

文档制作

曹东刚

caodg@sei.pku.edu.cn

Linux 程序设计环境

<http://c.pku.edu.cn/>



内容提要

1 排版系统

2 L^AT_EX

3 绘图

4 Docbook

5 程序文档

6 幻灯片

版面设计

- 版面设计是一门艺术. 文档的可理解性和可读性非常重要
 - 结构清晰: 标题序号与字体大小要适当
 - 页面美观: 行宽适当
- 作者、图书设计者、排版者
 - 作者: 交付图书原稿给出版公司
 - 图书设计者: 决定整本书的版面形式 (栏宽、字体、行距、标题前后间距等), 给出排版说明
 - 排版者: 根据排版说明进行排版

电子排版系统

- 所见即所得 (WYSIWYG) 的文字处理软件
 - M\$ Word, WPS, Open Office, Word Perfect, 等等
- 格式化排版程序
 - 首先编辑一个文本的源文件
 - 文档正文部分 (作者)
 - 文档结构描述部分 (图书设计者)
 - 然后用排版程序对源文件进行格式化处理 (排版者)
 - 将结果输出到输出设备, 如监视器或打印机等

Unix 排版系统

- troff, groff
- texinfo
- T_EX 与 L^AT_EX
- docbook
- 轻量标记语言: asciidoc, Markdown, reStructuredText

内容提要

① 排版系统

② L^AT_EX

③ 绘图

④ Docbook

⑤ 程序文档

⑥ 幻灯片

L^AT_EX 排版的公式

$$\begin{aligned}
 |I_1| &= \left| \int_{\Omega} g R u \, d\Omega \right| \\
 &\leq C_3 \left[\int_{\Omega} \left(\int_a^x g(\xi, t) \, d\xi \right)^2 \, d\Omega \right]^{1/2} \\
 &\quad \times \left[\int_{\Omega} \left\{ u_x^2 + \frac{1}{k} \left(\int_a^x c u_t \, d\xi \right)^2 \right\} \, d\Omega \right]^{1/2} \\
 &\leq C_4 \left\| f \left| \tilde{S}_{a,-}^{-1,0} W_2(\Omega, \Gamma_l) \right| \right\| \left\| |u| \overset{\circ}{\rightarrow} W_2^{\tilde{A}}(\Omega; \Gamma_r, T) \right\|.
 \end{aligned} \tag{1}$$

$$\begin{aligned}
 |I_2| &= \left| \int_0^T \psi(t) \left\{ u(a, t) - \int_{\gamma(t)}^a \frac{d\theta}{k(\theta, t)} \int_a^{\theta} c(\xi) u_t(\xi, t) \, d\xi \right\} \, dt \right| \\
 &\leq C_6 \left\| f \int_{\Omega} \left| \tilde{S}_{a,-}^{-1,0} W_2(\Omega, \Gamma_l) \right| \right\| \left\| |u| \overset{\circ}{\rightarrow} W_2^{\tilde{A}}(\Omega; \Gamma_r, T) \right\|.
 \end{aligned} \tag{2}$$

T_EX 与 Donald E. Knuth

Stanford 大学的计算机科学家 Donald E. Knuth (高德纳) 在上世纪 70 年代设计了 T_EX 和 Metafont



T_EX 历史与现状

- 1978: T_EX 第一版面世, 用 Pascal 书写, 输出与设备无关
- 1982: T_EX 稳定版本, 自此之后极少修订
- 1993: Knuth 宣布冻结 T_EX 与 Metafont, 不再进行更新
- 1995 至今: 只发现一个 T_EX 的排版错误, 非软件错误

T_EX 与 Metafont 最新版本:

