропакетов позволяет во многих случаях автоматизировать различные задачи набора текста и подготовки статей. Кроме того, ЛАТЭХ является расширяемым за счет возможности написания собственных макрокоманд для последующего применения, например при помощи встроенной в ТЭХ команды `\def`, описанной в книге Д. Кнута [1]. Чтобы случайно не переопределить существующую команду можно пользоваться встроенными в ЛАТЭХ командами `\newcommand` для определения новой команды и `\renewcommand` для осознанного переопределения существующей, описанными у С.М. Львовского [2].

Рассмотренные возможности удобны до тех пор, пока не потребуется определить значительное количество новых макрокоманд. Подобная задача возникла при разработке заданий для тестового контроля. Поскольку одни и те же вопросы могут повторяться в различных тестовых заданиях, то разумно определить для каждого из них макрокоманду. Это сокращает время как при верстке заданий, так и при корректировке вопросов. Вот здесь то и возникает новая сложность. Для каждой макрокоманды необходимо определить уникальное имя, в котором нельзя использовать цифры.

Необходимость удержания в памяти такого множества новых команд сводит к нулю все преимущества их повторного использования. При верстке тестов необходимо иметь возможность вставить фрагмент, соответствующий отдельному вопросу, в произвольное задание, не вспоминая при этом имя командной последовательности. У Д. Кнута [1] описана возможность реализации списочных макрокоманд. Данный подход не совсем удобен, так как предполагает сохранение всех фрагментов в определении одной списочной макрокоманды.

Одним из возможных решений является массив макрокоманд. Достаточно определить одно имя массива для фрагментов текста, разметить каждый вопрос (начало и конец) определенными ограничивающими командами и в дальнейшем обращаться к данным фрагментам по номерам, в соответствии с порядком их определения в тексте. Присвоение имен новым командам полностью перекладывается на компьютер, т.е. происходит прозрачно для пользователя.

1 Реализация массивов. Основные команды

Л^AT_EX не имеет собственных средств для работы с массивами макрокоманд. При помощи встроенных средств программирования [3] реализованы следующие возможности:

1. объявление массива с произвольным именем;
2. добавление в определенный ранее массив произвольных элементов;
3. перечисление всех или части элементов указанного массива в цикле с выполнением над каждым элементом произвольных команд форматирования.

Рассмотрим основные особенности реализации. Поскольку имя макрокоманды не должно включать арабских цифр, то заменим их на римские, которые отображаются при помощи букв латинского алфавита. Тогда последовательность команд `\qvery1`, `\qvery2`, `\qvery3`, ... может быть записана `\qveryi`, `\qveryii`, `\qveryiii`, ... При этом каждое имя будет уникальным, а набор символов перед римскими цифрами не должен повторять существующие команды Л^AT_EX.

Команда `\newArray{<имя массива>}` предназначена для определения нового массива. При ее выполнении автоматически создается индивидуальный счетчик с именем `arraycount<имя определяемого массива>`, начальное значение которого устанавливается равным единице. Параметром команды является имя нового массива, которое подчиняется тем же требованиям, что и имена счетчиков и макрокоманд. Приводим определение команды.

```
\def\newArray#1{\newcounter{arraycount#1}\setcounter{arraycount#1}{1}}
```

Для объявления нового массива необходимо записать:

```
\newArray{<имя массива>}
```

Следующая команда, `\AddItem{<имя массива>}{<элемент>}` предоставляет возможность добавлять элементы в уже объявленный массив. Каждый элемент массива будет храниться в новой команде, имя которой подчиняется следующим правилам:

`array<имя массива><значение счетчика римскими цифрами>`

Таким образом обеспечивается индивидуальность для имени каждого элемента. Например для массива с именем `qvery` имена макрокоманд для 1 - го, 2 - го и 3 - го элементов будут соответственно: `arrayqveryi`, `arrayqveryii` и `arrayqveryiii`. При выполнении команды счетчик, связанный с массивом будет увеличиваться на единицу. Команда имеет следующее определение.

```
\long\def\AddItem#1#2{%\expandafter\gdef\csname array#1\roman{arraycount#1}\endcsname{#2}\global\addtocounter{arraycount#1}{1}}
```

