

华昱资本
同创价值

XX 公司 XX 项目 投资价值分析报告

版本: 1.0



方莲

2014 年 5 月

XX 公司 XX 项目 投资价值分析报告

版本: 1.0

方 莲

2014 年 5 月

HY CAPITAL INVESTMENT GP
华昱资本管理公司

© 2008–2013 Alpha Huang
Alpha Culture & Education ®

All right reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher, except by a 雷人.

The author and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories, technologies and programs to determine their effectiveness. The author and publisher make no warranty of any kind, express or implied, with regard to these techniques or programs contained in this book. The author and publisher shall not be liable in any event of incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these techniques or programs.

Printed in the United States of America

目录



插图目录



表格目录



摘 要



包太雷同志是一个高尚的人,一个纯粹的人,一个有道德的人,一个脱离了低级趣味的人,一个有益于人民的人。

— 白求恩

一个人偶尔雷人容易,难的是一辈子雷人。

— 雷锋

书中自有颜如玉。

— 如花

包老师帅得惊动了党中央。

— 犀利

LXJX。

— 蓝翔技校

第一章 简介



滚滚长江东逝水,浪花淘尽英雄。是非成败转头空。青山依旧在,
几度夕阳红。

白发渔樵江渚上,惯看秋月春风。一壶浊酒喜相逢。古今多少事,
都付笑谈中。

— 杨慎《临江仙》

排版是人类生活中一项很重要的工作,也是传统印刷和电脑出版的核心活动。在一定的版面内摆放不同型态的对象(如数字、文字、表格和图形等),以合适的方法表现渲染它们,这个过程就是排版。

排版的版面可以有固定尺寸的印刷品,也可以是较为灵活的电脑软件、电子文档,还可以是狂野奔放的网页。

1.1 历史回顾

排版按照历史时期和技术方法可以划分为传统排版和数字排版两大类。数字排版的一个重要概念是光栅图像处理器 (raster image processor, RIP)。RIP 出现之前的印刷排版历史可参阅 ??。

RIP 生成的点阵图像可以输出到激光照排机或直接制版机等制版设备,也可以输出到打印机;它的输入可以是页面描述语言 (page description language, PDL),也可以是和输出设备分辨率不同的点阵图像。

RIP 可以是硬件、固件或软件。硬件 RIP 用于高档排版设备,1976 年莫诺公司 (Monotype Corp.) 的激光照排机 Lasercomp 就配有硬件 RIP。固件 RIP 在打印机内置微处理器上运行,每一台 PostScript (PS) 打印机都配有固件 RIP,比如 1985 年苹果公司的 LaserWriter。最早的软件 RIP 是 1986 年的 Ghostscript。

顾名思义, 页面描述语言是用来描述待输出页面的, 它比二进制的图像数据高级一点, 但是正常人类看起来还是会很费劲。所以人们要在它前面再加一层标记语言 (markup language)。图像数据、页面描述语言、标记语言的关系大致可以比喻为机器码、汇编语言、高级语言。而在排版领域, \TeX 是最精确最强大的标记语言, 可谓公鸡中的战斗机。

数字排版的工作流程和主要工具见 ??。本节以下部分将分别回顾常见的页面描述语言、标记语言和 \TeX 。

表 1.1: 数字排版工作流程及主要工具

标记语言	→	PDL	→	RIP	→	输出设备
troff 系列		PS		硬件 RIP		激光照排机
SGML 系列		PDF		固件 RIP		直接制版机
\TeX 系列		DVI		软件 RIP		打印机

1.1.1 页面描述语言

PostScript

最早的行式打印机只能打印字符, 后来的针式打印机可以用点阵的方式画出字符, 也可以画出粗糙的图形。当时矢量图只能用绘图仪来打印。

1969 年施乐推出首台激光打印机之后, 就一直在想办法精确描述页面图像, 从而结合点阵打印机和绘图仪的优点, 同时打印高质量的图形和文字。1975 年 Robert F. Sproull¹ 主持开发了一种格式 Press, 后来用于施乐的 Xerox Star 电脑 (一种个人电脑的雏形)。但是 Press 只是一种格式而不是语言, 所以施乐启动了 InterPress 研究计划。

1976 年, 埃文斯·苏泽兰公司 (Evans & Sutherland) 的 John E. Warnock (1940–)² 正在酝酿一种图形设计语言, 也就是后来的 PostScript。1978 公司想让他从旧金山分部搬到犹他州总部, 他不想搬家就跳槽到了施乐。

埃文斯·苏泽兰的两位创始人 David C. Evans (1924–1998)³ 和 Ivan E. Sutherland

¹1967 年哈佛大学物理学士, 斯坦福大学计算机系 1970 年硕士, 1977 年博士。1973–75 年供职于施乐, 后跳槽到卡耐基梅隆大学任计算机系副教授。1980 年和他硕士导师 Ivan E. Sutherland 成立了一家咨询公司苏泽兰·斯普罗及其同伙 (Sutherland, Sproull and Associates); 1990 年这家公司被 Sun 收购, 师徒两位都接着给 Sun 工作; 2010 年 Sun 被 Oracle 收购, 他留下来继续从事研究工作。美国科学院和工程院院士。

²犹他大学数学系 1961 年学士, 1964 年硕士, 1969 年电脑博士。

³犹他物理系 1949 年学士, 1953 年博士。毕业后加入本迪克斯公司 (Bendix Corp.), 1962 年跳槽到犹他电子系, 1965 年创立犹他计算机系。