- T_EX 版本: 3.141592, 收敛到 π
- Metafont 版本: 2.718, 收敛到 e

T_EX 扩展

最基本的 T_EX 程序只是由一些很原始的命令组成, 难于使用

- Plain T_EX: Knuth 设计了一个名叫 Plain T_EX 的基本格式
 - Plain T_EX 的重点还只是关注于如何排版
- AMS T_EX 与 AMS L^AT_EX: 美国数学协会提供的, 基于 Plain T_EX
 - 用于排版含有很多数学符号和公式的科技类文章或书籍
- L^AT_EX: Leslie Lamport 1985 年开发, 最流行和使用最为广泛
 - 构筑在 Plain T_EX 的基础之上, 并加进了很多的功能
 - 基本上不需要使用者自己设计命令和宏等, 易学易用
 - L^AT_EX2.09, L^AT_EX3, L^AT_EX2_ε

T_EX/L^AT_EX 的不足

- 不适合没有灵魂、没有思想的人¹
- 不能马上完全学会, 除非你是一个天才
- 即使开端良好, 也难以掌握其精髓
- 当发生错误的时候, 对新手而言排错可能是很困难的
- 设计一个新的版面非常困难
- 不适合排版非结构化的、无序的文档

¹“L^AT_EX does not work well for people who have sold their souls ...”

T_EX/L_AT_EX 的优势

- 高质量、专业级的输出
- 记住几个说明文档逻辑结构的命令即可应付多数情况
- 适合具有良好结构的文档
- 可很容易地生成目录、索引、脚注、引用等复杂结构
- 是可编程的, 易于定制与扩展
- T_EX 引擎是免费、开源的, 具有超常的稳定性
- 良好的通用性与跨平台性
- 超级技术支持, 众多的免费资源
- 一种乐趣

接受 L^AT_EX 的出版社

几乎所有的世界一流学术出版机构, 包括

- AMS
- Elsevier
- Kluwer Academic Publishers
- Cambridge University Press
- Springer
- IEEE
- ...

T_EX 的未来

下面列出了几个正在进行中的对 T_EX 进行扩展的项目：

- PDF_TE_X：完全兼容标准的 T_EX，但能够给出 PDF 输出
- e-T_EX：不仅完全兼容标准的 T_EX，还支持一种“扩展模式”
- Omega：几乎是完全重新写过的、支持 Unicode 的 T_EX 程序
- NTS：NTS 代表“New Typesetting System”，将来可能取代 T_EX 或 e-T_EX
- XeT_EX：内置 Unicode 支持的 L^AT_EX
- LuaT_EX：基于 PDF_TE_X，以 Lua 为嵌入式脚本语言，目标是增加开放和扩展性

L^AT_EX vs Word

- 易用: Word (所见即所得)
- 稳定美观: L^AT_EX
- 办公文档: Word
- 科技论文: L^AT_EX
- 短文: Word
- 复杂结构的长文档: L^AT_EX

L^AT_EX 学习要点

- 找一本详尽的使用手册, 通读所有内容, 了解 L^AT_EX 的全部功能, 但不必记住所有的命令
- 尽早使用各种高级功能, 如定理环境、公式、索引、交叉引用等
- 尽量选取最恰当的命令实现所需要的功能
- 尽量复用成熟的格式文件, 如各种科技论文宏包, 学位论文宏包等
- 选择一个或几个好的画图工具