Рассмотрим отдельные элементы языка T_EX, входящие в определение. Команда `\csname... \endcsname`

определяет новую командную последовательность без содержания, которую позже можно переопределить. Команда `\expandafter` указывает T_EXу сначала раскрыть последовательность, определяющую имя новой макрокоманды, а затем определить ее глобально при помощи `\gdef`. Команда `\roman{arraycount#1}` выводит значение счетчика, связанного с

массивом римскими цифрами. общий синтаксис команды:

`\roman{<имя счетчика>}`.

Для получения элемента массива по его номеру определена команда

`\GetItem{<имя массива>}{<элемент>}`,

имеющая два параметра: имя массива и номер элемента.

```
\long\def\GetItem#1#2{%
  \setcounter{temparraycount}{#2}
  \ifnum\arabic{temparraycount}<\arabic{arraycount#1}
    \csname array#1\roman{temparraycount}\endcsname
  \fi
}
```

В определении команды используется вспомогательный счетчик **temparraycount**, определенный ранее. Использован условный оператор `\ifnum<Команды>\fi`, позволяющий выполнять команду при условии, что номер вызываемого элемента не превышает число элементов массива. выполняется сравнение значений счетчиков:

temparraycount и **arraycount**<имя массива>,

представленные арабскими цифрами при помощи команды:

`\arabic{<имя счетчика>}`.

Число элементов массива на единицу меньше значения счетчика **arraycount#1**. Определение для команды `\ArrayCount{<имя массива>}` получения числа элементов массива представлено ниже:

```
\def\ArrayCount#1{%
  \setcounter{temparraycount}{\arabic{arraycount#1}}
  \global\addtocounter{temparraycount}{-1}
  \arabic{temparraycount}
  \global\addtocounter{temparraycount}{1}
}
```

Также необходимыми являются команды перечисления всех элементов массива. Их две. Одна является аналогом команды **foreach...in...** Она имеет два параметра: имя массива и команду, которую надо исполнить над каждым элементом при его выводе.

```
\long\def\foreachArray#1#2{%
  \setcounter{temparraycount}{1}
  \loop\ifnum\arabic{temparraycount}<\arabic{arraycount#1}
    #2{\GetItem{#1}{\arabic{temparraycount}}}\fi
    \global\addtocounter{temparraycount}{1}
  \repeat
}
```

Подобная ей команда является аналогом цикла **for...from...to**. она имеет четыре параметра: имя массива, номер первого выводимого элемента, номер последнего выводимого элемента, команда исполняемая над каждым элементом массива при выводе.

```
\long\def\ForFromArray#1#2#3#4{%
  \setcounter{temparraycount}{#2}
  \setcounter{temparraycounta}{#3}
  \ifnum\arabic{temparraycounta}<\arabic{arraycount#1}
    \addtocounter{temparraycounta}{1}
  \else
    \setcounter{temparraycounta}{\arabic{arraycount#1}}
  \fi
}
```

```

\loop\ifnum\arabic{temparraycount}<\arabic{temparraycounta}
  #4{\GetItem{#1}{\arabic{temparraycount}}}}
  \global\addtocounter{temparraycount}{1}
\repeat
}

```

В двух последних определениях использована команда `\loop\if<условие><тело цикла>... \repeat` позволяет использовать цикл. Тело цикла выполняется пока истинно условие после `\if`.

2 Примеры использования массивов команд

Рассмотрим различные примеры использования предложенных команд. Для этого в преамбуле документа подключаем файл с определениями.

```
\input{texarray}
```

Определяем два массива: **mytestq** и **mytestw**

```

\newArray{mytestq}
\newArray{mytestw}

```

Добавляем элементы в оба массива.

```

\AddItem{mytestq}{фрагмент 1}
\AddItem{mytestw}{фрагмент 2}
\AddItem{mytestq}{фрагмент 3}
\AddItem{mytestw}{фрагмент 4}
\AddItem{mytestw}{фрагмент 5}
\AddItem{mytestw}{фрагмент 6}
\AddItem{mytestw}{фрагмент 7}

