

deepin 集结

42 **【特别策划】**
>>>

深度科技 国产操作系统 打造者

04期
2016年12月

58 **【深度·讲坛】**
>>>

一些冷门的 C 语言知识
debian 系发行版 initramfs 调试方法
玩转深度服务器安装部署

【深度人·在说】

P22 夏彬：
深度于我
不仅是缘分

46 **【深度·社区】**
>>>

Deepin 的盈利模式
是什么？

50 **【深度·案例】**
>>>

紧急出动——Oracle11g
Rac 的挑战
龙芯优化的战斗小结

80 **【深度·伙伴】**
>>>

与军队客户的三问三答
10年回顾 我的 Linux 之路
为开源和商业正名



深度操作系统公众号



深度科技 总工程师
张磊

投入操作系统这个行当之前，我们显然没有对未来的困难有足够的认识，睁着眼睛，兴冲冲地就入了大坑，在里面奔流逐浪，起起伏伏，不知不觉都这么长时间了，真可谓是一入江湖少年老，忽而将军成白头。硬件兼容性、软件兼容性、功能、稳定、性能、安全、易用性、可用性……趟过了一个又一个的坑，好不容易喘了一口气，发现前面还有高山峻岭横着呢，不得不继续加快脚步，勇往直前。

开发基础软件是一件又苦又累的事情，但同时也乐在其中。当看到客户顺利部署并流畅使用我们系统时，我们会高兴；当亲手使用基于我们产品制作的设备时，我们会高兴；当收到用户来信点赞的时候，我们会高兴；当发现ebay和youtube上介绍我们产品，推荐我们产品的内容的时候，我们会高兴；当我们经过漫漫长夜的加班，搞定了高难度的特性或者bug，通过了unit tests和code review，在旭日东升之时做了最后一个git push时，我们也会高兴，甚至兴奋。这中间当然也会有批评，还会有挫折，但是所有人都知道，我们正在坚定地一步一个脚印地前行。

十几年前，我们的产品只有几百人使用，现在不仅在社区里，而且在政府、军队和行业里有了广泛的影响，在发行版里排行前十；从原来只能汉化和打包，发展到能改写整个桌面，能对系统各方面进行加固、调优；从只支持x86，到支持多个架构的处理器；从只有中文版，到拥有数十种网友志愿翻译的语言版本，镜像遍布全球。我们已经创造了历史，并将创造更好的未来。

在我想来，所谓的成功，不在于某个时刻瞬间的光彩耀目，而更在于持续的专注与努力，在于人与人之间从心底而来的投契，在于亲手创造的产品得到了首先是我们自己，然后是用户和客户的衷心认可。长夜漫漫，但是经过漫漫长夜，迎来的将是光明的明天。

deepin 集结

策划 Hosted by
武汉深之度科技有限公司 Wuhan Deepin Technology Co., Ltd.
编辑 Edited by
《deepin集结》杂志编辑部 Editorial Office of DEEPIN JIJIE

总编辑 Editor-in-chief
刘闻欢 Liu Wenhuan
副总编辑 Deputy Editor
许可 Xu Ke
执行编辑 Executive Editor
郝俊 Hao Jun
编辑 Editor
许峰 张文豪 Xu Feng Zhang Wenhao
脱文剑 郑丹 Tuo Wenjian Zheng Dan
采编 Assistant Editor
张凤玲 Zhang Fengling
李会会 蒋文 Li Huihui Jiang Wen
美术设计 Art Editor
秦娣 Qin Di

网站 Website
<http://www.deepin.com>

邮箱投稿 Contribution
deepin-magazine@deepin.com

市场推广 Marketing
account-marketing@deepin.com

武汉联络处 Wuhan Office
地址 Address
武汉市光谷大道77号 The 6th Floor, Building B18, FinancialHarbour,
光谷金融港B18栋6楼 No. 77, Optics Valley Avenue, Wuhan, China
邮编 430223
电话 +86-27-87805607

北京联络处 Beijing Office
地址 Address
北京市海淀区知春路 Room 501, Tower B, Jinqiu International Building,
锦秋国际大厦B座501室 Zhichun Road, Haidian District, Beijing, China
邮编 100088
电话 +86-10-62669499

上海联络处 Shanghai Office
地址 Address
上海市长宁区 Room 15A01, No. 1258, Yuyuan Road,
愚园路1258号15A01室 Changning District, Shanghai, China
邮编 200050
电话 +86-21-60726030

准印证号 (鄂) 4300107
承印单位 武汉金港彩印有限公司
出版日期 2016年12月

02 深度·春秋

- 02 国产操作系统典范：deepin 操作系统
- 05 深度看图 V1.1 发布
- 06 深度科技荣获第三届“楚天杯”工业设计大赛 一等奖
- 08 深度终端 V2.1 正式发布——终端之上，命令之下
- 11 2016 第六届深度开发者与用户大会完美收官
- 18 用光谷智慧捍卫国家信息安全
- 20 捍卫网络信息安全 国产操作系统在行动

22 深度人·在说

- 22 夏彬：深度于我不仅是缘分

28 行业·观察

- 28 第三届世界互联网大会发布《2016 年世界互联网发展乌镇报告》（全文）
- 33 为何境外 IT 巨头频频来华成立合资公司
- 38 面对境外 IT 巨头的合资潮 可以采取三项措施应对
- 40 制定统一标准 做大国产操作系统

46 深度·社区

- 49 Deepin 的盈利模式是什么？

50 深度·案例

- 54 紧急出动——Oracle11g Rac 的挑战
- 56 龙芯优化的战斗小结

58 深度·讲坛

- 58 一些冷门的 C 语言知识
- 66 debian 系发行版 initramfs 调试方法
- 70 玩转深度服务器安装部署

80 深度·伙伴

- 80 与军队客户的三问三答
- 82 10 年回顾 我的 Linux 之路
- 84 为开源和商业正名

42 深度·策划

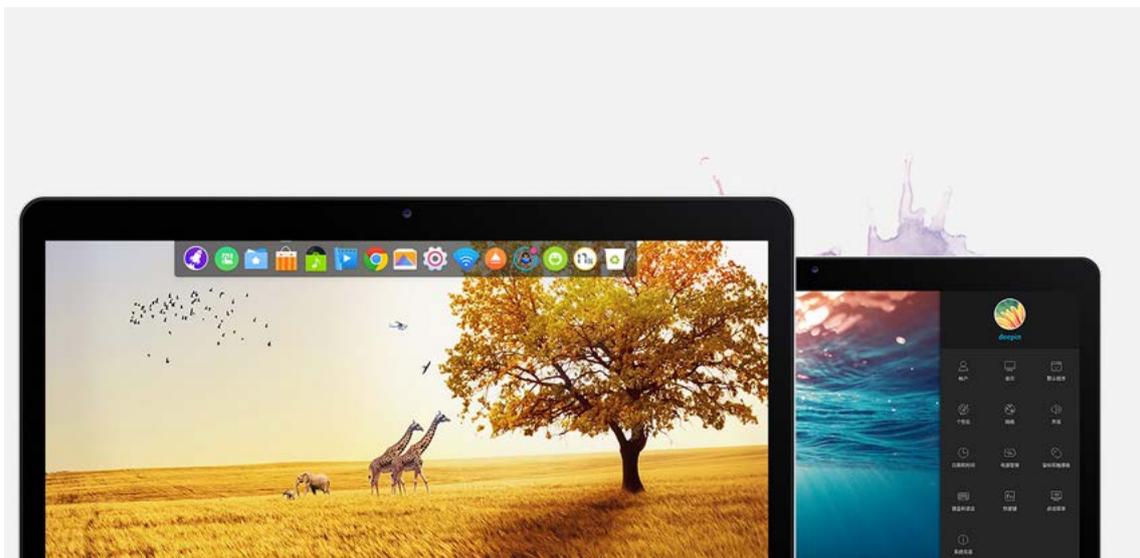


深度科技 国产操作系统 打造者

深度科技致力于为用户提供友好、美观、易用的操作系统，为了改善 Linux 平台上应用软件生态，除自己研发了深度影院、深度截图等应用，还建立深度软件商店为用户提供更多应用软件，努力打造 Linux 软件生态。深度科技作为国产操作系统生态打造者，在完善自有软件的同时，近年来联合国内外软硬件企业，开发了多款符合中国用户需求的应用软件。

74 深度·生活

- 74 “深度”的泰国之旅



国产操作系统典范：deepin 操作系统

● 摘自今日头条 一生为你 / 文

在中国，Windows 各个版本的市场占有率超过了 97%，macOS 的市场占有率是 0.72%，而各类 Linux 操作系统的市场占有率不超过 1%，其中国产操作系统更是微乎其微。

作为信息安全极为重要的一环，自主操作系统是国人的梦，也是国人的痛。

“Windows 为美国政府服务，开后门窃取其他国家信息”、“XP 全面停止升级，强制捆绑微软服务”……一方面，从政府到百姓都在呼吁真正的国产操作系统，另一方面却是惨淡的市场现状，曾经影响很大的中科红旗的红旗操作系统在 2014 年破产

拍卖，基本在市场上消失了。

deepin 是中国唯一一家不靠政府补贴或经费支持活下来的 Linux 厂商。

深度操作系统是一个致力于为全球用户提供美观易用、安全可靠体验的 Linux 发行版。它不仅对最优秀的开源产品进行集成和配置，还基于 Qt 技术开发了深度桌面环境和深度控制中心，并且开发了一系列面向日常使用的深度特色应用，如深度商店、深度截图、深度音乐、深度影院等。