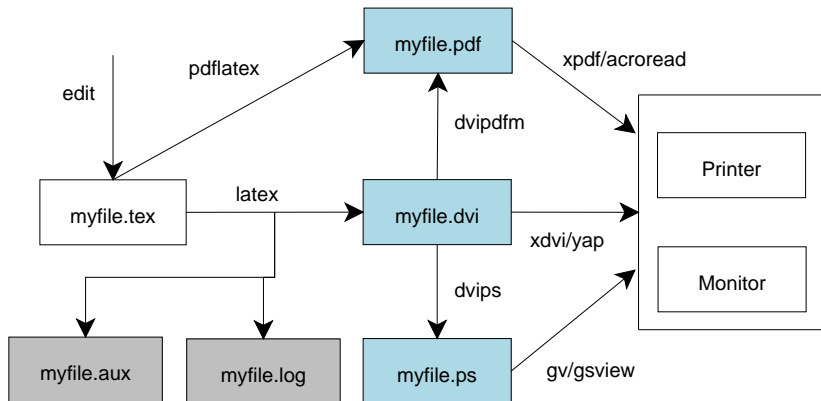
L^AT_EX 系统

- web2c: C 语言实现的 T_EX 系统, GPL 版权保护
 - 很多重要的 free L^AT_EX 实现都基于 web2c
- teTeX: 基于 web2c, 大多数 Unix/Linux 系统的缺省安装
- web2c-win32: 即 ftTeX, web2c 的 Win32 实现
- MikTeX: Windows 下最好用的 L^AT_EX 实现, 更新很快
- Tex Live: <http://tug.org/texlive/>, 主流发行版
 - 基于 teTeX 与 web2c-win32, 支持 Unix/Linux/Mac OS/Windows
- CTeX 与 ChinaTeX: 集成 MikTeX, 增加中文支持 (CCT 与 CJK)

中文支持

- CCT: 由中科院计算所张林波等设计
 - 在文档格式方面相当符合中文习惯
 - 与 L^AT_EX 的辅助程序兼容性较差
 - 最新版本的 CCT 兼容性有很大提高
- CJK: 由欧洲人 (Werner Lemberg) 设计
 - 与 L^AT_EX 的辅助程序兼容性相当好
 - 在文档格式方面的中文化较差
 - 逐渐成为中文处理的主流

L^AT_EX 文档处理步骤



各类常见 L^AT_EX 文件

.tex	T _E X 或 L ^A T _E X 源文件, 可用 latex 处理
.sty	L ^A T _E X 宏包文件, 可用\usepackage 加载
.dtx	文档化的 L ^A T _E X 文件, L ^A T _E X 宏包发布的主要格式
.ins	相应的.dtx 文件的安装文件. 可用 latex 对.ins 文件处理 从.dtx 中提取出宏包

.dvi	设备无关文件, 可预览排版效果
.log	记录编译的详细信息
.toc	记录所有章节的标题, 第二次编译时读入生成目录
.aux	记录交叉引用等信息, 用于向第二次编译传递信息
.idx	记录所有的索引词条, 需用 makeindex 生成
.lof	记录所有的图形目录

PS 文件与 PDF 文件

- 从 tex 文件生成 pdf 文件:

```
caodg@debian:~$ pdflatex myfile.tex
```

- 从 dvi 文件生成 ps 文件: caodg@debian:~\$ dvips myfile

- 从 ps 文件生成 pdf 文件:

```
caodg@debian:~$ ps2pdf myfile.ps
```

- 查看 ps 文件: caodg@debian:~\$ gv myfile.ps

- 查看 pdf 文件: caodg@debian:~\$ xpdf myfile.ps

查看 ps 文件与 pdf 文件的工具有多种

L^AT_EX 编辑工具

- vim + latexsuite
- emacs
- lyx (WYSIWYG)
- WinEdt (commercial)

L^AT_EX 源文件

为普通的 ASCII 文件, 包含待排版的文本和如何排版的命令

- 空白距离: 空格、制表符等空白字符被视为相同的空白距离
多个连续空白字符等同于一个空白字符

空行标志段落结束; 多个连续空行等同一个空行

- 特殊字符: # \$ % ^ & _ { } ~ \

除\外, 上述其它字符皆可通过前面加反斜杠得到

- 命令: 大小写敏感, 以反斜杠开始加上只包含字母字符命令名

如, `\TeX` `\emph{emphasize}` `\today`

- 注释: 百分号% 后面的字符、分行符、下一行开始的空白字符全部视为注释

文档和语言的结构

文档的主旨在于向读者传递观点、信息和知识, 如果排版风格反映了内容的逻辑和语义结构, 读者就会抓住文章脉络.