```

Выводим второй элемент массива **mytestq**: фрагмент 3

```
\GetItem{mytestq}{2}
```

Количество элементов в массиве **mytestq** равно: 2

```
\ArrayCount{mytestq}
```

Количество элементов в массиве **mytestw** равно: 5

```
\ArrayCount{mytestw}
```

Перечисляем все элементы массива **mytestw**:

```
\foreachArray{mytestw}{{}}
```

Результат: фрагмент 2 фрагмент 4 фрагмент 5 фрагмент 6 фрагмент 7

Добавляем еще один элемент в массив **mytestq**:

```
\AddItem{mytestq}{фрагмент 8}
```

Перечисляем все элементы массива **mytestq**

```
\foreachArray{mytestq}{{}}
```

Результат: фрагмент 1 фрагмент 3 фрагмент 8

Перечисляем все элементы массива **mytestq** с выполнением команды указанной во 2-м параметре над каждым элементом массива. Для этого определим команду, выводящую текст жирным курсивом.

```
\def\testcommand#1{\textbf{\textit{#1}}}
```

Применим указанную команду при вызове функции перечисления элементов массива.

```
\foreachArray{mytestq}{\testcommand}
```

Результат: ***фрагмент 1 фрагмент 3 фрагмент 8***

Перечисляем элементы массива **mytestw** со 2-го по 4-й.

```
\ForFromArray{mytestw}{2}{4}{}
```

Результат: фрагмент 4 фрагмент 5 фрагмент 6

Выводим элементы массива **mytestw** со 2-го по 4-й с выполнением команды над каждым элементом массива.

```
\ForFromArray{mytestw}{2}{4}{\testcommand}
```

Результат: ***фрагмент 4 фрагмент 5 фрагмент 6***

3 Применение массивов для создания тестов

Приводимые ранее команды являются достаточно общими, что позволяет их использовать в составе других команд. Рассмотрим одно из их практических применений для верстки тестовых заданий.

3.1 Счетчики и вспомогательные определения

Для работы с макрокомандами определим несколько счетчиков:

BiletNumber Счетчик номер билета

VoprosNumber Счетчик для вопросов

VarOtvNumber Счетчик вариантов ответов

Установим указанные счетчики равными 1 при помощи команд.

```
\newcounter{BiletNumber}
\setcounter{BiletNumber}{1}
\newcounter{VoprosNumber}
\setcounter{VoprosNumber}{1}
\newcounter{VarOtvNumber}
\setcounter{VarOtvNumber}{1}
```

3.2 Команды для разметки исходных вопросов

Для работы с тестами мы используем уже определенные нами команды для работы с массивами. Определим новый массив.

```
\newArray{qvery}
```

Определим команду для обозначения вопроса, добавляющую текстовый фрагмент в массив **qvery**, основанную на команде **\AddItem**.

```
\long\def\Qvery #1\endQvery{%
  \AddItem{qvery}{#1}
}
```

К вопросам обозначенным при помощи **\Qvery #1\endQvery**, можно обратиться по номеру, соответствующему их появлению в тексте, при помощи другой команды

```
\printQ{<номер вопроса>},
```

основанной на использовании команды:

```
\GetItem{qvery}{<номер вопроса>}.
```

```
\long\def\printQ#1{%
  \setcounter{VarOtvNumber}{1}
  \par\noindent\arabic{VoprosNumber}. %
  \GetItem{qvery}{#1}
  \addtocounter{VoprosNumber}{1}%
}
```

В результате выполнения команды печатается элемент массива **qvery** с номером указанным в параметре. Пример выполнения указанной команды (рисунок 1).

Рис. 1: Отдельный вопрос, выведенный командой **printQ**

Вариант ответа обозначается командой:

```
\long\def\VarQveryOtv{%
  \par\indent\ \arabic{VarOtvNumber}) %
  \addtocounter{VarOtvNumber}{1}%
}
```

3.3 Команды, определяющие вид печатного документа

Здесь рассмотрено несколько команд, которые имеют вспомогательное значение. Они могут быть использованы при необходимости.