深度操作系统非常注重易用的体验和美观的设



计，对于大多数用户来说，它易于安装和使用，能够很好的代替 Windows 系统进行工作与娱乐。近年来，深度操作系统发展迅速，获得全球四十多个国家用户的支持，目前全球下载量达 5 千万次，用户达几十万人，提供 30 种不同的语言版本，国外用户

在六成以上，已是全国乃至全球知名的系统之一，并成为在 Distrowatch 上排名最高的中国 Linux 操作系统发行版。

大部分硬件有了驱动支持，但是专用硬件和较

办公软件

类别	Windows软件	深度原生支持
文档处理	Office*、Acrobat、福昕	WPS、福昕、Libre Office、Calligra
浏览下载	IE 360浏览器 迅雷 QQ旋风	Chrome、火狐、uTorrent、axel
压缩软件	WinRAR*、WinZip、7-zip	7-zip、gzip、unrar
中文输入	搜狗输入、Rime、QQ输入	搜狗输入、Rime
即时通讯	QQ*、旺旺*、Skype、飞秋、RTX*	Skype、Iptux
图像处理	Photoshop*、美图秀秀*	Gimp、深度截图
邮件收发	Outlook、Foxmail	雷鸟、Evolution
安全杀毒	360卫士、腾讯管家	NOD32、Clamav、ufw、rkhunter



基础硬件

类别	主流产品	支持状态
CPU	英特尔、AMD、龙芯	支持良好
显卡	英特尔、英伟达、AMD	除了部分N卡外支持良好
声卡	创新、英特尔、Realtek	支持良好
网卡	英特尔、Realtek、Marvell	支持良好
打印机、扫描仪	惠普、佳能、三星、爱普生	基本支持良好
U盾	产品较多	需要开发新的软件
专用设备	读卡器、指纹识别仪等	需要开发新的软件
服务器设备	HBA、HCA	支持良好

生活娱乐

类别	主流产品
音频视频	深度影院、深度音乐
磁盘光盘	Clone Zilla、Gparted、Brasero
网上购物	淘宝、支付宝安全控件可以直接支持
银行金融	工行网银、招行网银专业版、建行网银、浦发银行、大智慧、中信证券至信版、华宝证券
大众游戏	Steam、魔兽争霸、CS等桌面游戏以及大部分页游



老的硬件没法迁移，必须重新开发。

除了桌面操作系统，深度操作系统服务器版率先实现了对 x86 和国产 CPU 平台的统一框架支持。通过对全生态环境的支撑，及多应用场景解决方案的构建，能够满足企业级用户对服务器高可用性、高稳定性、高可靠性的要求。

笔者有幸听过深度项目组王勇的报告，觉得他以及他的团队是一群敢想敢做，有着爱国梦的人才。

deepin 系统推广的主要阻力：旧有工具的使用习惯、新环境使用理念、对国产东西的不信任。主要优势：自主可控、政策支持、爱国情怀。

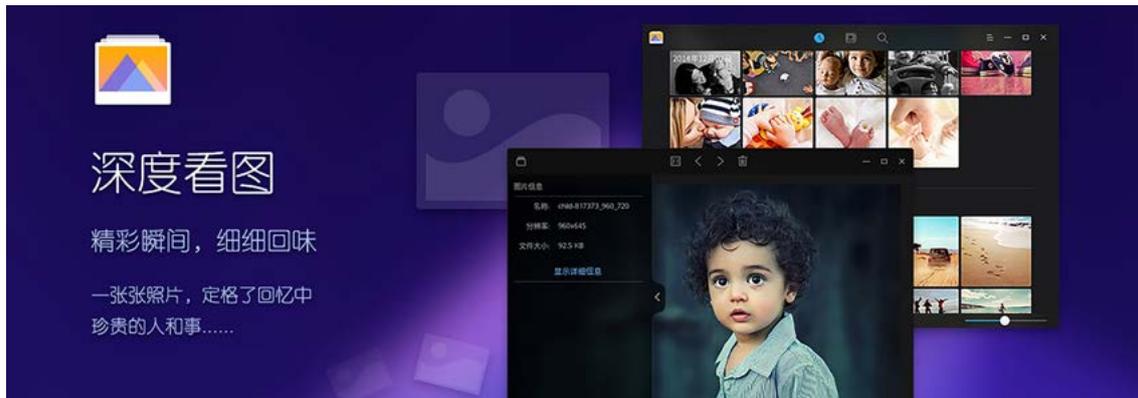
好消息是 deepin V15 版本对小米笔记本高配版做了适配，可完美支持小米高配版笔记本。笔者认为，这是积极的信号，看好 deepin 操作系统，和小米或者华为合作吧，因为他们都拥有自己的 CPU 技术了，



虽然现阶段只是手机 CPU，但我想很快会有 PC cpu 的，国产系统配上中国芯，就完美了吧。

假如：小米和深之度合作，共同打造 MiOS 闭环生态系统，用小米的用户群、社会影响力、硬件生产链，我想会是双赢的结局。小米不再只做 UI 了，合作开发能无缝连接 PC 和移动端的系统。软硬同时出击，说不定有抗衡微软、苹果的实力！

好的东西是会发光发亮的，相信美好的事情即将发生。。。d



深度看图 V1.1 发布

● 深度科技 产品部 / 文

深度看图 V1.1 为修正版本，不仅对界面细节进行了调整，整体功能和性能得到了优化，同时修复了用户反馈的 Bug。

界面细节调整优化：

时间线和相册

- 主面板和工具栏的背景颜色调整
- 相册的封面图标和绘制细节调整，相册相框改为深色，突出中间的图片
- 时间线界面的布局，将时间移到了分割线上方，调整了不同时间段的间距
- 相册界面的布局，调整相册之间的间距
- 工具栏图标的选中样式调整，顶部工具栏和底部状态栏增加层次感
- 工具栏下面的蒙层，加入黑色渐变，优化浅色背景下时间显示
- 初始界面的空白图案和导入按钮替换，并调整文字与图案图标间距

查看图片

- 增加左右按钮和导航窗口的阴影效果
- 调整工具栏图标间的位置防止误触
- 图片详细信息面板的布局调整

- 幻灯片的播放效果调整

性能优化和功能调整：

- 优化窗口的绘制和切换性能
- 扩展了更多的格式支持，目前支持的格式如下：BMP, ICO, JPG, JPE, JPEG, JNG, PCD, PCX, PNG, TGA, TIF, TIFF, PSD, XPM, DDS, GIF, SGI, J2K, JP2, PCT, RAW, WEBP, WDP, CR2, PEF, ARW, NEF, DNG, RAF, ORF, SVG, MEF, MRW
- 多选菜单功能的调整
- 在部分删除操作中增加删除对话框
- 修改了缩略图的存储方式

Bug 修复：

- 修复全屏时窗口卡顿的问题
- 修复图片缩放滚动值计算不够准确的问题
- 修复导航窗口的定位偏移的问题
- 修复部分图片无法正常导入的问题
- 修复工具栏出现残影问题
- 修复部分图片无法正常设置壁纸的问题
- 修复部分图片无法正常解析的问题

Ps：大家可以通过更新系统以获取深度看图 V1.1，或者直接在深度商店搜索下载。[d](#)



深度科技荣获 第三届“楚天杯”工业设计大赛一等奖

● 深度科技 市场部 / 文

湖北·武汉举办的以“创新·融合·引领·发展”为主题的第三届“楚天杯”工业设计大赛日前揭晓所有获奖作品，武汉深之度科技有限公司参加本届“楚天杯”工业设计大赛的“软件与交互产品设计”项，并获得该项设计的一等奖。

工业设计是生产性服务业的重要组成部分，是创建自主品牌、提升产品附加值、增强产业竞争力

的重要措施和途径。本次软件与交互产品设计参赛作品需突出“互联网+”、智慧城市等新型理念的商业、办公、家居、医疗、教育、通讯等设计作品。深度桌面作为深度科技团队自行研发的软件与交互产品设计，无论是在用户体验还是美观程度上都极大的展现了深度操作系统的友好、易用。

深度桌面完全符合“楚天杯”的五项评审标准：



深度桌面



深度控制中心



深度分类应用窗口



深度无边框 UI 设计



深度图标设计

创新性、技术性、审美性、环保性、表现力。

深度科技副总经理王勇在进行终极答辩时，首先向各位评审介绍了深度科技的发展历程，深度科技是湖北省内唯一的操作系统研发企业，经过几年的发展，已经广泛用于党政军以及 ATM 行业等金融行业。然后依次向各位评委展示了深度桌面极简式设计风格：即看即点，降低用户心理负担；深度操作系统桌面图标设计：一切为了识别度；沉浸式操作体验：无边框设计和响应式设计。