(1938-) ⁴ 都是犹他大学的教授,也是 Warnock 的博士导师。他们对施乐挖走得得意门生耿耿于怀。1980 年 Sutherland 从施乐挖走另一位弟子 Sproull,成立了那家咨询公司,后被 Sun 收购。

在施乐这边,Warnock 和 Martin Newell 开发了新的图形系统 JaM (John and Martin),它后来被合并到 InterPress 中去。这两位还开发过另一个系统 MaJ。

1982 年,Warnock 和同事 Charles M. Geschke (1939-) ⁵ 一起离开施乐,创立了 Adobe。Newell 后来也加入了 Adobe,于是施乐的 InterPress 胎死腹中。

1984 年 Adobe 发布 PostScript 后不久,乔帮主 (1955–2011) 跑来参观,并建议用它来驱动激光打印机。次年,苹果推出了配备 PS 解释器的 LaserWriter。之后 Adobe 分别于 1991 和 1997 年推出了 PostScript 2 和 PostScript 3,这三代 PostScript 都是当时的流行标准。据说 1980 年代 Adobe 的收入多数来自 PS 解释器的许可费。

1990 年代后期,廉价喷墨打印机的出现使得 PostScript 逐渐式微,因为 PS 解释器对它们毕竟是一个成本负担,另外 PostScript 过于复杂,对微处理器和内存要求都很高。

PDF

1993 年,Adobe 推出了另一种格式 portable document format (PDF),它在 2008 年成为开放标准 ISO 32000。除了开放,PDF 比起 PostScript 还有其他一些优势:

- PDF 基本上是 PostScript 的一个子集,因此更轻便。
- PDF 支持更先进的字体,具体见第三章。
- PDF 支持透明图形,还支持动画。
- PDF 支持加密等安全特性。

虽然拥有上述优势,PDF 最初的推广却并不顺利,因为其读写工具 Acrobat 太贵。后来 Adobe 推出了免费的 Acrobat Reader (后更名为 Adobe Reader),并不断改进 PDF,终于使它超越了 PostScript,成为网络时代电子文档的新标准。

⁴1959 年卡耐基梅隆电子学士,1960 年加州理工学院电子硕士,1963 年 MIT 电脑博士。1963 年国家安全局 (National Security Agency, NSA) 中尉,1964 年国防部高等研究计划局 (Defense Advanced Research Projects Agency, DARPA) 信息处理主管。1966 年哈佛电子系副教授,1968 年与其硕士生 Sproull 发明了虚拟现实技术。1968 年犹他计算机系教授,并与同事 Evans 创立埃文斯·苏泽兰公司。1974 年创立加州理工计算机系。1988 年获图灵奖 (Turing Award),1998 年获冯·诺依曼奖 (IEEE John von Neumann Medal)。

⁵泽维尔大学 (Xavier University) 1962 年西方古典学士,1963 年数学硕士,卡耐基梅隆 1972 年电脑博士。



其他页面描述语言

其他页面描述语言还有:

- 爱普生打印机标准码 (Epson standard code for printers, ESC/P) , 主要用于针式打印机。
- 惠普的打印机命令语言 (printer command language, PCL) , 主要用于喷墨打印机。
- 惠普图形语言 (HP graphics language, HPGL) , 主要用于绘图仪。
- \TeX 家族的设备独立文件格式 (device independent file format, DVI) , 详见 ?? 节。
- 微软的 XML 纸张规范 (XML paper specification, XPS) 。2009 年, 基于 XPS 的 Open XPS 被 Ecma 国际 (Ecma International) 批准为 ECMA 标准。

1.1.2 标记语言

troff 系列

1964 年, MIT 的 Jerome H. Saltzer (1939–)⁶ 在参与开发第一个分时操作系统时, 写了一个文本排版程序 RUNOFF。

后来贝尔实验室的 Robert H. Morris⁷ 把 RUNOFF 移植到 GE 635 上, 改名为 roff。1969 年, Malcolm D. McIlroy (1932–)⁸ 把 roff 用 BCPL 语言重写, 移植到 DEC PDP-7。

1971 年, 一小撮 Unix 开发人员想把他们的电脑升级到 PDP-11, 谎称升级的原因是要给贝尔的东家 AT&T 的专利部门做一个文件排版系统。技术员 Joseph E. Ossanna (1928–1977)⁹ 就把 roff 改写为 nroff。后来他们弄到一台王安公司 (Wang Laboratories) 的排版机 WANG Graphic Systems CAT, Ossanna 又把 nroff 用 PDP-11 的汇编语言改写为 troff。

领导说, 小同志啊, 现在 C 语言是三个代表的。Ossanna 没日没夜地把它用 C

⁶MIT 电子系 1961 年学士, 1963 年硕士, 1966 年博士, 同年留校任教, 1995 年退休。他是一位从一而终的典范。

⁷哈佛数学系 1957 年学士, 1958 年硕士。1960 年加入贝尔实验室, 1986 年跳槽到 NSA, 曾任 NSA 电脑安全中心首席科学家, 1995 年退休。他为 Unix 写了数学库函数。1988 年他儿子 Robert T. Morris (1965–) 在康奈尔大学念研究生时, 联网到 MIT 释放了第一个电脑蠕虫; 被判三年监禁, 400 小时社区劳动, 罚款一万。