Значение переменной **printotv** определяет, помечать ли в конечном документе правильные ответы, если используется команды для обозначения правильных вариантов ответов. Если значение **printotv** равно единице, то правильные ответы отмечают. Значение переменной **prnsimple** определяет, будут ли печататься все заголовки, определенные для теста. Если значение **prnsimple** равно единице, то печатаются только основные заголовки.

```
\newcount\printotv
\printotv=0
\newcount\prnsimple
\prnsimple=0
```

Значения указанных переменных для любой части документа можно установить командами:

`\printsimple` Печатать без заголовков.

`\printall` Печатать с заголовками.

`\setprintrtrueotv` Команда помечать правильные ответы при печати

`\setnopintrtrueotv` Команда не помечать правильные ответы при печати

`\trueotv` Маркер правильных ответов

Все вышеперечисленные команды действительны для оставшейся части текста, пока не встретится противоположная команда. Ниже приведены определения указанных команд.

```
\def\printsimple{\prnsimple=1}
\def\printall{\prnsimple=0}
\def\setprintrtrueotv{\printotv=1}
\def\setnopintrtrueotv{\printotv=0}
\def\trueotv{\ifnum\printotv=1$\bullet $\ \fi}
```

Кроме того, вид печатного документа определяют переменные шаблона, определяющие последующий текст, которые можно задать в любом месте документа. Дальнейшее рассмотрение ведется на примере экзаменационных билетов. Соответственно определены следующие переменные: Нижеприведенные команды позволяют определить ряд строк в заголовке, например в экзаменационных билетах это строки: «Утверждаю», «Предмет», «Специальность», «Зав каф», «Составил». Данные команды определяют в свою очередь новые команды, используемые внутри шаблона.

```
\def\templateutv#1{\edef\ltemplateutv{#1}}
\def\templatepredm#1{\edef\ltemplatepredm{#1}}
\def\templatespez#1{\edef\ltemplatespez{#1}}
\def\templatezavkaf#1{\edef\ltemplatezavkaf{#1}}
\def\templatesost#1{\edef\ltemplatesost{#1}}
```

Аналогично определяется ряд других параметров теста, таких как название подразделения, фамилия руководителя, дата утверждения, название организации, фамилия составителя, заголовок теста, название предмета, шифр специальности.

```
\def\kafedra#1{\edef\lkafedra{#1}}
\def\zavkaf#1{\edef\lzavkaf{#1}}
\def\datautv#1{\edef\ldatautv{#1}}
\def\vuz#1{\edef\lvuz{#1}}
\def\biletttitle#1{\edef\lbiletttitle{#1}}
\def\biletauthor#1{\edef\lbiletauthor{#1}}
\def\predmet#1{\edef\lpredmet{#1}}
\def\spez#1{\edef\lspez{#1}}
```

Переопределение соответствующих строк шаблона может быть сделанов любом месте документа и имеет силу пока не встретится новое определение.

Выполняется начальная инициализация переменных до определения шаблонов и окружений, где они используются.

```
\templateutv{УТВЕРЖДАЮ}
\templatepredm{Предмет}
\templatespez{Специальность}
\templatezavkaf{Зав.кафедрой}
```

```

\templatesost{Составил}
\kafedra{<Название кафедры>}
\zavkaf{<Фамилия И.О.>}
\datautv{<чч.мм.гг>}
\vuz{<Название ВУЗа>}
\biletttitle{<Заголовок билета>}
\biletauthor{<Фамилия И.О.>}
\predmet{<Название предмета>}
\spez{<Специальности>}

```

3.4 Команды для компоновки отдельных тестов

Для работы с отдельными тестами предусмотрена команда для заголовка теста **\BiletHeader**.