最后阐述了深度操作系统桌面设计原则：

1、内容至上：

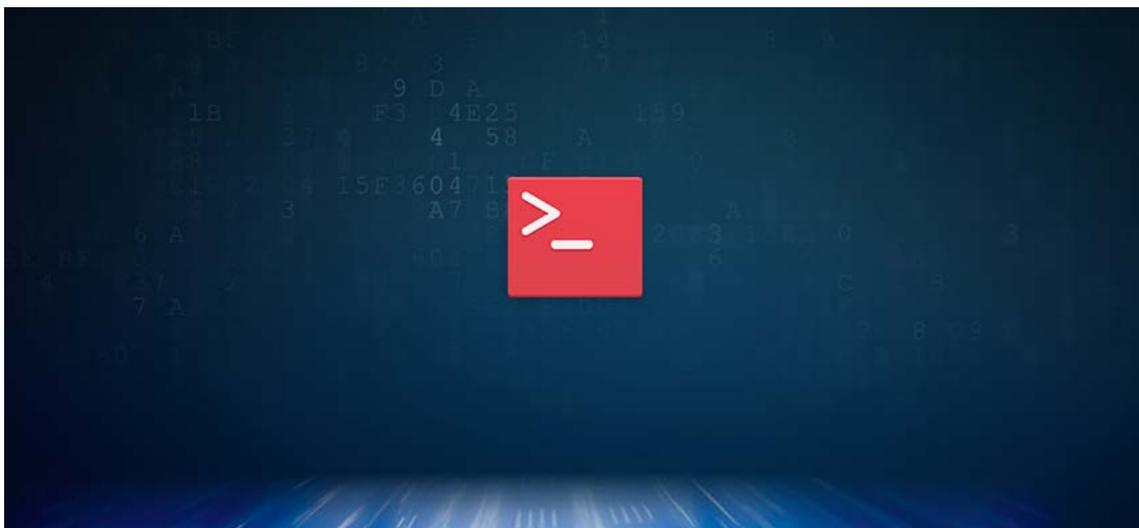
设计应该是内容展示的载体，而不是喧宾夺主；

2、尊重用户习惯：

好的设计应该是基于用户多年的习惯而直觉化操作；

3、响应式设计：

用户只用关注当前的操作，而不是堆砌功能增加用户额外心理负担。d



深度终端 V2.1 正式发布 ——终端之上，命令之下

● 深度科技 产品部 / 文

作为 Linux 业界的造轮子团队——深度科技，我们一直在思考什么才是终端用户最需要的呢？总结来说，就是速度快、资源占用少、极致的终端体验，然后就是高颜值了。

经过深度科技团队成员几个月的研发，深度终端 2.1 版本终于面世了。秒速启动和关闭，拥有行云流水般的流畅体验；轻松创建多工作区并以标签页的形式展示，以及分割工作区成多个窗口，多线程高效工作体验；加强版远程管理支持，上传下载文件再也不是问题；更多多套主题随心换，终端也可以美美哒。

更多强大功能，静待您的发掘！

毫秒级启动，享受速度感

响应速度在 200 毫秒之内，真正做到毫秒级启动，同时优化内存占用率，系统资源消耗更小。



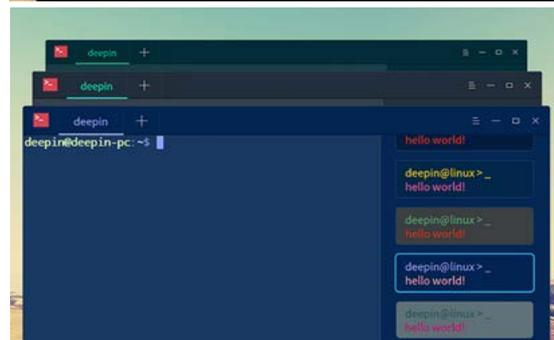
标签化设计，沉浸式体验

深度终端顶部采用类似 Chrome 的标签式设计，整体超窄边框显示，只为给你带来沉浸式的终端体验。



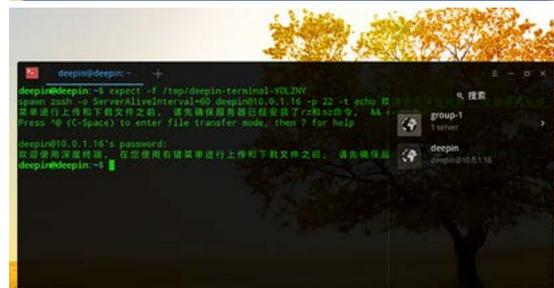
集多样主题，百变随心

默认内置 40 余款主题，总有一款捕获你的心，实现一秒快速切换，适合善变的你。



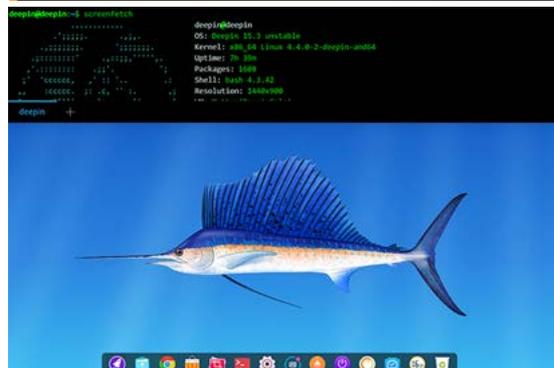
远程化管理，告别命令行

快速远程连接访问，让分组、自动执行命令更直观，上传下载文件轻而易举，从此告别命令行，纵有成千上万的服务器也可快速管理。



灵感于雷神，复古型字体

雷神模式取灵感于著名的 3D 游戏《雷神之锤》的命令行窗口，快速从上弹出终端窗口，节省空间又不会遮挡内容，只要系统安装了点阵字体，即可体验复古的终端信息显示。





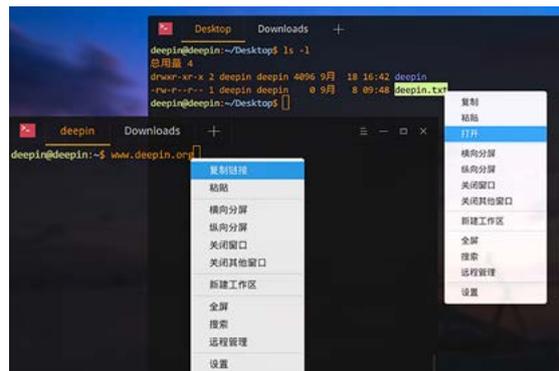
智能化感知，通知更个性

我们总是徘徊在多个机器和命令之中，当后台工作区任务完成后，会通过标签的颜色变化第一时间通知，实现智能化感知。



响应式设计，贴心式触发

右侧面板的激活和隐藏采用响应式设计，点击选项即出现，空白即消失；自动识别光标下的链接地址以及选中文件 / 文件夹关联打开。



简单获取，轻松安装

全新安装的深度操作系统 15.3 的用户，系统会默认预装深度终端 v2.1。

升级用户可以在控制中心的系统消息推送模块进行升级。

崇尚开源理念，分享开源快乐

“从开源中来，到开源中去”是深度操作系统

始终秉承的原则，我们一直崇尚开源的理念和推动 Linux 社区生态的建设。

分享的精神在我们心中从未改变，在此，我们号召其他社区的开发者体验并参与打包，让其更好的传播出去。d

深度终端 2.1 github 地址：

<https://github.com/manateelazycat/deepin-terminal>



2016 第六届深度开发者与用户大会 完美收官

● 深度科技 市场部 / 文

2016 第六届深度开发者与用户大会 (Deepin Developer & User Conference, 简称 DDUC) 在 2016 年 9 月拉开帷幕。本次活动是深度科技举办的第六届深度开发者与用户大会, 2016 年延续 2015 年第五届深度开发者与用户大会的主题“给所有人的深度”, 旨在向所有需要的人提供交互体验完美的深度操作系统。

第六届 DDUC 以全国分站的形式进行, 在西安、北京、上海、武汉、广州五个城市与开源开发者、爱好者展开交流, 与当地院校以及 Linux 兴趣小组联合举办, 共同宣传、阐述开源文化。

从建立社区团队到组建公司, 从非盈利到走商业化道路, 深度科技磕磕绊绊已走过 12 个岁月, 我们一直致力于开源操作系统的开发, 将美观、易用、安全、可控的深度操作系统给所有需要的人使用是深度科技坚持不懈努力的方向。

本次活动不仅是年度社区爱好者的聚会, 同时也是集结更多开源小伙伴的基地。我们坚信, 会有更多的学生、开发者、爱好者加入到这场盛宴, 共同探讨分享开源之乐。



01st 西安

2016.9.17

开源不止是眼前的编译
还有希望和生活
只有当你更热爱生活时
你才能创造更多的开源贡献

——王勇同学语录

深度科技联合西安邮电大学 Linux 兴趣小组举办的“软件自由日”活动暨 2016 第六届深度开发者与用户大会，在 9 月 17 日圆满举行。深度科技副总经理王勇在本次活动与现场同学分享了《开源不仅仅是热爱》的主题演讲，介绍了自己对国内开源社区的看法，通过自己的经验实实在在的告诉同学们应该如何参与开源，他认为开源社区的本质应该首先是有独立的思想，然后创造贡献，最后才是分享。

本次活动中进行分享的还有国防科技大学马俊博士、Fedora Project 中国区大使何宗元先生、IBM 王森以及 Linux 课程老师陈莉君教授。



02nd 北京

2016.9.21

北京交通大学站

2016 第六届深度开发者与用户大会（DDUC）北京站的第一场，于9月21日在北京交通大学完美收官。活动于当日下午 18:30 分正式开始，现场共有 250 余人参加，我们邀请到在开源领域影响力颇深的 Red Hat 与 Intel 公司技术专家共同分享开源技术与经验。

首先是 Red Hat 公司申伟与现场的同学分享了《如何优雅的使用 python》，讲述了 python 作为自由软件在开源软件开发中的特色。

Intel 公司王智给大家带来的演讲是《基于 INTEL GPU 的图形虚拟化技术》，介绍了虚拟化技术给嵌入式开发人员带来的灵活性和稳定性，并与现场同学活跃互动。