⁸1954 年康奈尔工程物理学士, 1959 年 MIT 应用数学博士。1958 年加入贝尔实验室, 1997 年退休。曾任图灵奖委员会主席。

⁹1952 年毕业于韦恩州立大学 (Wayne State University) 。



又写了一遍,7000 行不带注释的程序。领导又说,小同志啊,CAT 已经不时髦了,我们要高瞻远瞩放眼 21 世纪。Ossanna 又去加班,累倒在工作岗位上,终年 49 岁。据说他在弥留之际的遗言是,“同志们,不要管我,抢救公社的……要紧!”包子曰:四流大学小本在大公司里真的很难混啊。

关键时刻,白求恩的老乡 Brian W. Kernighan (1942-) ¹⁰ 挺身而出,把 troff 改写成独立于排版机的 ditroff,还发表了一篇技术报告, *A Typesetter-independent TROFF*。包子曰:编程、灌水两手抓,两手都要硬。

troff 包含一系列命令,用来设置字体、间距、段落、旁注、脚注等,能够将字符在页面上任意定位,甚至重叠。当遇到复杂任务时,人们还可以利用 Unix 的管道功能,将 troff 和其他程序结合使用。

1990 年代贝尔实验室大分家,Unix 部分卖给了 Novell,其他部分大多重组进朗讯科技。troff 也就花落几家,重组后的贝尔一份,朗讯一份,Sun 的创始人之一 Bill N. Joy (1954-) ¹¹ 一份,GNU 也搞了一个 groff。人心散了,队伍不好带了。

目前 troff 还是 Unix 手册的缺省文件格式,但是它在排版方面的领地已经基本被 L^AT_EX 和一些所见即所得软件瓜分。

SGML 系列

1969 年,IBM 的 Charles F. Goldfarb ¹² 和同事 Edward Mosher, Raymond Lorie 发明了通用标记语言 (generalized markup language, GML), GML 其实是他们三人姓氏的首字母。

GML 把结构、内容、格式分开,这样作者就可以专心写作。另外还可以为输出设备指定各自的特性文件 (profile),从而实现设备独立。

1978 年,Goldfarb 等开始改进 GML。1986 年标准通用标记语言 (standard generalized markup language, SGML) 成为国际标准 ISO 8879。SGML 提出了条目、标签、元素、属性、实体等语法概念,还增加了有效性检验机制。但是因为 SGML 过于复杂,一直没有流行起来,只有一些大公司或政府部门在使用。

1989 年,欧洲核子研究中心 (European Organization for Nuclear Research, CERN)

¹⁰1964 年多伦多大学工程物理学士,1969 年普林斯顿大学电子博士。2000 年离开贝尔实验室,加入普林斯顿。1977 年与同事 Alfred V. Aho (1941), Peter J. Weinberger 一起发明了 AWK 语言。1978 年与 C 语言发明人 Dennis M. Ritchie (1941–2011) 合写了 *The C Programming Language*。1990 年与同事 Robert Fourer, David M. Gay 发明了 AMPL 语言。

¹¹1975 年密歇根大学电子学士,1979 年伯克利大学电脑硕士。他开发了 vi, C Shell,还参与了 BSD Unix、网络文件系统协议 (Network File System, NFS) 的开发。

¹²1960 年哥伦比亚大学文学学士,1964 年哈佛法学博士。做了几年律师后于 1967 年加入 IBM,不知何年离开又变回律师。



的 Tim J. Berners-Lee (1955-) ¹³ 设计了超文本标记语言 (HyperText Markup Language, HTML), 并写了浏览器和服务端软件。他认为 HTML 是一种 SGML 的应用。Berners-Lee 被称为万维网 (world wide web) 之父。1994 年, Berners-Lee 纠集欧盟委员会 (European Commission) 和 DARPA, 在 MIT 创建了 W3C。

1993 年, HTML 成为互联网工程任务组 (Internet Engineering Task Force, IETF) 的建议标准。1995 年 IETF 发布了 HTML 2.0。1997 年, HTML 3.2 成为 W3C 推荐标准, 同年升级到 4.0, 次年升级到 4.01。

HTML 擅长表现布局、外观, 但是缺乏对内容数据的关怀, 所以人们从 1990 年代中期就开始寻求面向数据交换的新格式。1998 年 W3C 发布了在 SGML 基础上简化来的扩展标记语言 (Extensible Markup Language, XML), 2004 年升级为 XML 1.1。

从那时起, 基于 XML 的新名词儿如雨后春笋不断涌现。其中和排版关系比较大的是 1998 年 W3C 发布的数学标记语言 (mathematical markup language, MathML), 2003 年升级为 MathML 2.0。

1991 年豪电脑 (Hal Computer Systems) 和欧莱利出版公司 (O'Reilly Media) 推出了 DocBook, 一种面向技术文档的标记语言。最初它算是 SGML 的应用, XML 出现后改换门庭, 最新版本是 2009 年的 5.0。它采用了显示中性的方法, 用它编写的内容在出版时可以选择 HTML, PDF, CHM ¹⁴ 等多种格式, 而用户无须修改源码。

Scribe

1980 年, 卡耐基梅隆的 Brian K. Reid (1949-) ¹⁵ 提交了他的博士论文 *Scribe: A Document Specification Language and its Compiler*。1981 年, Reid 鼓动 Goldfarb 一起参加了一个会议, Goldfarb 发现 Scribe 和 SGML 在几处重要概念上是相似的, 可谓英雄所见略同。

