



schors сегодня в 12:00

Пишем презентации в LaTeX

LaTeX, Конференции, Презентации

Tutorial

В процессе подготовки доклада на GolangConf'2019 я использовал LaTeX. Несмотря на некоторые устаревшие вещи, немного необычное поведение, скучную или сложную документацию, я внезапно получил удовольствие. Я не стал "гуру" LaTeX, но я смог делать весьма неплохие слайды. И я хочу описать основные точки создания современной презентации на LaTeX. Особенно это должно "зайти" айтишникам, которые по роду занятий программируют. Программировать свою презентацию — это увлекательно.

В самом начале

- 1970-1975. Томпсон и Ритчи создают UNIX и C
- В 80-е бурно развиваются компьютерные сети
- К концу 80-х некоторые считают, что UNIX морально устарел

5



Почему LaTeX:

- Гибкие оверлеи (когда информация на слайде появляется по очереди)
- Повторяемость и однозначность
- Пишем как программу. Некоторые ресурсы можно хранить отдельными файлами — код, список ссылок

Почему НЕ LaTeX:

- На выходе только PDF
- На тонкую подстройку таблиц и колонок, если они есть, может уйти больше времени, чем подвигать мышкой. Не надо делать презентацию в пять сложных слайдов в LaTeX. А в 55 — надо

Начнем с установки

Практически, сейчас существует один лидирующий дистрибутив утилит LaTeX/TeX, который поглотил все остальные — TexLive. Выпуски там нумеруются по годам. Не надо ставить археологические версии ранее 2018 года. Так что почти всем

пользователям Ubuntu сразу [сюда](#). Для "маководов" есть отдельный набор. Или можно использовать универсальный установщик, в том числе и [для Windows](#).

Пользователи CentOS должны страдать.

В любом случае можно создать свой Dockerfile и использовать его. Но учтите, что некоторые утилиты захотят видеть системные шрифты.

Дистрибутив TexLive увесистый, занимает до 5Gb. Ставить надо texlive-full или подобный пакет (название может разниться от дистрибутива к дистрибутиву).

Однако, на дворе XXI век, и вы можете воспользоваться одним из онлайн-сервисов для работы с LaTeX. Почему-то немногие даже научные работники об этом знают. Например, [один](#) из самых продвинутых.

Немного о LaTeX

Это не пошаговое руководство, как начать работать в LaTeX. Мы поверхностно посмотрим, как сделать презентацию, а по пути вы сами воспользуетесь Google. Но базовые отсылки я буду делать. Также обратите внимание на викикнигу.

Так будет выглядеть простейшая презентация с использованием пакета beamer — именно он отвечает за презентации:

```
\documentclass{beamer}% тип документа
% далее идёт преамбула
\title{My Super Topic}
\author{Ashot and Alice}

\begin{document}% начало презентации

\begin{frame}% первый слайд
\titlepage
\end{frame}

\begin{frame}% второй слайд
Hello, World!
\end{frame}

\end{document}
```

Поехали

А вот так выглядит готовая презентация, которую я сделал на GolangConf2019. Вокруг неё мы и будем крутиться.

Первое, что вы должны сделать, это забыть как страшный сон все отсылки к шрифтам TeX. Дональд Кнут начал разрабатывать систему шрифтов METAFONT в 1977 году, а в 1979 представил её. Это было ОЧЕНЬ круто на то время. Но на дворе 2019 год, тащить в сегодня стандарты наименований файлов 8+3 и специальные растровые шрифты можно только в минуты скуки. Все современные системы работают с современными векторными шрифтами. Утилиты TexLive имеют в своём составе новый компилятор документов — **xelatex**. Он полностью совместим с pdflatex, но работает с системными шрифтами. Всё что нужно дополнительно сделать, это подгрузить пакет fontspec или xltextra (он подгружает fontspec) и задать шрифты:

```
% !TeX TS-program = xelatex

\documentclass[aspectratio=169]{beamer}

\usepackage{xltextra}
\usepackage[main=russian,english]{babel}
```

```
\setmainfont{Arial}
\setromanfont{Times New Roman}
\setsansfont{Arial}
\setmonofont{Courier New}
...
```

В примере мы сразу задали размер презентации (16см на 9см), сделали комментарий чем собирать и подключили пакет babel, который мы разбирать не будем. Для пользователей Linux напомним, что в 2019 году TTF-шрифты можно кидать в директорию .fonts в домашнем каталоге.