最后，深度科技潘海涛在本次活动与现场同学分享了《你也是一个深度贡献者》的主题演讲，向现场同学介绍了开源与 Linux 发行版以及深度操作系统的知识，呼吁更多的在校大学生加入开源、加入深度软件的开发。





02nd 北京

2016.9.25

中国科学院大学站

继西安邮电大学“软件自由日”活动结束后，9月25日深度科技联合中国科学院大学举办的“软件自由日”中国科学院大学雁西湖校区活动（2016 第六届深度开发者与用户大会中国科学院大学站）也顺利举行。

本次活动于下午 14:00 正式开始，现场共计 150 余人参加。深度科技潘海涛在本次活动中分享了《你也是个深度贡献者》的主题演讲，向现场同学介绍了开源与 Linux 发行版以及深度操作系统的知识，呼吁更多的在校大学生加入开源、加入深度软件的开发，引发现场同学极大的兴趣。

除了演讲分享外，还在现场实地演示装机，向同学们展示了深度操作系统的美观、易用。在本次活动中进行分享的还有 Red hat 工程师的主题演讲《github 的介绍及使用》，一铭软件项目介绍以及中国科学院大学的黄涌老师的分享。

DDUC 2016 将为深度的开源事业开启新的章程，我们不再局限于在线上社区中与开源爱好者互动，更是亲临学校，与大学生在一起，共同探讨、分享、交流开源技术、文化、思想。d



03rd 武汉

2016.10.12

金秋十月，深度科技在主场武汉大学开启了 2016 DDUC 武汉站活动，作为主场活动，是本届 DDUC 的第三站，也是回归武汉之行。活动于 10 月 12 日下午 18:30 分正式开始，我们邀请到在开源领域影响力颇深的 Redhat 公司工程师王亚宁和《黑客道诠释》译者洪峰共同分享开源技术与经验。

活动开始，首先由中国开源软件推进联盟常务副秘书长宁固进行致辞，他从当下中国开源现状出发，鼓励并倡导中国开源文化在高校中的推广。

深度科技副总经理王勇在本次活动与现场同学分享了《开源不仅仅是热爱》的主题演讲，分享了自己对国内开源社区的看法，通过自己的经验告诉同学们应该如何参与开源，他认为开源社区的本质应该首先是有独立思想，然后创造贡献，最后才是分享。

《黑客道诠释》译者洪峰与现场同学分享了《黑客和创客的四种思维》。

Redhat 公司工程师王亚宁与现场同学分享了《What Is Red Hat Virtualization Host And Related Technologies》。

最后的分享者是深度科技技术工程师王耀华，作为深度操作系统开发项目组组长，他给大家带来的演讲是《一次 deepin 系统的发布之旅》，讲述了系统开发、测试、发布等过程。





04th 上海

2016.10.16

2016 DDUC 上海站活动也于 10 月 16 日圆满结束。在“魔都”这个时尚的都市，深度科技联合上海 SHLUG 在华东师范大学开启了深度开发者与用户大会在上海的第一次开源盛会，我们邀请到 Intel 软件工程师张锐以及上海 Linux 用户组负责人 Thomas Yao 给现场参会同学带来精彩演讲。

活动开始，主持人兼演讲嘉宾深度科技副总经理王勇分享了《开源不仅仅是热爱》演讲，分享了自己对国内开源社区的看法，深度科技近几年在开源领域的开发成果以及深度操作系统下一步开发计划：解决 Linux 下兼容所有类型打印机驱动，新版的深度音乐、深度影院、深度截图以及高分屏支持等，深度科技研发团队将致力于给深度操作系统用户带来无限可能。

Intel 软件工程师张锐分享了《Linux suspend/resume optimization》的演讲，讲述了 Linux 内核的优化相关知识。

上海 Linux 用户组负责人 Thomas Yao 分享了《我的开源社区之路》演讲，讲述了从接触计算机到接触开源以及自身在开源领域的创业故事，并与现场同学活跃互动。 



05th 广州

2016.10.29

2016 第六届深度开发者与用户大会（DDUC）于 10 月 29 日在广州圆满收官。广州站作为本次活动的最后一站，邀请到开源大神在贝塔咖啡与各位开源爱好者及开发者进行了交流与分享。

活动于当天下午两点开始，首先是开源大神 Shawn C 给现场的朋友分享了《征途：HardenedLinux 社区 2016》的演讲；95 后小伙郑兴达现场分享《关于 arm 内核主线移植》；开源圈 Robin 大神做了关于《GObject, Introspection & Vala》的演讲；深度科技副总经理王勇带给大家的是《开源不仅仅是热爱》的演讲，分享了自己对国内开源社区的看法，深度科技近几年在开源领域的开发成果，同时还分享了接下来深度的一些开发计划以及新版本的透露。

至此，2016 第六届深度开发者与用户大会（DDUC）全国活动已全满结束，共走过 5 座城市举办 6 场活动，在 5 所高校中与当地开源爱好者交流、分享中国开源技术。我们期待明年的深度开发者与用户大会与更多人见面，共同建设、传播开源文化。





用光谷智慧捍卫国家信息安全

● 湖北日报 / 文

近日，北京市民周森在金融大街上的某国有银行 ATM 机上取现 1000 元。虽然外观与其他 ATM 机毫无二致，但这台机器的屏幕上显现着武汉“面孔”。它运行的是由武汉深之度科技有限公司（以下简称深度科技）自主研发的深度操作系统。

说起操作系统，大多数人首先想到的是微软的 Windows 和苹果的 Mac。深度操作系统，这套被誉为“汉产 Windows”的操作系统，具备怎样的“撒手锏”？

12 年深耕自主操作系统

深度操作系统的准确名称是 Linux 系统国内发行版，基于开源 Linux 内核开发。“起初只是出于兴趣。”深度科技创始人刘文欢介绍，2004 年，他与一群 Linux 爱好者组建团队，尝试打造一个 Linux 社区操作系统。随着其作品受到国内外用户广泛欢迎，他萌发成立公司的想法。2011 年，深度科技在光谷落户，这也是我省唯一一家研发操作系统的企业。

对于应用到政府和关键行业的基础软件平台，相比 Windows 系统，深度操作系统最大优势在于其代码安全可控。形象地说，Windows 系统是在黑屋子里做出的产品，可能留有危及信息安全的“后门”，且微软公司不允许他人检查。深度操作系统则是基于开源代码加上大量自研成果而形成的，研发过程透明，源代码对用户公开可供审查，同时还具备根据用户需求进行自主修改和发展的能力。

8月25日，一个黑客组织入侵了泰国近千台使用 WindowsXP 系统的 ATM 机。深度科技副总经理王勇介绍，全球有 9 成以上 ATM 机使用 WindowsXP 操作系统，由于 2014 年 4 月微软公司已停止维护该系统，这些 ATM 机都有可能成为黑客的攻击对象。“使用深度操作系统的 ATM 机，提高了对这种风险的防范力度。”

中国工程院院士倪光南表示，当下是国产操作系统发展最好的时代，寄望深度操作系统有朝一日能替代微软 Windows 系统。

深度科技创始人刘文欢说，近年来，深度科技持续良性发展。目前深度操作系统已通过了相关认证，在政府、金融、运营商、教育等客户中得到应用。截至今年，深度操作系统下载量超过 5000 万次，提供 30 种不同的语言版本，是唯一一款登上全球排名榜单的国产操作系统。

致力打造全局生态圈

“研发一个操作系统不容易，更困难的是整个

生态系统的打造。”刘文欢坦言，相比 Windows 系统和 Mac 系统，国产操作系统最大短板在于应用程序的兼容性和推广力度。

由于装机数量有限，软件商不愿投资研发基于 Linux 系统的应用程序。而应用程序不够多，个人用户和企业用户也缺乏使用国产操作系统的积极性。

为解决这个问题，深度科技主动出手，一方面上门找软件商，出钱出力协助研发；另一方面，与专业公司合作，通过“移植”的方式，让一些原本只能在 Windows 系统使用的软件也能在深度操作系统上畅快运行。目前，深度操作系统的软件商店中已经有 2000 多款常用软件可供下载使用，日常使用与 Windows 系统差距越来越小。

今年，深度科技参与武汉市信息中心电子政务内外网平台建设。金山软件提供 office 在线编辑系统、金格控件提供电子签章功能、力龙信息提供 OA 办公平台，最终通过深度操作系统整合呈现。目前，这一平台在试用中，用户反馈良好。

“国产化替代不是深度科技的单打独斗，它需要打通硬件制造、软件适配、操作系统研发多个环节。”王勇认为，该项目的建设过程诠释了国产自主安全可控操作系统生态圈的涵义。

近日，多名省领导到深度科技调研。“这给予我们莫大的鼓励。”刘文欢说，深度科技将一如既往深耕操作系统，用光谷智慧捍卫国家信息安全。d



捍卫网络信息安全 国产操作系统在行动

● 深度科技 市场部 / 文

在已经落下帷幕的 2016 年国家网络安全宣传周上，深度科技作为武汉市重点企业亮相网络安全博览会。深度科技的参会主题为“安全可控 营造生态”，旨在向外界展示深度科技作为操作系统研发企业，竭力为国产化及网络信息安全做出贡献。