Результат ее выполнения зависит от переменных шаблона. Происходит установка счетчика номеров вопросов **BiletNumber** равным единице, а счетчик номеров билета **VoprosNumber** увеличивается на единицу.

```

\newcommand{\BiletHeader}{%
  \ifnum\prnsimple=0
    \nopagebreak \par
    \noindent\textbf{\ltemplateutv}
    \vspace{0.4cm}\nopagebreak
    \par
    \ \ltemplatezavkaf <<\lkafedra >>
    \par%
    \rule[-1pt]{2 cm}{0.4pt} \lzavkaf
    \vspace{0.2cm}\nopagebreak
    \par%
    \ \ltemplatepredm : <<\lpredmet >>\nopagebreak \par
    \ \ltemplatespez : \lspez\nopagebreak \par
    \ldatautv
    \vspace{0.1cm}\nopagebreak \par
    \begin{center}
      \textbf{\lvuz}\nopagebreak
    \end{center}
  \fi
  \begin{center}
    \textbf{\lbiletttitle \No \arabic{BiletNumber}}\nopagebreak
    \vspace{0.2cm}\nopagebreak
    \addtocounter{BiletNumber}{1}
    \setcounter{VoprosNumber}{1}
    \nopagebreak
  \end{center}
}

```

Команда для нижней части теста **\BiletFooter**. Здесь только команды форматирования, выводимые в зависимости от условий.

```

\newcommand{\BiletFooter}{%
  \ifnum\prnsimple=0
    \vspace{0.4cm}\nopagebreak \par
    \noindent\ltemplatesost

```



```

\rule[-1pt]{1.5cm}{0.4pt}
\lbletauthor \nopagebreak
\vspace{0.1cm}
\fi
\par
}

```

Окружение для билета, в котором используются обе вышеприведенные команды для заголовка и нижней части билета.

```
\newenvironment{EkzBilet}{\BiletHeader}{\BiletFooter }
```

Результат работы данной команды представлен на рисунке

Окружение для списка вопросов

```

\newenvironment{ListQvery}{%
  \par\noindent\textbf{\ltemplateutv}
  \vspace{0.4cm}
  \\
  \ \ltemplatezavkaf <<\lkafedra >> \rule[-1pt]{3cm}{0.4pt} \lzavkaf
  \vspace{0.2cm}
  \\
  \begin{center}
  \textbf{Список вопросов по курсу <<\lpredmet >>}\\
  для специальности \lspez
  \end{center}
  \setcounter{VoprosNumber}{1}
}{\BiletFooter}

```

Данное окружение использовано в команде для распечатки списка вопросов.

```

\def\makelistqvery{%
  \begin{ListQvery}
    \setcounter{VoprosNumber}{1}
    \foreachArray{qvery}{\par\noindent\arabic{VoprosNumber}.\%
      \addtocounter{VoprosNumber}{1}}
  \end{ListQvery}
}

```

Используется команда, распечатывающая все элементы массива **qvery**. Перед распечаткой очередного элемента выполняется команда, распечатать порядковый номер вопроса и увеличить его на единицу.

```
\par\noindent\arabic{VoprosNumber}. \addtocounter{VoprosNumber}{1}
```

Результат работы команды представлен на рисунке 2. Для компоновки отдельного теста можно воспользоваться как вышеприведенными окружениями, так и командой **\printbilet**, параметром которой являются номера вопросов, включаемых в тест, разделенные командой **\and**. Результат работы команды представлен на рисунке 3.

```

\long\def\printbilet #1{
  \let\and=\printQ %
  \begin{EkzBilet}
    \printQ #1
  \end{EkzBilet}
}

```

Рис. 2: Список вопросов, выведенный командой **makelistqvery**

Рис. 3: Билет распечатанный при помощи окружения **EkzBilet** или при помощи команды **printbilet**

3.5 Команды для компоновки тестов

Для компоновки блоков из 2-х и 4-х тестов используется окружение **tabular**. Определена команда **\twotestblok** для компоновки блока из 2-х тестов, которая имеет два параметра, задаваемых по аналогии с параметром команды **\printbilet**. Результат работы команды представлен на рисунке 4.

```
\long\def\twotestblok#1#2{
  \begin{tabular}{p{0.9\linewidth}}
    \hline \ \ \printbilet{#1}\ \ \hline \ \ \printbilet{#2} \ \ \hline %
  \end{tabular}
}
```