Темы оформления

Можно делать презентацию и без специального оформления, но чувство прекрасного по умолчанию в beamer странное, поэтому практически все пользуются теми или иными темами. Обычно, темы оформления — это набор файлов .sty, размещенных там, где их может найти компилятор, т.е. там же, где и сам файл презентации. Подключается тема соответствующей строкой в преамбуле презентации:

```
\usetheme{gcr2019}% в скобках название темы
```

Есть, кстати, чудесное руководство по созданию простейшей темы оформления презентации. Я сделал 3 темы на ближайшие конференции Олега Бунина, одну для себя, две — для иллюстрации к этой статье: [раз](#), [два](#) и [три](#).

В принципе, всё что определено в этих темах, может быть и обычной простыней в вашем файле презентации. В этом нет ничего плохого. Темы нужны, когда нужна повторяемость. В каждой из приведенных выше тем оформления есть пример использования и скомпилированный результат.

Оверлей

Оверлей — это когда у вас информация постепенно появляется на одном и том же слайде. Вот например:

```
\begin{frame}
  \begin{itemize}
    \item<1-> Живи, безумец!.. Трать, пока богат!
    \item<2-> Ведь ты же сам – не драгоценный клад.
    \item<2-> И не мечтай – не сговорятся воры
    \item<3-> Тебя из гроба вытащить назад!
  \end{itemize}
\end{frame}
```

Сначала появится первая строка, затем вторая и третья, а затем четвертая. Заодно мы увидели, как сделать нумерованный список. Систему оверлеев можно прикрутить практически к чему угодно. Я, например, делал так подсветку строк кода (рассмотрим ниже).

Обратите внимание на строки в моих шаблонах:

```
\setbeamercolor{itemize item}{fg=black}
\setbeamercolor{itemize subitem}{fg=black}
\setbeamercolor{alerted text}{fg=black}
\setbeamerfont{itemize items}{\textbullet}% отключает кошмарные синие треугольники
\setbeamerfont*{itemize/enumerate subbody}{parent=itemize/enumerate body}
\setbeamerfont*{itemize/enumerate subsubbody}{parent=itemize/enumerate subbody}
\setbeamerfont{alerted text}{series=\bfseries}
```

Закомментируйте строки (для комментариев используйте символ "%") и посмотрите, что будет. Не тот цвет, синие треугольнички и вот это вот всё.

Не нашел куда это определить. beamer предоставляет возможность несколько раз использовать один и тот же слайд. Это вот прямо очень круто:

```
\begin{frame}[label=again]{Слайд покажем ещё раз}
...
\end{frame}
...
\againframe{again}
```

Картинки

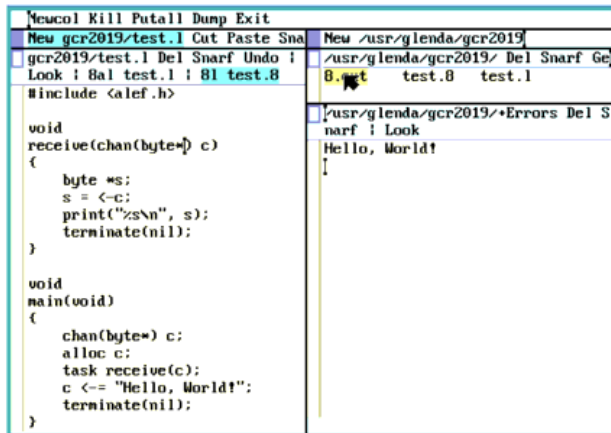
Возьмем пример из моей презентации. Две колонки, слева нумерованный список, справа — картинка:

```
\begin{frame}{Редактор Асме}
  \begin{columns}[T,onlytextwidth]
    \begin{column}{0.44\textwidth}
      \begin{itemize}
        \item Для разработчиков
        \item Автор --- \emph{Роб Пайк}
        \item Написан на Alef
        \item Alef = Асме
        \item 30 лет с нами
      \end{itemize}
    \end{column}
    \begin{column}{0.56\textwidth}
      \includegraphics[width=\textwidth]{acme-alef.png}
    \end{column}
  \end{columns}
\end{frame}
```

Параметр [T,onlytextwidth] говорит о том, что вертикальное выравнивание производится по базовой линии строк вверх, а блок колонок имеют ширину равной ширине фрейма для текста. LaTeX сам сожмет картинку как надо. Можете создать свой пример и поиграться параметрами, чтобы увидеть, как и что будет перекрываться.