L^AT_EX 只需要你告诉它文档的逻辑和语义结构

- 段落: L^AT_EX 中最重要的文档单位
- 句子: L^AT_EX 中较小的文档单位
 - 句子也有结构, 谨慎使用标点符号
- 逻辑结构: 章、节、子节等高级结构

最简单的 L^AT_EX 源文件

```
\documentclass{article}  
\begin{document}  
Hello world!  
\end{document}
```

L^AT_EX 源文件结构

```
\documentclass[选项]{文档类}  
\usepackage[options]{package}  
全局命令与定义  
\begin{document}  
文本与局部命令  
\end{document}
```

- 导言 (preamble): `\begin{document}` 之前的内容, 对宏包的引用放在导言区
- 正文 (body): `\begin{document}` 之后的内容

文档类

L^AT_EX 处理源文件时,首先需要知道要处理的文档类型,例

```
\documentclass[11pt,twoside,a4paper]{article}
```

文档类:

- article : 排版科技文章、短报告、程序文档等
- report : 排版多章的长报告、短篇书籍、博士论文等
- book : 排版书籍
- frames : 排版幻灯片

文档类选项

- 10pt(缺省), 11pt, 12pt
- a4paper, b5paper, letterpaper(缺省)
- fleqn : 数学公式左对齐, 而不是中间对齐
- leqno : 公式编号居左
- titlepage, notitlepage : 文档标题后是否分页
- onecolumn, twocolumn : 单列或双列排版
- twoside, oneside : 双面或单面格式
- openright, openany : 新章在奇数页开始还是下一页开始

篇章结构示例

```
\documentclass[a4paper,11pt]{article}
\usepackage{amsmath}
\title{Demonstrate \LaTeX Structure}
\author{Donggang Cao}
\begin{document}
% generates the title
\maketitle
% insert the table of contents
\tableofcontents
\section{Begin}
Well, and here begins my lovely article.
\subsection{Theorem One}
\section{Conclusion}
Here it ends.
\end{document}
```

Demonstrate L^AT_EX Structure

Donggang Cao

May 23, 2006

1 Begin

Well, and here begins this lovely article.

1.1 Theorem One

Here proves theorem one.

2 Conclusion

Here it ends.

大型文档

处理大型文档如书籍、博士论文时, 最好将源文件分为几个部分

- 方法 1: `include` 和 `includeonly` 命令

在正文区: `\include{filename}`, 在新页开始排版

在导言区: `\includeonly{filename}`

只有 `includeonly` 声明的文件会被包含进来, 可用于大型文档的草稿调试阶段

- 方法 2: `\input{filename}` 命令, 在当前位置排版
既可在正文区, 也可在导言区, 常用于包含设置文件等

环境

一个环境用命令`\begin{环境}` 开始, 以`\end{环境}` 结束

环境的作用在于: 其内的文本要根据环境参数进行不同的处理, 有可能暂时改变处理特征, 如行距、字体、缩进等. 这些改变只在环境内起作用

常见的环境

- `itemize`, `enumerate`, `description`
- `flushleft`, `flushright`, `center`
- `quote`, `quotation`, `verse`
- `verb`, `verbatim`

列表环境

```
\begin{enumerate}
\item You can mix the list environments to your taste:
\begin{itemize}
\item But it might start to look silly.
\item[-] With a dash.
\end{itemize}
\item Therefore remember:
\begin{description}
\item[Stupid] things will not become smart because they are in a list.
\item[Smart] things, though, can be presented beautifully in a list.
\end{description}
\end{enumerate}
```

列表环境输出

① You can mix the list environments to your taste:

- But it might start to look silly.
 - With a dash.

② Therefore remember:

Stupid things will not become smart because they are in a list.

Smart things, though, can be presented beautifully in a list.

数学公式

文档结构是 L^AT_EX 的灵魂; 数学是 T_EX 的灵魂

L^AT_EX 用数学模式来排版数学符号和公式, 以区分通常的文本模式

- 数学模式中的空格和分行都被忽略. 空格要么是公式的逻辑衍生物, 要么通过命令得到
- 数学模式不允许有空行, 每个公式中只能有一个段落
- 每个字符都被看作变量名进行排版. 如果要插入文本, 需要用命令输入