Reid 在毕业时把 Scribe 卖给了一位教授 Michael I. Shamos (1947-) ¹⁶ 的统一

¹³1976 年牛津大学物理学士。1980 年成为 CERN 合同工, 1984 年转正。2004 年起任南安普敦大学电子系系主任。

¹⁴Compiled HTML Help, 微软的一种帮助文件格式, 它对 HTML 进行了索引、压缩。

¹⁵1970 年马里兰大学物理学士, 1980 年卡耐基梅隆电脑博士。同年加入斯坦福, 1987 年终身教职被拒, 跳槽到 DEC。1999 年跳到贝尔实验室, 2001 年跳到卡耐基梅隆。2002 年跳到 Google, 2004 年 Google 首次公开募股 (IPO) 前被解雇, 原因是他 too old。按 IPO 的价格, 他损失了 1000 万美元左右, 愤而起诉, 官司好像还没打完。

¹⁶1968 年普林斯顿物理学士, 1970 年瓦沙学院 (Vassar College) 物理硕士, 1972 年美国大学 (American University) 技术管理硕士, 耶鲁大学计算机系 1973, 74 年双硕士, 1978 年博士。1981 年杜肯大学 (Duquesne University) 法学博士。有有搞错, 七个学位。1975 年加入卡耐基梅隆, 历任数学、统计、电



逻辑公司 (Unilogic)。Shamos 跟卡耐基梅隆为 Scribe 的知识产权打了几年官司, 然后给它装上时间炸弹迫使用户交钱。Richard M. Stallman (1953-) ¹⁷ 谴责说这是对程序员精神的背叛, 是对人类的犯罪。Scribe 后来无疾而终。

1.1.3 T_EX 家族

包子曰: 自施乐以降, 豪杰并起, 跨州连郡者不可胜数。初 SGML 名微而众寡, 然遂能克 troff, 以弱为强者, 非惟天时, 抑亦人谋也。今 SGML 已拥百万之众, 挟互联网而令诸侯, 此诚不可与争锋。土坯据有 PDL, 已历三世, 国险而民附, 贤能为之用, 此可以为援而不可图也。纳德将军既帝室之胄, 信义著于四海, 总揽英雄, 思贤如渴, 若跨有公式、算法, 保其岩阻, 西和 DocBook, 南抚 Scribe, 外结好土坯, 内修政理; 天下有变, 则命一上将将公式之军以向 AMS, SIAM, 将军身率算法之众出于 TUG, 百姓孰敢不箪食壶浆, 以迎将军者乎? 诚如是, 则霸业可成, 雷太赫可兴矣。

引擎

谈到 T_EX, 人们首先会想起 Donald E. Knuth (1938-) ¹⁸。1962 年 Knuth 开始写一本关于编译器设计的书, 原计划是 12 章的单行本。不久 Knuth 觉得此书涉及的领域应该扩大, 于是越写越多, 如滔滔江水连绵不绝, 又如黄河泛滥一发不可收拾。1965 年完成的初稿居然有 3000 页, 据出版商估计, 这些手稿印刷出来需要 2000 页。出书的计划只好改为七卷, 每卷一或两章, 这就是 *The Art of Computer Programming* ¹⁹。

1976 年, 当 Knuth 改写第二卷的第二版时, 很郁闷地发现第一卷的铅版不见了; 而当时数字排版刚刚兴起, 质量还差强人意。于是 Knuth 仰天长啸: “我要扼住命运的咽喉”, 决定自己开发一个全新的排版系统, 这就是 T_EX。

1978 年 T_EX 第一版发布后好评如潮, Knuth 趁热打铁在 1982 年发布了第二版等系科助理教授、客座教授、图书馆馆长、研究所主任等职位。2001 年香港大学电子系访问教授。

¹⁷1971 年在哈佛念大一时, 高分通过了号称全美大学本科最难的数学课 Math 55。这门课的学生高考都是满分, 最后只有四分之一能及格, 通过者包括比尔·盖茨。1974 年获物理学士后, 入 MIT 念博士, 后为专心当黑客而退学。1970 年代开发了 Emacs, 1983 年创立 GNU, 1985 年创立自由软件基金会 (Free Software Foundation, FSF)。

¹⁸1960 年凯斯工学院数学学士, 因为长得帅同时获赠硕士, 在校期间曾加入黑社会外围组织 ΘX。1963 年加州理工数学博士, 同年留校任教。1968 年跳槽到斯坦福, 1974 年获图灵奖, 1992 年退休, 1995 年获冯·诺依曼奖。

¹⁹已完成的前三卷是: *Fundamental Algorithms*, *Seminumerical Algorithms*, *Sorting and Searching*。第四卷 *Combinatorial Algorithms* 的第一部分 4A 已出版, 其余部分和第五卷 *Syntactic Algorithms* 正在写作中, 预计 2020 年完成。第六卷 *Theory of Context-free Languages* 和第七卷 *Compiler Techniques* 尚未安排上工作日程。



版, 1989 年发布的 $\text{T}_{\text{E}}\text{X}$ 3.0 将 7 位字符改为 8 位。之后 Knuth 宣布除了修正漏洞停止 $\text{T}_{\text{E}}\text{X}$ 的开发, 因为它已经很稳定, 而且他要集中精力完成那部巨著的后几卷。