深度科技近年来与国内外硬件厂商展开紧密的合作，深度操作系统也在政府、教育、金融等领域不断投入应用。



深度科技展台



自主可控

目前，深度操作系统已通过公安部安全操作系统认证、工信部国产操作系统适配认证、亦入围国管局中央集中采购名录，并支持各类国产 CPU 平台，符合中办安全可靠的要求。与 360、绿盟科技进行系统安全分析合作，定期面向全球的安全补丁升级体系。

营造生态

与各芯片、整机、中间件、数据库等厂商保持紧密合作；与国内众多知名企业联合开发出多款符合中国用户需求的应用软件。

国际影响

深度操作系统下载超过 5000 万次，提供 30 种不同的语言版本，以及遍布六大洲的 70 多个镜像

站点的升级服务。

“网络安全产业，需要有底线、有情怀的企业。坚持走自主可控的道路，政府应该扶持有技术的企业活得更好”，深度科技总裁刘文欢表示，随着武汉近年来城市迅速建设发展，创业生态日益良好，政府加大支持力度，产业聚集必将加快。

在武汉网络安全周期间，深度科技展台通过将国产操作系统搭载在不同 CPU 平台上，供参观者使用体验的方式，向宣传周期间的参观人员展示了国产基础软硬件产品的发展现状和良好的使用体验，并有专门的技术工程师从安全可控、易用美观角度给参观者进行分析讲解，得到现场观众的一致好评。d



夏彬： 深度于我 不仅是缘分

深度科技 开发团队 技术总监

【编者按】

在一件事情上坚持很久是件不容易的事情，尤其在生命中最宝贵的青春岁月中坚持着一件看似枯燥的事情，而我可爱的同事中有很多这样的人，不畏漫长过程，不畏前方阻碍，用满腔的热情为青春奋斗，为理想奋斗，为一些毋庸置疑、可坚持的事情奋斗。

文章中的主人公说“人如果喜欢一个事物时，就会把各种没有直接关系的现象扯在一起称之为‘缘分’”。在深度，很多人都是这样。也许，若干年后，你回过头去看，这段岁月会构成你回忆中最为动容的画面。

学生时代的计算机之缘

我和很多资质平庸的人一样，小时候成绩还不错，越大越差最后只能混完高考了。

记得学前班的第一天开始我就控制不住的要睡觉，毫不夸张地说，上课时间里有一半的时间在睡觉。

小学是各种“业余”课多，老师管的不严。

小升初时，勉强考上市里最好的中学。开始了远离家人，在校住读的生涯。也是从那时有了更多时间和机会偷跑到网吧去玩星际争霸、CS 这些游戏。

到了高中，发现如果你表现的足够无药可救，高中生活其实可以过的还不错。就在那时学会了去网吧包夜（我至今也想不通，为什么初中住读时周末拥有大把的机会，我没完全没有意识到过网吧是可以玩一晚上的）。

学会了网吧包夜这个技能后，能接触电脑的机会更多了，可以带着书去研究一些不懂的问题。那

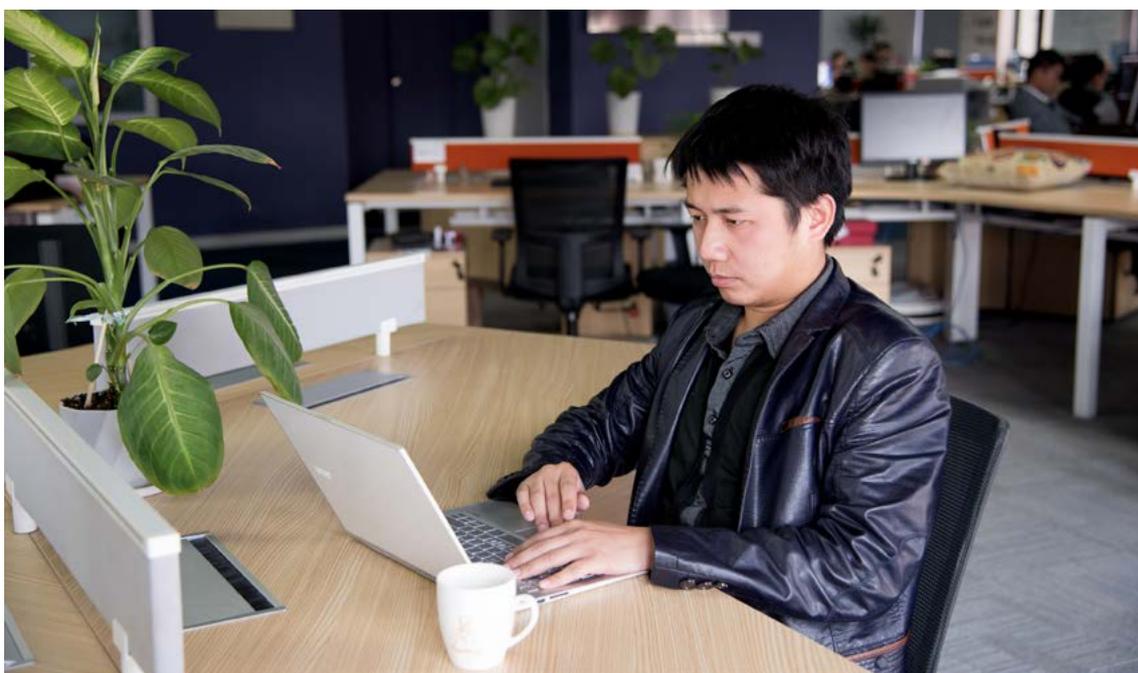
段时间，我那些刚接触电脑的中学同学比“程序员”还菜，所以打星际时经常是前半场我快速发展去掠夺一番后，切出游戏去研究如何写 HTML，等到差不多时就 A 过去结束游戏。（说到游戏，深度这些程序员的水平真是我见过最菜的一个群体了，大家有机会可以虐虐他们。）

让我想不到的是，那个时候学校图书馆竟然有不少“电脑报”这类和“学习无关”的书刊让我打发了不少时间。

不过那时还是会有一些不愉快的事。一个是，和女朋友分手了（此时我还不算程序员）——“你老这样不好好学习，以后怎么养我”；一个是，老爸因为我贪玩迁怒之下命我把那些杂七杂八的电脑书扔掉。

也因为那时条件有限，让我这个年龄的人也有机会稍微感受了下在纸上写“代码”的感觉。

不过我比较偏，先了解的是 HTML、Flash，之后是 8086 汇编，然后是逻辑电路。





在纸上画完第一个加法机正好是高考备战的最后几天了，学习 C 语言都是这之后的事情了。

高考结束后我表现的非常开心，母亲看着我开心，以为我又超常发挥，也跟着开心，直到分数下来，她才发现被我套路了。

经过一系列曲折的事件后，来到了我刚进深度时驻扎的那所学校，和大家一样肆无忌惮的玩起来了。

之后更快活充实的日子就是在图书馆中泡着。记得第一次进图书馆后我就开始担忧，这么多想看的书我怎么看的完呀。也记得在大雪纷飞的日子里，一个人一边跺着脚驱寒一边痴迷的从书上了解计算机的世界。

后来大家都出去实习了，我却突然好学起来了，觉得自己应该更系统的学习下计算机。

然后就在空无一人（此时我可以算是程序猿）的寝室楼里独自复习了一个学期。期间只和一个坐着

长途汽车来学校找我通宵玩游戏后，第二天一大早又坐着长途汽车回家的逗比说上几句话。

再后来顺利的考上了一所学校又无忧无虑的安心学习了两年。

学习的时光是快乐的，快乐时光是短暂的，短暂时光却也是无法摆脱情感上的羁绊。所以在这个下课后食堂一楼男女比例高达惊人的 500:5，我依旧在朝着成为一个好的程序员方向而努力——没有女朋友。

初入深度 | 2012 年 5 月 31 日入职深度

人如果喜欢一个事物时，就会把各种没有直接关系的现象扯在一起称之为“缘分”。

我和深度的一面之缘早在进深度的一年多前。



当时还在学校，看到一家武汉的公司在网上招聘 Linux 发行版相关工作。

“哇，不错呀。这不正是我想要加入的么。”接着一看，“我勒个去！！深度 XP 呀。你们这些人不要来祸害高尚的开源社区。”然后就关闭页面，断了这一面之缘。

但种种原因，此后一年后还是来到了深度，并和他一起成长，度过了人生中最宝贵的几年。

人生中最“枯燥”的日子也是在刚入职时公司所在的学校度过的。

第一天上班，这家奇葩的公司办公地点搬到了学校租的两间教室里。巧的是这间教室是我 3 年前上 PCB 实习课时呆过两个星期的教室。作为一名新人，带着“老人”们熟练的走着各种“捷径”，而“老人”们则带着我熟悉公司的情况。

第二天，我有点不知所措，因为那天是儿童节，公司放假了。

在深度工作的几天后。一大早来到办公室，进门看到大家竟然都来了，但个个精神恍惚，气氛有点诡异。我默默的看了下时间确认我真的没有迟到，后来到工位，从网上知道了原来这群人连夜发布了 12.06 的镜像，现在正处于待命状态，随时解救社区的小白同学们。

记忆中那是个很老的版本了，但现在翻翻发行日志，会看到那时已经有了深度音乐、深度影音、深度截图、深度软件中心、桌面环境（Deepin Gnome shell）等重要组件了。