如果希望在当前行排版数学公式, 可用\$公式内容\$ 界定

公式示例

```
\begin{displaymath}
\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

输出结果:

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

图形

L^AT_EX 可以直接生成简单的图形, 包括文本、各种斜率的直线、箭头、矩形、椭圆形等

图形环境

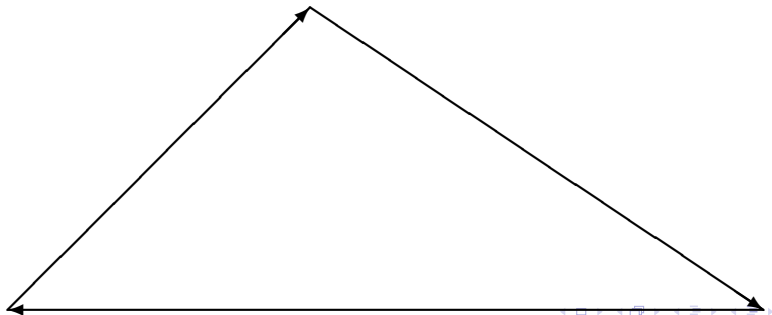
```
\begin{picture}(x尺寸, y尺寸)
```

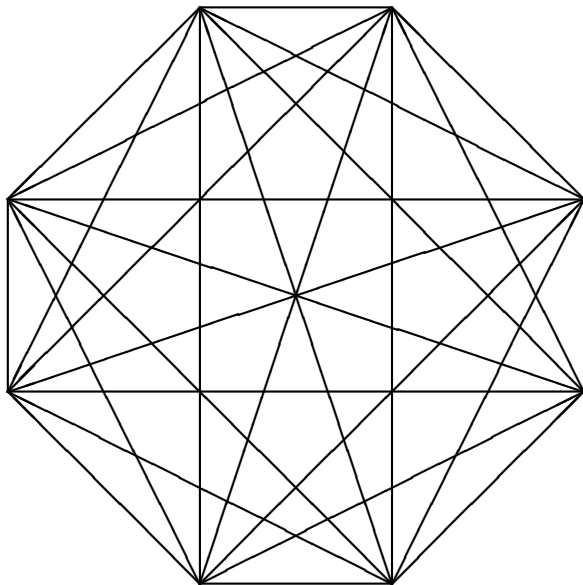
画图命令

```
\end{picture}
```

图形示例

```
\setlength{\unitlength}{2cm}  
\begin{picture}(5,2)\thicklines  
  \put(5,0){\vector(-1,0){5}}  
  \put(0,0){\vector(1,1){2}}  
  \put(2,2){\vector(3,-2){3}}  
\end{picture}
```





正文内的引用

- 交叉引用

首先标记: `\label{chap01:sec1:fig:myfig}` 引用:

`\ref{chap01:sec1:fig:myfig}`

- 参考文献的引用

首先生成参考文献

引用: `\cite[附加信息]{关键词}`

生成参考文献

最好的方法是通过参考文献数据库生成, 运行命令**bibtex**

- 首先, 声明样式\ bibliographystyle{样式}
常用样式
 - plain : 文献按字母顺序排列
 - unsrt : 文献按\cite 命令出现的先后排列
- 然后, 引用文献数据库\ bibliography{数据库文件}
数据库文件在计算机上以.bib 为扩展名, 此处只给出主干名

文献数据库

参考文献数据库条目示例:

```
@book{Haters94,  
  author      = "Simson Garfinkel and Daniel Weise",  
  title       = "The UNIX-HATERS Handbook",  
  publisher    = "IDG Books",  
  address     = "California",  
  year        = "1994",  
  url         = "http://www.simson.net/ref/ugh.pdf"  
};
```

用 L^AT_EX 制作幻灯

推荐工具

- beamer
- pdfscreen + ppower4
- foiltex
- prosper
- texpower

内容提要

① 排版系统

② L^AT_EX

③ 绘图

④ Docbook

⑤ 程序文档

⑥ 幻灯片

L^AT_EX 插图

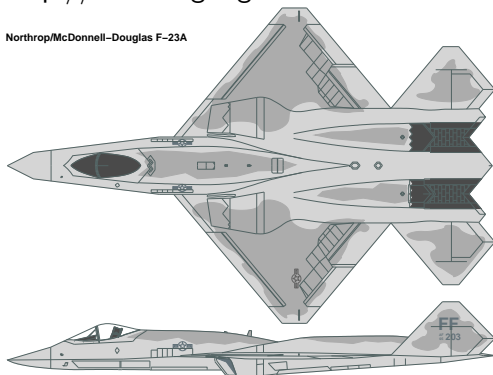
- 当 Knuth 编写 T_EX 的时候, 还没有 PostScript/EPS, JPEG, GIF 等, 因此 DVI 并不直接支持这些格式的图形. 不过, T_EX 允许 DVI 文件中包含 **special** 命令来向 DVI 处理程序传递命令
- 1994 年发布的 L^AT_EX 2_ε 的图形宏包套件增加了对图形的支持
- 由于 DVI 文件经常被转为 PS 文件, 所以支持最好的是对 EPS 格式 (Encapsulated PostScript, 是 PS 语言的子集) 的图像

xfig

xfig是功能非常强大的画矢量图的工具, 可以方便的导出为各种图形格式. 缺点在于绘图习惯大多数人不适应

<http://www.xfig.org>

Northrop/McDonnell-Douglas F-23A

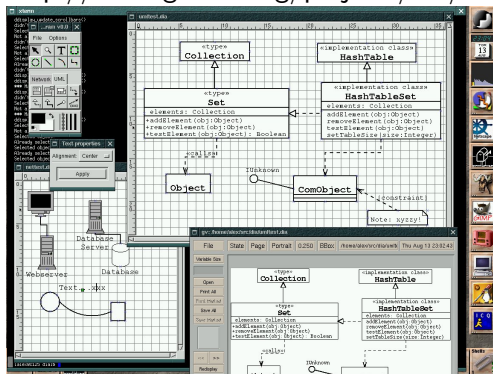


Distribution of this artwork as part of the xfig package, where xfig is part of a commercially sold software package is permitted. © 1997, Carlo Kopp, Hsin-Ho Wu

dia

dia是基于 GTK 的绘图工具, 类似于 windows 下的visio

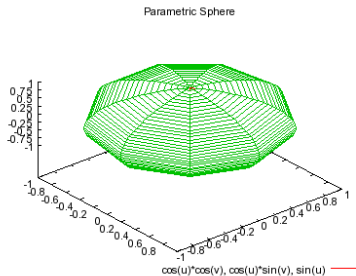
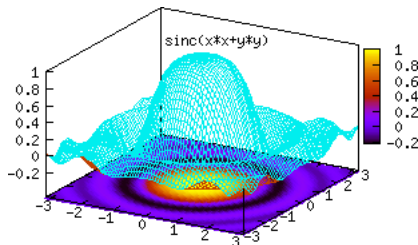
<http://www.gnome.org/projects/dia/>



Gnuplot

gnuplot是将数据和函数转换为专业的图表的工具, 可以实现比 M\$ excel 强大的多的绘图功能

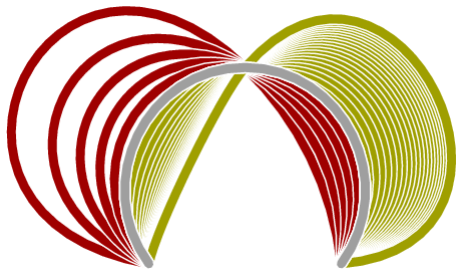
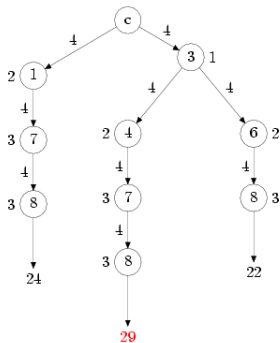
<http://www.gnuplot.info/>



MetaPost

MetaPost 是基于 Knuth 的 Metafont 的绘图语言, 可以产生 PostScript 输出. 唯一的缺点就是太强大、太精确了

<http://www.tug.org/metapost.html>



graphviz

Graphviz (Graph Visualization Software 的缩写) 是一个由 AT&T 实验室启动的开源工具包, 用于绘制 DOT 语言脚本描述的图形。Graphviz 是一个自由软件, 其授权为 Common Public License。

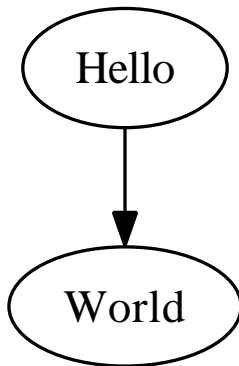
graphviz 构架

Graphviz 由一种被称为 DOT 语言的图形描述语言与一组可以生成和/或处理 DOT 文件的工具组成：

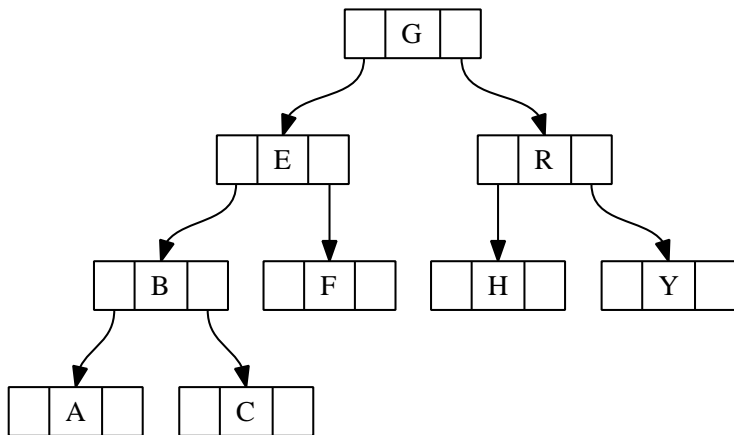
- dot：一个用来将生成的图形转换成多种输出格式的命令行工具。其输出格式包括 PostScript, PDF, SVG, PNG, 含注解的文本等等。
- neato：用于 sprint model 的生成。
- twopi：用于放射状图形的生成。
- circo：用于圆形图形的生成。
- fdp：另一个用于生成无向图的工具。
- dotty：一个用于可视化与修改图形的图形用户界面程序。
- lefty：一个可编程的控件，可以显示 DOT 图形。

graphviz 示例: Hello World