Редактор Асме

- Для разработчиков
- Автор — *Роб Пайк*
- Написан на Alef
- Alef = Асме
- 30 лет с нами



The screenshot shows the Aasm editor window. The menu bar includes: New, Col, Kill, Putall, Dump, Exit. The toolbar has icons for New, gcr2019/test.1, Cut, Paste, Snarf, Del, Snarf, Undo, Look, Bal, test.1, 81, test.8. The main text area contains a Go program with syntax highlighting: `#include <alef.h>`, `void receive(chan(byte*) c) { byte *s; s = <-c; print("%s\n", s); terminate(nil); }`, `void main(void) { chan(byte*) c; alloc c; task receive(c); c <-= "Hello, World!"; terminate(nil); }`. The right sidebar shows the file list: `/usr/gclenda/gcr2019/`, `Errors`, `Del S`, `narf`, `Look`, `Hello, World!`.

9



Листинг кода

Сразу оговорюсь, что подсветка синтаксиса в листинге кода в презентации не нужна. Если она там вам нужна — вы что-то делаете не так. Больше нужны выделения. Я использую стандартный пакет `listings`, который умеет подсветку синтаксиса, и не умеет выделений. Вы можете посмотреть настройки оформления блока листинга в любом из моих примеров:

```
\definecolor{acmebg}{HTML}{FFFFFF}
\definecolor{acme1}{HTML}{52AAAD}
\lstset{
    columns=flexible,
    keepspaces=true,
    showstringspaces=false,
    showtabs=false,
    tabsize=4,
    frame=single,
    basicstyle=\fontsize{10pt}{12}\bf\ttfamily\color{black},
    backgroundcolor=\color{acmebg},
    commentstyle=\color{black},
    keywordstyle=\color{black},
    stringstyle=\color{red},
    rulecolor=\color{acme1},
    framerule=1pt
}
```

Для выделения фона я пользуюсь пакетом `lstlinebgd`, который "заброшен". Приходится вставлять исправляющий код. А вот этот дополнительный код даёт мне команду `\btLstHL`, которая позволяет подсвечивать строки в оверлее:

```
\begin{frame}[fragile]{Limbo. Примеры синтаксиса}
    \begin{lstlisting}[basicstyle=\fontsize{12pt}{12}\bf\ttfamily\color{black},linebackgroundcolor={%
        \btLstHL<1>{1}%
        \btLstHL<2>{2-3}%
        \btLstHL<3>{5,6}%
        \btLstHL<4>{8,9}%
        \btLstHL<5>{11}%
    }]
i, j: int;
k := 1;      # int
\end{lstlisting}
\end{frame}
```

```

b := byte 2;    # byte

s: string;      # длина в символах
s1 := s[0:];    # срез

a: array of int;
a = array[10] of int;

myfunc: fn(i, k: int, s: string) : (list of string, int);
    \end{lstlisting}
\end{frame}

```

Limbo. Примеры синтаксиса

```

i, j: int;
k := 1;      # int
b := byte 2; # byte

s: string;    # длина в символах
s1 := s[0:];  # срез

a: array of int;
a = array[10] of int;

myfunc: fn(i, k: int, s: string) : (list of string, int);

```

30



Библиография

Для простоты и универсальности я использую пакет biblatex. Вот тут есть сложность. Пакет beamer перекрывает часть умолчаний biblatex. Вот эти команды убирают иконки и задают шрифт библиографии:

```

\setbeamertemplate{bibliography item}{\insertbiblabel}
\renewcommand*{\bibfont}{\fontsize{8}{1}\selectfont}

```

Надо помнить важный момент — библиография задаётся в отдельном файле. В примерах в шаблонах я пишу библиографию в основном тексте с помощью пакета filecontents, который в свою очередь создаёт нужный реальный файл. И компиляция происходит в ЧЕТЫРЕ шага. Сначала просто xelatex компилирует файл и создаёт вспомогательные файлы. Затем утилита biber компилирует файл библиографии, затем надо ещё дважды прогнать xelatex. Да, с препроцессорами там всё сложно.

Пакет biblatex тоже настраивается. Но на первое время хватит того, что есть у меня в примерах.

Послесловие

Я получил удовольствие и от подготовки презентации, и от подготовки материалов к этой статье. Я надеюсь, если кто-то хотел, но не мог решиться начать использовать LaTeX в своей работе — я сломал этот барьер. Конечно, я не сделал пакетов, я не копал в глубину, я не упомянул важные темы рисования. Но я думаю, что это уже всё придёт, как только вы начнете пользоваться инструментом. Неплохие практические статьи есть и на Хабре по тегу #latex.