从那时起, 每发布一个修正版, 版本号就增加一位小数, 趋近于 π ; 当前版本是 2008 年的 3.1415926。他的另一个软件 METAFONT 的版本号趋近于 e , 目前是 2.718281。Knuth 希望在他离世时, $\text{T}_{\text{E}}\text{X}$ 和 METAFONT 的版本号永远固定下来, 从此人们不再改动他的代码。

Knuth 在软件工程方面也独树一帜。一般认为, 编程语言大致可以划分为四类: 机器语言、汇编语言、过程语言、面向对象语言。1970 年代时, 基于过程的结构化编程方法占据主导地方。它认为程序只应该包含顺序、分支、循环等三种结构, GOTO 跳转大大地不好, 应该禁止。而 Knuth 认为只要使用得当, GOTO 没什么不好的。

人们在编程时为了使程序清晰, 常常在代码间插入注释。Knuth 认为这不够人性化, 他主张按照程序员的思维逻辑, 在注释间插入代码, 这就是文学编程 (literate programming)。

包子曰: 于我心有戚戚焉。包老师写的东西往往注释连篇累牍, 八卦亦真亦幻, 正文不知所云。突然有一天, 正文不见了, 剩下的都是八卦, 老包便已登堂入室, 或可荣膺 Nobel Zhuangbility Prize。

1981 年, Knuth 提出了 WEB 编程系统, 它把程序分成编程和文档两部分, 编程部分采用 Pascal, 文档部分则用 $\text{T}_{\text{E}}\text{X}$ 。1987 年普林斯顿的 Silvio Levy (1959-) ²⁰ 提出 CWEB, 把编程部分换成 C 语言。

格式

$\text{T}_{\text{E}}\text{X}$ 是一种语言也是一个排版引擎 (engine), 引擎的基本功能就是把字排成行, 把行排成页, 涉及到断字、断行、分页等算法。基本的 $\text{T}_{\text{E}}\text{X}$ 系统只有 300 多个元命令 (primitive), 十分精悍, 但是很难读懂, 只适于非正常人类。所以 Knuth 提供了一种格式 (format, 宏命令的集合) 对 $\text{T}_{\text{E}}\text{X}$ 进行了封装, 这就是 Plain $\text{T}_{\text{E}}\text{X}$, 包含 600 多个宏命令, 然而它还是不够高级。

1980 年代初期, 斯坦福研究所 (Stanford Research Institute, SRI) 的 Leslie Lamport (1941-) ²¹ 开发了一种新的格式, 也就是 \LaTeX 。1992 年 \LaTeX 2.09 发布后,

²⁰1979 年巴西理论数学与应用数学研究所 (IMPA) 硕士。普林斯顿数学系 1985 年博士, 1986 年博士后。1988 年加入明尼苏达大学几何中心, 1995 年跳槽到陈省身创立的数学科学研究所 (Mathematical Sciences Research Institute, MSRI)。

²¹1960 年 MIT 数学学士, 布兰迪斯大学 (Brandeis University) 数学系 1963 年硕士, 1972 年博士。1970 年加入麻省计算机同伙公司 (Massachusetts Computer Associates, MCA), 1977 年跳槽到 SRI, 1985 年跳到 DEC, 2001 年跳到微软。2008 年获冯·诺依曼奖。



Lamport 退居二线,之后的开发活动由 Frank Mittelbach²² 等人接管。他们发布的最后版本是 1994 年的 $\text{\LaTeX} 2_{\epsilon}$, $\text{\LaTeX} 3$ 的开发也在进行中,只是正式版看起来遥遥无期。

宏包

\LaTeX 出现之后,在它的基础上出现了很多宏包 (package)。起初,美国数学会 (American Mathematical Society, AMS) 看着 \TeX 是好的,就派 Michael D. Spivak (1940–)²³ 开发基于 Plain \TeX 的宏包 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$, 它的开发进行了两年 (1983–1985)。后来与时俱进的 AMS 又看着 \LaTeX 是好的,就想转移阵地,但是他们的字体遇到了麻烦。

恰好 Mittelbach 和 Rainer Schöpf²⁴ 刚刚搞了个字体系统 new font selection scheme for \LaTeX (NFSS), AMS 看着还不错,就拜托他们把 AMSFonts 加入 \LaTeX , 继而在 1989 年请他们开发 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$ 。次年 $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$ 正式发布,之后它被整合为 $\mathcal{A}\mathcal{M}\mathcal{S}$ 宏包。

驱动

Knuth 最初设计的 \TeX 只能用于施乐图形打印机 (Xerox graphic printer, XGP), 这台打印机本身还需要一台 PDP-6 为它服务。1979 年, David R. Fuchs²⁵ 提出把 \TeX 的输出改为设备无关的格式, 也就是 DVI。

DVI 和其他页面描述语言的主要区别是, 它不能嵌入字体和图形。所以它只能称作准页面描述语言, 用户需要用驱动程序 (driver) 把它转换为其它格式, 比如 PostScript 或 PDF 等。

1985 年斯坦福电脑博士生 Tomas Rokicki²⁶ 在 Fuchs 和 Knuth 的帮助下开发了 dvips, 它把 DVI 转为 PostScript。后来维护它的是 \TeX 用户组 (TeX Users Group, TUG) 的 Karl Berry²⁷ 等人。