```
echo "digraph G {Hello->World}" | dot -Tpng >hello.png
```



graphviz 示例：数据结构 - 树



graphviz 示例: 数据结构 - 树

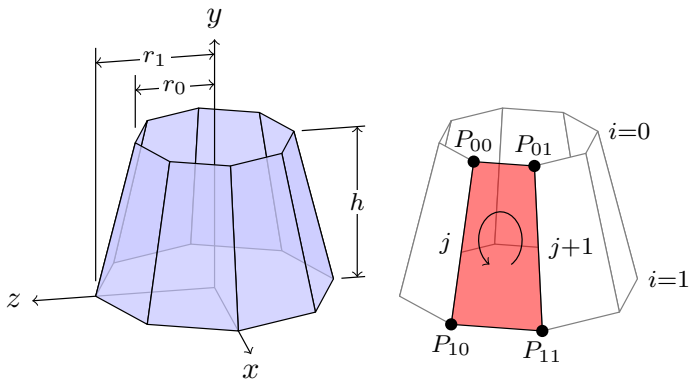
```

digraph g {
    node [shape = record,height=.1];
    node0[label = "<f0> |<f1> G|<f2> "];
    node1[label = "<f0> |<f1> E|<f2> "];
    node2[label = "<f0> |<f1> B|<f2> "];
    node3[label = "<f0> |<f1> F|<f2> "];
    node4[label = "<f0> |<f1> R|<f2> "];
    node5[label = "<f0> |<f1> H|<f2> "];
    node6[label = "<f0> |<f1> Y|<f2> "];
    node7[label = "<f0> |<f1> A|<f2> "];
    node8[label = "<f0> |<f1> C|<f2> "];
    "node0":f2 -> "node4":f1;
    "node0":f0 -> "node1":f1;
    "node1":f0 -> "node2":f1;
    "node1":f2 -> "node3":f1;
    "node2":f2 -> "node8":f1;
    "node2":f0 -> "node7":f1;
    "node4":f2 -> "node6":f1;
    "node4":f0 -> "node5":f1;
}