我就这样错过了第一次的版本发布，也是在那时王勇同学天天一边骂着 Gnome Shell，一边擦干眼泪继续撸 Gnome Shell 插件。

当时我在 deepin-ui 里面帮忙打打杂，写写控件，偶尔去调试几个自己受不了的 bug。日子过的挺开心，当时离开校园没多久，来到深度后最大的感觉就是：比学校舒服多了，有了空调，冬天不冷夏天不热，晚上还不断网。和在学校时做的事基本一样，

还有人给你薪水。

直到有一天王勇拿出一本“如来神掌”出现在我面前，然后大家自行脑补“维护世界和平的重任就交给你了”这种场景。

也是从那时起，“我们要做自己的桌面环境”从一句口号转变为了实际行动，也是从那时起我变得忙碌起来了。

在深度的第二年

扯在一起的“缘分”在第二年继续验证着

由于在学校办公有着种种意想不到的阻碍，所以深度的大本营继续南迁，来到了武汉的最南边，绕城高速的入口处。巧的是正好就是之前说的那所学校的后门对面。

虽然办公地点更偏了，但场地大了好多，不用再几个人挤一起了。王勇当时还大言不惭的说，新办公室绝对宽敞，绝对坐满了。

来放张我们快要搬离时的图你们看看什么叫做偏。

不知不觉来到这里后深度的同学也好像突然变多了。当初蓝精灵（王勇的闺女）周岁生日时，深度所有的同学带家属一张桌子就能坐下。

携带第一版深度桌面环境（DDE）的 12.12 也被硬拖到 2013 年 6 月 19 日发布，我也知道了原来跳票应该用年来做计量单位。

早期为了满足





先“有”的第一要务，所以选择了 webkit 来实现桌面环境的 UI 部分，一个开发者，一个设计师就开工了。也没有什么需求列表，设计图啥的。自己先实现出想要的效果，然后喊来设计师抱着小板凳，按下 F12 出现 webkit 的调试器，大家用 css 进行交流，设计师一边想效果，一边指挥着。程序狗吐槽几句后，望了望窗外的夕阳后乖乖照改。刚开始写 DDE 时是从最外表来入手的，也就是登录器、桌面、启动器、任务栏这几个组件。其他部分就直接拿着 Gnome 的东西，后来又有一批同学开始去重写了控制中心。

刚开始是扔我一人试水，发现水没有看着那么深，还淹不死人后大家都来了。

接下来的两年里在武汉最南边的深山中，深度逐步找到了自己的定位，让更多和我当初一样带着不屑眼光的人正确的认识到了，“你竟然是这样的一个深度”。

然而第一次较大的危机也在那时展露出来了。仅靠着刘同学的一腔(坑)热(小)情和(伙)朋友(伴)的信任很难再一直支撑这支填坑专业队。估计刘同学也感觉玩大了，要不行了。所以深度那时开始做了一段时间的外包项目。虽然有了一点点收入但却也导致了不少人员的流失。

恰到好处的，国内一夜之间刮起了一阵“国产化”的风，做 Linux 这个毫无前途的公司突然在一些领域被看重了。

有了预期的市场，资本也就来了。所以，虽然当时“国产化”不是深度的目的，但确实如果当时没有这场“风”，大家也没缘分一起走下去了。

深度的成长

虽然人员翻了好几倍，但又可以像头几年一样



无拘无束的自由的成长了，然而却也不完全像当初一样了

王勇同学，从当初的害羞男孩转变为了除某同学外“最不要脸”的人了，顺带着酒量也从第一次见他一瓶就趴桌子上，到现在程序员中喝酒时属他叫的最凶。

刘同学来武汉公司探亲的次數也越来越多了（以前的武汉公司可是放养的娃），为了公司还累晕在家中过。

我自己对开源社区、计算机、感情、生活了解的也越来越多，懂得了更多自己知道与不知道的知识，也变得更加迷茫、不爱说话了。

深度本身也更加正规化了，以前用铁锹挖坑，现在直接上挖掘机了。然而生死存亡的危机却从来没有解除过，只是职责更加细分了，人也藏的更深了。这些压力我们大多数人是不要去感受的。

难能可贵的是，深度的大部分资源依旧放在了产品与社区这两个可能在永远看不到金钱收益的方向上。

原来我第一次看到深度招聘时的不屑来自于，对开源社区的认知太少了。和很多人一样误以为商业公司是万恶的，开源社区才是高尚的存在。

多少人和我一样从一个愣头青时了解到了斯托曼的自由软件运动，Eric《大教堂与市集》倡导的开源运动后，觉得这些才是一个热爱计算机的程序员该追求的呢？然而这些人中又有多少人去完全了解过他们发起的这些运动的内容呢？至少我只是在一些片段中了解到传奇般的人物后就兴奋的开始觉得那些商业软件行为就是十恶不赦。只有社区才是激奋人心奇迹般的存在。

随着自我的成长，发现我这种行为是可怕的民粹。我们对事物的认知太表面太轻浮了。所以后来“开源”、“自由”再也不会让我内心有一丝起伏。我的意思并不是说“开源”、“自由”不好，相反



我依旧很尊重那些真正在实践这些人。只是现在有太多的人到处贴标签随意攻击他人，而我自己对这些标签已经没有任何感觉了，我所热衷的只是计算机本身，越多的了解它，越多的看不透。

如今深度已经完全有能力去尝试任何的用户体验，也能在一些非 UI 部分有一些自己的思想。

围绕着 DDE 相关的一些工作也进入自己的轨道，不再缺了谁就跑不动了。2016 是 DDE 沉淀的一年。在即将发布的 2015.4 中替换了 policykit-gnome 以及 nautilus 之后不论是 UI 还是非 UI 已经完全没有 gnome 的影子了。4 年前推出“第一版”DDE 时的我也没想过深度可以完完全全实现一套独立的 DDE。然而写到这里却发现深度已经做到了。

从 windows 迁移方案，内核维护，仓库构建以及支撑整个系统持续交付所需基础设施到测试环境，每个研发部门都在各自的轨道上运行着。

现在的深度也早已离开了那荒郊野岭的园区，虽然每个人都有各自的生活，然而大家现在或曾经与深度一起走过的路，多年后回头再看一定是值得说道的路途。

这里有过你怀恋的人，有过你的青春，有过你的奋斗。

有一天就算深度不在了，deepin 却会在社区中一直存在下去，也许是另一种形式。d



第三届世界互联网大会发布 《2016年世界互联网发展乌镇报告》(全文)

● 世界互联网大会 南风一号 / 文

2016年11月16日至18日，以“创新驱动造福人类——携手共建网络空间命运共同体”为主题的第三届世界互联网大会在中国浙江乌镇召开，中国国家主席习近平通过视频发布讲话，中共中央政治局常委刘云山在大会开幕式作主旨演讲，来自全球的110多个国家和地区、16个国际组织的1600名嘉宾共聚一堂展望互联网发展前景。世界互联网大会组委会高级别专家咨询委员会（高咨委）审议并通过了《2016年世界互联网发展乌镇报告》，中英文版本于11月18日正式发布。

序言

当前，人类社会进入信息革命时期，以互联网

为代表的信息技术日新月异，引领了社会生产新变革，创造了人类生活新空间，拓展了治理新领域，极大提高了人类认识世界、改造世界的能力。世界因互联网而更丰富多彩，生活因互联网而更丰富。构建和平、安全、开放、合作的网络空间，让互联网更好造福世界、造福人类成为人们的共同意愿。

2015年12月16日至18日，第二届世界互联网大会在中国乌镇召开，来自全球120多个国家和地区的政府代表，国际组织负责人，互联网企业领军人物，专家学者等2000多名嘉宾出席大会，共商世界互联网发展大计。中国国家主席习近平在开幕式上提出，推进全球互联网治理体系变革“四项原

则”和构建网络空间命运共同体“五点主张”；巴基斯坦总统侯赛因、俄罗斯总理梅德韦杰夫、哈萨克斯坦总理马西莫夫、吉尔吉斯斯坦总理萨里耶夫、塔吉克斯坦总理拉苏尔佐达、乌兹别克斯坦第一副总理阿济莫夫、联合国副秘书长吴红波、国际电信联盟秘书长赵厚麟等 500 多位嘉宾发表意见，就全球互联网建设、发展和治理进行了深入探讨。经大会高级别专家咨询委员会提议发表的《乌镇倡议》，提出国际社会应关注互联网建设和普及，促进文化的多样性与繁荣，推进数字经济发展和普惠数字红利，保障网络空间的安全，以及推动全球互联网治理等观点。

一年来，《乌镇倡议》得到全球社会广泛关注，全球互联网基础设施建设进展显著，数字经济发展方兴未艾，各国网络文化交流互鉴成果丰硕。互联网在持续改变人类社会同时，自身也在发生着变化。人工智能、电子商务、物联网、大数据、移动通信等技术快速发展，应用日益广泛；网络金融、共享经济、工业 4.0 等正成为各个国家经济复苏的新引擎；互联网作为有史以来最为强大的信息平台正在发挥着集聚和共享人类智慧、资源和能力的重要作用。

然而，数字鸿沟从网络基础设施扩展到数字利用与素养上的能力差异，大规模的数据泄露令人担忧，对公共基础设施的网络攻击不断，用户隐私及儿童和青少年上网保护不足，新型网络犯罪、网络恐怖主义等日趋严峻，全球互联网发展不平衡、规则不健全、秩序不合理等问题依然存在，互联网发展与治理仍然面临复杂问题和严峻挑战。