²²毕业于美因茨大学 (Johannes-Gutenberg University of Mainz)。1989 年加入电子数据系统公司 (Electronic Data Systems, EDS)。

²³1964 年普林斯顿数学博士。

²⁴ $\text{\LaTeX} 3$ 的开发者之一。

²⁵普林斯顿毕业后, 1978 年进入斯坦福攻读电脑博士。他不是 Knuth 的学生, 但是完成过一些 \TeX 的开发任务。在 Adobe 工作过一段时间, 后混入娱乐圈, 电影 *Red Diaper Baby* 和 *Haiku Tunnel* 的制片人。

²⁶1985 年毕业于得克萨斯农机大学后, 来到斯坦福念博士。博士毕业后加入惠普, 1999 年离开创业。

²⁷1980 年代中期就读于马萨诸塞大学。毕业后跟着 Stallman 在 FSF 混, 之后混迹于因特里弗 (Interleaf)、哈佛、直觉公司 (Intuit) 等地, 2003 年起任 TUG 总统。



1996 年, 俄罗斯高能物理研究所 (Institute for High Energy Physics) 的 Sergey Lesenko 发布了 `dvipdf`, 它把 DVI 转为 PDF。后来凯特林大学 (Kettering University) 的 Mark A. Wicks 开发的 `dvipdfm` 可以更好地处理字体、图形, 它在 2001 年之后基本停止开发。`dvipdfm` 只能处理单字节字符, 后来日本的平田俊作 (Shunsaku Hirata) 和韩国的赵珍焕 (Jin-Hwan Cho, 1968-) ²⁸ 分别把它扩展为 `dvipdfm-jpn` 和 `dvipdfm-kor`, 2002 年两者被合并为 `dvipdfmx`。它在字体、编码、图形方面都更优秀。

革命

1990 年代初, Knuth 和 Jiří Zlatuška (1957-) ²⁹, Philip Taylor (1947-) ³⁰ 等探讨过怎样改进扩展 $\text{T}_{\text{E}}\text{X}$, 但是 Knuth 坚持只有他才能改动 $\text{T}_{\text{E}}\text{X}$ 。1992 年这两位就纠集人马, 在一位匿名赞助者的支持下搞了个新项目 `new typesetting system (NTS)`, 企图改朝换代。他们不喜欢文学编程, 就派 Karel Skoupý 把 $\text{T}_{\text{E}}\text{X}$ 逆向工程, 还想换用 Java。

这次农民起义运动由于其历史局限性, 缺乏马克思主义的世界观、方法论和一般原理的引导, 同时遭到统治阶级的镇压, 最终失败了。但是它留下了革命的火种: $\epsilon\text{-T}_{\text{E}}\text{X}$, 把 $\text{T}_{\text{E}}\text{X}$ 支持的字符从 256 个扩充到 32768 个。

1994 年, Zlatuška 让他的学生 Hàn Thê Thành (1972-) ³¹ 试试改进 $\text{T}_{\text{E}}\text{X}$ 直接生成 PDF, 这就是后来的 `pdfTEX`。1996 年马萨里克大学授予 Knuth 荣誉数学博士, 他亲切地接见了 Hàn, 并拍着他的小脑袋说, $\text{T}_{\text{E}}\text{X}$ 要从娃娃抓起。

Hàn 回越南期间, 上网不便, 其他一些人参与了 `pdfTEX` 的开发。2009 年底开发组决定除了修正漏洞外基本停止 `pdfTEX` 的开发, 因为他们要转向一种新的引擎, `LuaTEX`。目前 `LuaTEX` 还处于测试阶段。

$\text{T}_{\text{E}}\text{X}$ 和 `pdfTEX` 的字体配置过程都很繁琐。2004 年初 Jonathan Kew ³² 发布的 `XYTEX` 支持 Unicode 字符集和 AAT 字体, 但是只能用于 Mac OS X。2005 年加入了对 OpenType 字体的支持, 2006 年移植到 Linux 和 Microsoft Windows, 2007 年被纳入 `TeX Live 2007` 和 `MikTeX 2.7` 发行版。

²⁸1999 年数学博士, 大阪市立大学 (Osaka City University) 博士后。2001 年韩国高等研究院 (Korea Institute for Advanced Study) 研究员, 2004 年水原大学 (University of Suwon) 讲师。

²⁹捷克马萨里克大学 (Masaryk University) 电脑教授。

³⁰16 岁起在英国邮局总局 (General Post Office, GPO) 当技工, 业余就读于几家技校。1970 年跳槽到莫林斯烟草机械 (Molins Tobacco Machinery), 1972 年跳到伦敦大学, 先后任程序员、网管、研究员等职。曾任英国 TeX 用户组主席。

³¹1991 年进入马萨里克大学信息系, 1996 年硕士, 2001 年博士。毕业后任教于胡志明市师范大学 (Ho Chi Minh City University of Pedagogy)。2006 年移居德国, 现任职于河谷科技 (River Valley Technologies)。包老师猜他的名字应该翻译成韩世城。

³²毕业于剑桥大学。1985 年加入一家从事语言教育的非盈利组织 SIL。



X_YT_EX 工作时有两个步骤, 第一步输出中间文件, Extended DVI (xdv), 第二步用驱动把 xdv 转为 PDF。缺省方式是两步一起执行, 直接输出 PDF, xdv 只在内存中露过一小脸儿。用户也可以要求只执行第一步, 保存 xdv。X_YT_EX 现有两个驱动, 一个是通用的 xdvipdfmx, 看名字就知道它是 dvipdfmx 的亲戚; 另一个是 Mac OS X 上的 xdv2pdf。