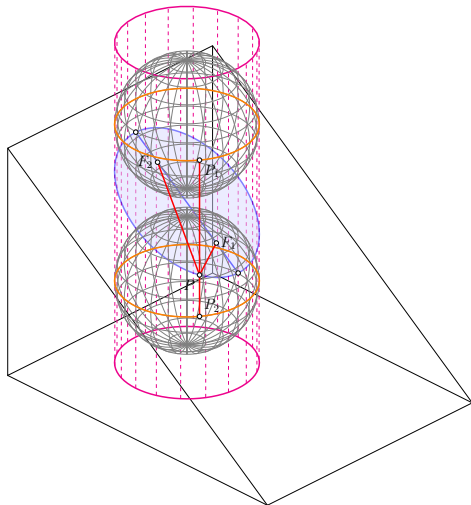
```

在 L^AT_EX 直接绘图: Tikz

示例见: <http://www.texample.net/tikz/examples/>



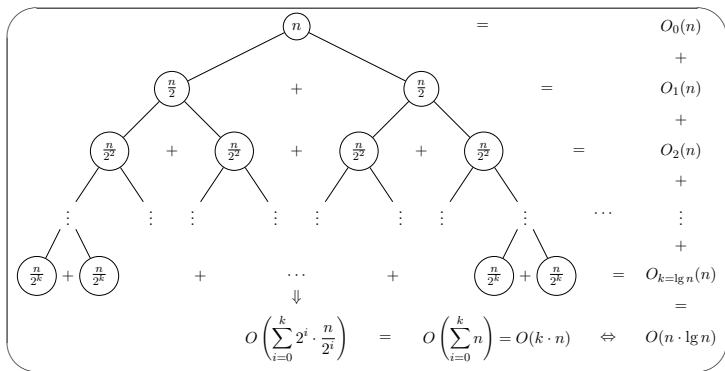
在 L^AT_EX 直接绘图: Tikz



在 L^AT_EX 直接绘图: Tikz



在 L^AT_EX 直接绘图: Tikz



内容提要

① 排版系统

② L^AT_EX

③ 绘图

④ Docbook

⑤ 程序文档

⑥ 幻灯片

Docbook 是一种基于 XML 的新式文档工具

- 很多开源软件的标准文档工具
- 文档结构 DTD 和样式单
- 各种 XML 处理工具的支持
- 适合项目文档, 不适合学术报告

内容提要

① 排版系统

② L^AT_EX

③ 绘图

④ Docbook

⑤ 程序文档

⑥ 幻灯片

Doxygen

doxygen 是一个自动文档生成系统，它对使用 C++, C, Java, Objective-C, Python, IDL, PHP, C# 等语言编写的代码及注释进行分析，自动生成说明文档。Doxygen 支持多种格式的说明文档，包括 html, RTF(Word), Latex, XML 等。

DoxygenToolkit.vim

VIM 插件，自动生成

- License 模板: DoxLic
- Author 模板: DoxAu
- 注释模板: Dox

Doxygen 示例

```
/* Copyright (C)
 * BSD blablabla...
 */

/**
 * @file A.java
 * @brief bababa
 * @author Donggang Cao
 * @version 1.0
 * @date 2011-05-15
 */

public class A {
    /**
     * @brief main : blabla
     *
     * @param args : blabla
     *
     * @return : blabla
     */
    public static void main(String[] args) { }
}
```


内容提要

① 排版系统

② L^AT_EX

③ 绘图

④ Docbook

⑤ 程序文档

⑥ 幻灯片

幻灯片制作

传统工具

PowerPoint, Keynote, Beamer, Google Presentation

HTML 展示

- Presi, Google Presentation
- S5, Slidy, Impress.js