值得欣慰的是，2016 年我们看到了国际社会的共同努力。完成 10 年回顾的信息社会世界峰会 (WSIS) 进入新的发展阶段；联合国互联网治理论坛

(IGF) 开启第二个十年进程；G20 杭州峰会通过了《数字经济发展与合作倡议》；联合国信息安全政府专家组 (UNGGE) 确认，包括国家主权原则在内的《联合国宪章》等国际法准则适用于网络空间；互联网数字分配机构 (IANA) 职能移交顺利完成，互联网名称与数字地址分配机构 (ICANN) 改革取得进展。这为各方在网络时代提供了可预期、确定性的保障，为各国携手构建网络空间命运共同体做出积极贡献。

一、互联网进一步普及 推动弥合数字鸿沟

- 互联网基础设施建设成就显著。各国加快互联网基础设施建设，移动蜂窝网络已覆盖 95% 的人口。预计 2016 年底，全球固定宽带用户数将达到 8.84 亿，移动宽带用户数将达到 36 亿。发达国家和发展中国家分别有 75% 和 50% 的用户带宽达到 10Mbps。截至 2015 年底，全球大多数发达国家、83 个发展中国家和 5 个最不发达国家都已实现宽带委员会提出的价格可承受性目标。过去一年，全球 IPv6 增长翻番，有力支持未来基础设施的发展。

- 信息通信技术领域保持创新活力。信息通信技术继续成为全球研发投入最集中、创新最活跃、应用最广泛、辐射带动最大的技术创新领域。2015 年，在世界知识产权组织专利合作条约 (PCT) 框架内，计算机技术类、数字通信类和电子机械类专利申请数量位列前三。2016 年美国麻省理工学院 (MIT) 发布的全球十大科技突破性技术中，涉及信息通信技术领域的占 6 项。

- 云计算、大数据、物联网等产业规模高速扩张。全球云计算市场规模预计 2016 年至 2020 年复合年增长率达 22%，大数据市场规模预计从 2015 年的 0.14 万亿美元将增长到 2020 年的 1.03 万亿美元。



随着“工业 4.0”、“工业互联网”等概念的出现，对海量数据深度挖掘和专业分析、庞大计算能力等的需求，培育了新的市场，云计算、大数据、物联网技术三者结合，成为向各行业渗透的重要切入点。智慧城市项目全面启动，中国、印度、韩国、新加坡、美国等国均推出了智慧城市建设和项目。

- 弥合数字鸿沟取得实质性进展。全球互联网用户保持增长，从 2015 年 32 亿提升至 2016 年 35 亿，互联网普及率达到 47.1%，但仍有半数以上人口未使用过互联网。发达国家互联网用户普及率超过 80%，而最不发达国家和地区网民数量（2.7 亿）普及率仅为 23.5%。金砖国家果阿峰会认为应采取多维度、包容性措施，致力于消除发达国家和发展中国家之间的数字和技术鸿沟。

二、网络文化繁荣发展 交流互鉴不断深化

- 国际组织积极推动网络文化多样性。联合国通过的《2030 年可持续发展议程》明确提出信息技术对保护文化创意和文化多样性的重要作用。WSIS 十年审议决议提出要用数字技术保护和记录文化遗产。G20 杭州峰会《数字经济发展和合作倡议》主张要通过网络空间繁荣文化，认可多语言原则的重要性。经济合作和发展组织 (OECD) 提出，将年轻人对全球性问题的理解以及对文化多元性和文化宽容的态度纳入“国际学生评估项目 (PISA)”。

- 互联网对多语种的支持让更多文化内容得以展现。人类文化以数字化形式在网络空间持续繁荣，呈现多样化发展态势。2016 年 7 月，联合国教科文组织启动“世界语言地图”项目，促进世界语言在互联网上的发展。互联网应用所能支持的语言种类已超过 300 种，网络的包容性让更多文化成果得以

传承。多语种域名发展取得明显进展，包含保加利亚、印度等 39 个国家和地区提交申请的 53 个多语种国家和地区顶级域字符串通过 ICANN 评估流程，其中 43 个已入根。

- 移动社交平台为不同文化群体的交流提供便利。移动社交平台对文化多样性的促进作用日益显现，2016 年移动社交用户达 24.4 亿人，成为人们网络社交应用的主流。得益于人工智能技术，语音识别、翻译软件等应用取得显著发展，进一步便利了不同语言间的文化交流。

三、数字经济持续发展 各方共享数字红利

- 多国和地区加强网络安全顶层设计。2016 年，全球多个国家和地区出台或完善网络安全战略规划，颁布法律法规，设立专职机构，完善机制建设，开展宣传教育，提升全民网络安全意识，培育网络安全文化，加强能力建设和国际合作。

- 关键基础设施和数据保护成为全球关注重点。各国政府纷纷将关键基础设施保护提升到国家安全层面，发布政策和标准，开展专项行动，着力提升防护能力。各方高度重视数据保护的标准建设，积极探索跨境数据流动的规则，探讨分享最佳实践。

- 各方积极应对网络犯罪与网络恐怖主义威胁。网络犯罪与网络恐怖主义严重危及全世界的和平安全。各国致力于通过跨国司法与安全合作，有效打击网络犯罪与网络恐怖主义，防止将技术、通信和资源用于犯罪或恐怖主义目的，根除恐怖主义和极端思潮在网络空间传播的基础与土壤。打击网络犯罪的双、多边合作机制相继建立。

· 各国和地区加快发展数字经济。世界经济加速向以网络信息技术产业为重要内容的经济活动转变。在电子商务、金融科技、互联网媒体和数字物流等领域，创新型 ICT 企业和互联网初创企业正在创造新的商业模式。2015 年以来，欧盟推动数字化单一市场建设。2016 年，G20 杭州峰会宣布将数字经济作为全球经济增长日益重要的驱动力；OECD 召开的数字经济部长会议确认了数字化议程的四个关键政策领域。

· 电子商务快速发展。互联网推动更多跨境货物、服务贸易，使更多消费者和企业摆脱国家边界限制。据预测，2016 年，全球电子商务零售市场规模将达 1.9 万亿美元，其中欧洲电商市场规模将达 0.51 万亿欧元，中国网络零售额将达 0.8 万亿美元。

· 数字红利有力支撑全球可持续发展。互联网通过消除信息障碍、优化要素配置等，降低了全社会的信息、交易成本，为实现联合国《2030 年可持续发展议程》的目标提供有力支撑，使得边远地区、贫困家庭以及各类弱势群体的人们都有机会享受数字红利。

四、网络安全问题凸显 国际社会积极应对

· 多国和地区加强网络安全顶层设计。2016 年，全球多个国家和地区出台或完善网络安全战略规划，颁布法律法规，设立专职机构，完善机制建设，开展宣传教育，提升全民网络安全意识，培育网络安全文化，加强能力建设和国际合作。

· 关键基础设施和数据保护成为全球关注重点。各国政府纷纷将关键基础设施保护提升到国家安全层面，发布政策和标准，开展专项行动，着力提升防护能力。各方高度重视数据保护的标准建设，积极探索跨境数据流动的规则，探讨分享最佳实践。

· 各方积极应对网络犯罪与网络恐怖主义威胁。网络犯罪与网络恐怖主义严重危及全世界的和平安全。各国致力于通过跨国司法与安全合作，有效打击网络犯罪与网络恐怖主义，防止将技术、通信和资源用于犯罪或恐怖主义目的，根除恐怖主义和极端思潮在网络空间传播的基础与土壤。打击网络犯罪的双、多边合作机制相继建立。



五、网络空间国际治理进展积极 新成果令人期待

· 网络空间国际规则对话积极活跃。第五届联合国信息安全政府专家组会议聚焦网络空间国家行为规范及国际法在信息通信技术领域的适用、信任措施等问题，取得良好开端。上合组织元首理事会会议发表《塔什干宣言》，支持在联合国框架内制定网络空间负责任国家行为的普遍规范、



原则和准则。G20、OECD、金砖国家、东盟等均积极开展合作，共同应对网络空间的威胁与挑战。

· 国际社会积极推动互联网治理体系构建与完善。更多国家强调尊重各国平等参与全球网络空间治理的权利，认为应在尊重主权和不干涉他国内政原则基础上加强合作，构建和平、安全、开放、合作的网络空间治理新秩序。2016年WSIS与IGF启动新的十年进程，支持包容性与可持续发展成为治理主题。G20杭州峰会一致认为互联网治理应继续遵循WSIS成果，强调政府、私营部门、民间社会、技术团体和国际组织等各方应根据其各自的角色和责任充分、积极参与互联网治理。国际电信联盟（ITU）、世界经济论坛（WEF）等国际机构持续讨论网络空间发展治理问题。IANA职能管理权移交完成，各方期待ICANN进一步国际化。

· 国际社会持续致力于儿童在线保护。国际社会高度重视有害信息对未成年人的影响，制定行动计划，推动各方参与，开通投诉热线，开展国际合作，有效打击网上儿童色情，治理网络欺凌等。第二届“儿童在线保护”（We Protect）全球论坛上，41个参会国共同签署联合行动声明。

展望未来

2016年11月16日至18日，以“创新驱动造福人类——携手共建网络空间命运共同体”为主题的第三届世界互联网大会在中国浙江乌镇召开，中国国家主席习近平通过视频发表讲话，中共中央政治局常委刘云山在大会开幕式作了主旨演讲，来自全球的110多个国家和地区、16个国际组织的1600名嘉宾共聚一堂展望互联网发展前景。我们，大会高级别专家咨询委员会委员认为，今后一段时