1990 年代初 Hans Hagen³³ 等开发的 ConT_EXt 是在格式方面的一员革命小将。它一方面支持先进的语言、字体和排版技术, 另一方面可以任意选择 T_EX, pdfT_EX, LuaT_EX 等引擎。除了功能上的差异, 它们在以下几方面也有所不同,

- 架构, L^AT_EX 内核小, 很多功能通过宏包来实现; ConT_EXt 内核大, 事无巨细都亲力亲为。
- 宏包, L^AT_EX 谁都可以参与设计宏包, ConT_EXt 则比较封闭。L^AT_EX 的宏包多了, 它们之间会有冲突; ConT_EXt 没这个问题。
- 运行, L^AT_EX 占用内存小, 速度快; ConT_EXt 占用内存大, 速度慢。

1.1.4 小结

包老师语重心长地总结道, 数字排版有四个重要环节: 标记语言、页面描述语言、光栅图像处理器、输出设备。T_EX 是最精确、最高级的面向专业排版的标记语言。T_EX 家族可以划分为四个层次: 引擎、格式、宏包、驱动。包老师通常选择 X_YT_EX 引擎和 L^AT_EX 格式。

1.2 优点缺点

通过上节内容我们已经知道, T_EX 相对于其他标记语言有较大优势, 但是在桌面印刷领域还有一种不可忽视的类别, 所见即所得 (WYSIWYG) 系统, 比如微软的 Word。其实 Word 也有自己的域代码 (field code), 只是一般用户不太了解。

一般而言, T_EX 相对于所见即所得系统有如下优点:

- 高质量, 它制作的版面看起来更专业, 数学公式尤其赏心悦目。
- 结构化, 它的文档结构清晰。
- 批处理, 它的源文件是文本文件, 便于批处理, 虽然解释 (parse) 源文件可能很费劲。
- 跨平台, 它几乎可以运行于所有电脑硬件和操作系统平台。
- 免费, 多数 T_EX 软件都是免费的, 虽然也有一些商业软件。

³³1986 年创办普瑞格玛 (Pragma)。NTS, pdfT_EX, LuaT_EX 等项目的参与者。



相应地, \TeX 由于其工作流程, 设计原则, 资源的缺乏, 以及历史局限性等原因也存在一些缺陷:

- 语法不如 HTML 和 XML 严谨、清晰。
- 制作过程繁琐, 有时需要反复编译, 不能直接或实时看到结果。
- 宏包鱼龙混杂, 水准参差不齐, 风格不够统一。
- 排版样式比较统一, 但因而缺乏灵活性。
- 相对于商业软件, 用户支持不够好, 文档不完善。

2000 年有记者在采访 Lamport 时问: “为什么当前没有高质量的所见即所得排版系统?” 他回答道: “门槛太高了, 一个所见即所得系统要做到 \TeX 当前的水平, 工作量之大不是单枪匹马所能完成³⁴。微软那样的大公司可以做, 但是市场太小了。我偶尔也会想加入 ‘Dark Side’, 让微软给我一组人马来开发一个这样的系统。”³⁵

窃以为这两大阵营其实是萝卜青菜的关系, 与其抱残守缺、互相攻讦, 不如各取所需; 甚至可以捐弃前嫌、取长补短, 共建和谐社会。

1.3 软件准备

初学者面对上述那些引擎、格式、宏包、驱动等概念可能手足无措, 所幸是有好事者把这些东东连同一些实用程序 (utilities), 遵照 \TeX 的规范打包集成在一起, 形成一个发行版 (distribution) 或者说实现 (implementation)。

与此类似的例子有 Java 和 Linux, 比如 Sun, IBM, BEA³⁶ 等公司都有自己的 Java 虚拟机, 它们都被称作 Java 的实现; 而 Linux 有 Red Hat/Fedora, Ubuntu, SUSE 等发行版。

发行版的基本作用是提供 \TeX 后台处理机制和命令程序, 用户还需要前台编辑器来编辑源文件。有的发行版也会附带一个编辑器, 但是未必符合用户的口味。常用的 \TeX 发行版和编辑器见 ??。在使用 \TeX 的过程中可能还需要其它一些软件, 包老师会在相关章节中分别介绍。

成熟稳重历经沧桑的用户更喜欢通用的编辑器, 比如 Emacs 和 Vim, 包老师喜欢轻量级的 PSPad。专用编辑器中, TeXstudio 和 TeXnicCenter 功能较强。Eclipse

³⁴ \TeX 也不是那几个大腕儿完成的, 他们背后还有众多默默无闻的小人物, 比如当年 Knuth 手下的大批学生。正所谓一将功成万骨枯。

³⁵他果然于次年加入微软。

³⁶1995 年 Sun 的几位员工另立门户成立了 BEA, 2008 年被 Oracle 收购, 2009 年 Oracle 又收购了 Sun。包子曰: 天下大势, 分久必合。



表 1.2: \TeX 发行版与编辑器

操作系统	发行版	编辑器
通用	TeX Live	TeXworks
Windows	MikTeX	TeXstudio
Mac OS	MacTeX	TeXShop

有一个插件 `TeXlipse`, 工作流程管理做得较好; 但是 `Eclipse` 的自动换行有毛病, 写程序还行, 写文章就不合适了。

1.4 学习方法

限于篇幅和水平, 本文只能介绍一些皮毛。最流行的英文入门资料是 Tobias Oetiker (1969–)³⁷ 的 *lshort*[?], 若想全面深入地了解 \TeX , 可以拜读 Mittelbach 的 *LaTeX Companion*[?]