Определена команда **\fiertestblok** для компоновки блока из 4-х тестов, которая имеет четыре параметра, задаваемых по аналогии с параметром команды **\printbilet**. Результат работы команды представлен на рисунке 5.

```
\long\def\fiertestblok#1#2#3#4{
  \small
  \begin{tabular}{p{0.45\linewidth}|p{0.45\linewidth}}
    \hline \ \ \printbilet{#1} & \printbilet{#2} \ \ \hline \ \ %
    \printbilet{#3} & \printbilet{#4} \ \ \hline
  \end{tabular}
}
```

Рис. 4: Блок из двух билетов, полученный при помощи команды **twotestblok**

Рис. 5: Блок из четырех билетов, полученный при помощи команды **fiertestblok**

Все рассмотренные выше команды собраны в отдельный файл **testsmakr**, подключаемый командой **\input{testsmakr}**. Приведем пример рабочего файла, при компиляции которого получены результаты представленные на рисунках.

```
\input{testsmakr}%Определения документа
\kafedra{название}
\zavkaf{Ф.И.О.}
\datautv{01.01.2012}
\vuz{название}
\biletttitle{Билет}
\biletauthor{Ф.И.О.}
\predmet{предмет}
\spez{00.00.00}
\begin{document}
%Определяем сами вопросы, которые при этом на печать не выводятся
\Qvery 1-й вопрос\endQvery
\Qvery 2-й вопрос\endQvery
\Qvery 3-й вопрос\endQvery
\Qvery 4-й вопрос\endQvery
\Qvery 5-й вопрос\endQvery
\Qvery 6-й вопрос\endQvery
\Qvery 7-й вопрос\endQvery
\Qvery 9-й вопрос\endQvery
```

```

\Qvery 10-й вопрос\endQvery
\printQ{6} %Вывод на печать 6-го вопроса
\makelistqvery{} %Создать список вопросов
\newpage %Билет с помощью окружения EkzBilet
\begin{EkzBilet}
\printQ{4}
\printQ{7}
\printQ{1}
\printQ{4}
\end{EkzBilet}
%Билет с помощью команды \printbilet с теми же вопросами
\printbilet{{1}\and{7}\and{1}\and{4}}
\newpage %блок из двух билетов
\twotestblok{{4}\and{1}}{{5}\and{4}}
\newpage %блок из четырех билетов
\fiertestblok{{8}\and{3}}{{9}\and{2}}{{2}\and{3}}{{3}\and{2}}
\end{document}

```

В данном примере определены десять вопросов, которые используются при выводе тестов при помощи рассмотренных в статье команд.

Заключение

Вместо множества команд для вставки каждого фрагмента пользователю достаточно запомнить лишь шесть:

1. команды ограничителя начала и конца запоминаемого фрагмента;
2. команды вставки отдельного фрагмента;
3. команды вставки всех фрагментов или их части;
4. команды вставки блоков из двух или четырех билетов;
5. два новых окружения, специализированных для подготовки тестов: список вопросов и билет.

Команды и окружения, указанные в 4-м и 5-м пунктах, в отличие от пунктов 1-3 специализированны для подготовки тестовых заданий. Кроме того, при выводе всех фрагментов имеется возможность выполнения произвольного форматирования над каждым фрагментом.

Преимуществом подобного подхода является возможность для автора сосредоточиться на содержании самих вопросов, а не на их оформлении при компоновке тестов. Значительно упрощается корректировка вопросов. Править необходимо только в одном месте без использования поиска и автозамены, как при использовании копирования – вставки из буфера обмена в Microsoft Word.

Результатом использования рассмотренных в статье макрокоманд является повышение эффективности работы за счет переложения на компьютер рутинных задач.

Список литературы

- [1] Кнут Д. Всё про \TeX .— Протвино, R $\mathcal{D}\mathcal{T}\mathcal{E}\mathcal{X}$, 1993.
- [2] Львовский С.М. Набор и вёрстка в пакете $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$. — М., Космосинформ, 1994.
- [3] <http://www.tex.uniya.ru/> \TeX в ЯрГУ