[Comprehensive TeX Archive Network](#) (CTAN) 和 TUG 都提供了丰富的资源。常用宏包的简介见 [TeX Catalogue](#)。常见问题可以参考英国 TeX 用户组 (UK TUG) 的 [TeX Frequently Asked Questions](#)。

中文资料可参考李果正的《大家来学 \TeX 》[?], *lshort* 有吴凌云 (1975–)³⁸ 等的中文译本³⁹。中文 \TeX 论坛有 [水木清华 BBS TeX 版](#)、[CTeX 论坛](#)。中文问题可参考 [CTeX FAQ](#)[?]

在学习过程中, 我们要先宏观后微观: 先对体系、结构、框架等有所了解, 慢慢再掌握使用方法, 细节处理更要靠经验的积累。同时还要勤于思考 *what, why, how* 等问题。

在科学上没有平坦的大道, 只有那些不畏劳苦沿着陡峭山路攀登的人, 才有希望达到光辉的顶点。

— 卡尔·马克思

无他, 唯手熟尔。

— 卖油翁

³⁷ 奥腾大学 (Kantonsschule Olten) 学士, 苏黎世联邦理工学院 (Swiss Federal Institute of Technology Zurich) 电子硕士, 1995 年留校工作。

³⁸ 1997 年武汉大学应用数学学士, 2002 年中科院运筹学博士, 香港科技大学、中科院、康奈尔博士后, 2007 年中科院数学与系统科学研究院副研究员。

³⁹ 就在英文版的隔壁, 更新得慢一点。



用心。

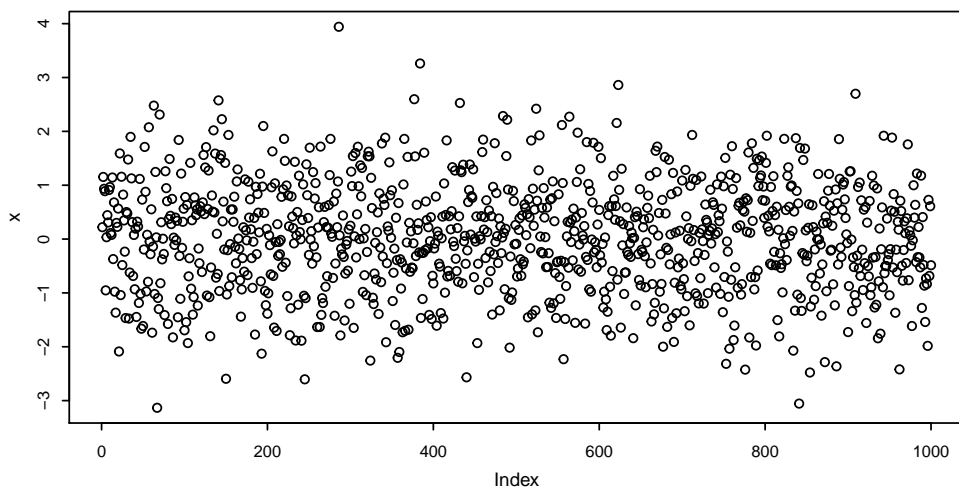
— 斯蒂芬·周



第二章 R



```
x <- rnorm(1000)
plot(x)
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor

This is a paragraph in the right margin

lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in,

This is a paragraph in the left margin





judge 均衡的件很件很将很站跟涵盖跟汗的 judge 均衡的件很件很将很站跟涵盖
跟汗的 judge 均衡的件很件很将很站跟涵盖跟汗的 judge 均衡的件很件很将很站
跟涵盖跟汗的 judge 均衡的件很件很将很站跟涵盖跟汗的 judge 均衡的件很件很
将很站跟涵盖跟汗的 judge 均衡的件很件很将很站跟涵盖跟汗的 judge 均衡的件
很件很将很站跟涵盖跟汗的 judge 均衡的件很件很将很站跟涵盖跟汗的 judge 均
衡你好 你好 你好 dsdasgasg as gadsfgafdgfdagfdash asth dga dg ads fghdighidhg 的件
很件很将很站跟涵盖跟汗的 judge 均衡的件很件很将很站跟涵盖跟汗的 judge 均
衡的件很件很将很站跟涵盖跟汗的 judge 均衡的件很件很将很站跟涵盖跟汗的
judge 均衡的件很件很将很站跟涵盖跟汗的 judge 均衡的件很件很将很站跟涵盖
跟汗的 judge 均衡的件很件很将很²

W.F.: hti 站跟涵盖跟汗的 judge
均衡的件很件很将很

² this is sidenog
fg f gfd gf hgf hg
hfte

你好 你好 你好 dsdasgasg as gadsfgafdgfdagfdash asth dga dg ads fghdighidhg

W.F.: hti 站跟涵盖跟汗的 judge
均衡的件很件很将很

你好 你好 你好 dsdasgasg as gadsfgafdgfdagfdash asth dga dg ads fghdighidhg

W.F.: hti 站跟涵盖跟汗的 judge
均衡的件很件很将很