间，全球互联网发展和治理将呈现如下趋势：

1、发展中国家将继续在全球互联网发展中保持令人瞩目的势头，数字鸿沟问题仍然备受关注，数字红利逐渐惠及全人类，以促进联合国《2030年可持续发展议程》目标的实现。

2、网络空间文化交流将更加频繁，网络文化日益繁荣，文化多样性获得更多的理解与尊重。

3、各国将继续保持对网络安全问题的高度重视，在尊重网络主权、尊重《联合国宪章》等国际法和国际关系准则基础上，制定各方普遍接受的网络空间国际规则成为国际社会的共同愿望。

4、多边参与、多方参与将成为互联网治理常态，政府、国际组织、互联网企业、技术社群、民间机构、科研院所、公民个人等各个主体积极作为，共同推动“共享、共治”的务实合作进一步深化，为实现互联网的可持续发展做出实质性贡献。

5、网络安全和互联网治理的国际交流与合作将成为国际社会最关注的话题之一，越来越多的国际组织将在推动全球互联网发展与治理方面发挥更加积极的作用。

互联网是人类共同家园，我们期盼国际社会戮力同心、相向而行、同舟共济，携手共建网络空间命运共同体，加强沟通交流，开展国际合作，创新驱动发展，促进全球网络基础设施互联互通，促进网络文化交流互鉴，促进数字经济开放发展，促进网络空间和平发展，促进全球互联网治理体系更加公正合理，推动网络空间互联互通、共享共治，更好地造福全世界，开创人类发展新未来。 d



为何境外 IT 巨头频频来华成立合资公司

● 铁流 / 文

9月20日，微软与中国电子科技集团正式宣布合资成立神州网信技术有限公司。根据中国电科与微软在乌镇世界互联网大会上签署的备忘录，中国电科将占合资公司51%股份，微软占49%。合资公司的目标是为中国政府和关键基础设施领域的国企用户提供符合“安全可控”要求的Windows 10操作系统。

有观点认为，通过和境外IT巨头合作，有可能掌握其核心技术，借力IT巨头的成功生态圈加速管理变革并与国际接轨，同时也能够顺应我国政府“保障信息安全、核心技术自主可控”的科技发展战略。

但也有观点认为，从长远发展的角度分析，这种做法对于外资企业有利，对国家信息安全来说却存在很大的挑战和隐患。

境外 IT 巨头合资的目的在于获取高额利润

随着西方经济持续低迷不振和中国经济蓬勃发展，国家又相继出台一系列网络安全和自主可控政策，越来越多的境外IT巨头来华寻找战略合作伙伴，以成立合资公司的方式将品牌本土化，一些产品甚至打进党政军市场。

2012年，EMC与联想宣布成立合资公司

LenovoEMC Limited, 其中联想占股 51%, EMC 占股 49%, 携手进军服务器和存储市场。

2013 年, 上海兆芯集成电路有限公司成立, 上海兆芯由上海市国资委下属上海联和投资有限公司和 VIA 公司合资成立, 注册资本 2.5 亿美元, 研发中国自主知识产权的 X86 架构核心处理器芯片。

2014 年, IBM 通过 Open power 基金会向中国本土企业开放 Power 技术, 同年, 苏州中晟宏芯发布了 Power CPU 芯片 CP1, IBM 并与华胜天成合资成立新新东方, 2015 年, 基于 Power 微处理器以及 AIX 操作系统的全系列新新东方国产服务器上市。

2015 年, 惠普公司出售其中国大陆服务器和储存系统 51% 的股权给清华紫光, 并与清华紫光合资成立新华三公司。

2015 年, 思科与浪潮在中国成立合资公司, 其中浪潮占股 51%、思科占股 49%, 业务范围包括网络、数据中心、云服务、物联网等领域的产品及技术。

2015 年, 重庆市政府与 ARM 公司在渝签署合作谅解备忘录, 根据谅解备忘录, ARM 将和重庆一起, 在渝北区仙桃数据谷建立 ARM 产业生态园, 引入 ARM 的集成电路创新加速器计划和 ARM 大学计划, 通过产、学、研的结合, 推动重庆集成电路设计企业快速发展。

2016 年, 高通和贵州省政府合资成立签署战略合作协议并为合资企业——贵州华芯通半导体技术有限公司。贵州华芯通半导体技术有限公司首期注册资本 18.5 亿人民币, 贵州方面占股 55%, 高通以技术出资, 占股 45%。

2016 年, Intel 公司、清华大学和澜起科技(上海)有限公司在北京正式签署协议, 宣布联手研发融合可重构计算和英特尔 x86 架构技术的新型通用 CPU, 以满足市场和用户需求。

2016 年, AMD 公司与天津海光公司达成协议设立合资公司, 按照协议, AMD 向合资公司提供

x86 芯片技术许可, 合资公司将利用该技术开发只在中国销售的服务器芯片; 作为回报, AMD 预计可获得 2.93 亿美元许可费和版税收入。

2016 年, 9 月 20 日, 微软与中国电子科技集团宣布, 双方筹建的合资公司神州网信技术有限公司正式成立。根据中国电科与微软在乌镇世界互联网大会上签署备忘录, 中国电科将占合资公司 51% 股份, 微软占 49%。合资公司的目标是为中国政府和关键基础设施领域的国企用户提供符合“安全可控”要求的 Windows 10 操作系统。



在世界互联网大会上的签约仪式

这种以共同组建合资公司的形式, 积极推行境外品牌本土化战略, 既显示了境外巨头对中国市场的重视, 某种程度上也是对中国市场做出的重要让步。但本质上是境外 IT 巨头在国家出台对自主可控的政策要求下的妥协产物——由于境外 IT 巨头无法舍弃中国这个全球第二大 IT 市场, 遂借助合资的方式取得合法身份, 减少国家政策和当地社会对外来资本的危机情绪, 深度介入蒸蒸日上的中国 IT 市场甚至敏感行业市场, 从而获取丰厚利润。

这种做法虽然对外资有利, 但对中国未必有利可图。虽然部分源代码和技术的开放, 为我们提供了加快技术发展的机遇, 但目前这种浅层次的合作授权模式, 难以做到期望的核心技术消化吸收, 难以改变核

心技术受制于人、信息安全受制于人的现状。

无法掌握核心技术

对于 IT 产品而言，真正意义上的本地化要求很高，涉及到产品研发、产品制造、供应链、人力资源、品牌、营销管理、市场开拓等多个方面，其中，人力资源本土化和研究开发本土化是最根本最深刻的本土化，也是我们最需要的本土化。而成立合资公司，充其量只是实现了品牌本土化，并不改变国外企业掌握核心技术的实质。

对于国外 IT 品牌的本土化，目前其绝大部分做法都是向国内企业提供“授权”而非“转移”知识产权，“授权”的目的仅仅在于可供国内企业能够合法地使用国外企业的知识产权，国内企业并不真正拥有该知识产权，因此在核心技术及其发展决策方面，即便合资企业由国内资本控股，国内企业也仍然不具备核心技术及其发展的话语权。

举例来说，微软和中电科的主要合作内容并没有涉及核心技术和知识产权，也不涉及 Windows 系统内核开发能力培训、以及技术共享和转让。而微软 Windows 和设备集团副总裁 Yusuf Mehdi 也表示，“微软将保留所有关于 Windows 10 的技术知识产权”。

顺便说一下，IBM、Intel、ARM 的授权也是类似，国外厂商为了长远自身利益考虑，目前也并没有在核心技术或知识产权上对中国进行开放：IBM 在针对 Power CPU 的对外授权中，浮点运算单元等关键部件是受到美国出口限制的；Intel 公司也并不向任何第三方提供 X86 指令集授权；虽然 ARM 公司在授权方面相对 Intel 更加开放，但也在授权同时提供了很多限制性条款，如非永久性授权、授权指令集不得自行修改扩充、被限制使用范围等……

目前信息产业大部分核心技术掌握在以美国为首的西方国家中，美国对中国的遏制策略是长期国

策，对于真正重要的信息产业核心技术，即使美国企业出于自身利益考虑希望出口，美国政府层面也不可能同意。

过去，国内应用系统生长在国外基础技术平台之上，Intel 和微软等国外厂商把基础软硬件做成了黑盒子，导致应用软件开发人员对软硬件底层机理不清，安全无法得到保障。现在，出于国家政策和自主技术越发成熟的因素，国外厂商愿意把部分代码授权给国内，使过去的黑盒子变成灰盒子，以表明开放合作态度。

但事实上，国外厂商的开放只是有限程度的开放，对于一些关键的代码或者说明，很难做到完全公开透明，所谓灰盒子相对于黑盒子仅仅是五十步和一百步的区别。即便微软全部如实开放源代码，在相关的设计文档和消化吸收的能力未必能得到充分保障的情况下，合资公司很难有能力看懂吃透 Windows 上千万行的源代码，更不会拥有相应的机理分析、修改与安全保障能力。

信息技术产业技术引进成果不尽如人意

对于信息产业而言，关键是要形成自主创新的能力，而不是单一产品或技术的引进。对于 CPU 和操作系统这样的高技术产品，其产品或技术的实现以及后续发展主要依赖于自身形成的能力，而能力是无法通过技术或产品的转移来获得的。

事实上，目前开展技术引进的合资或合作公司，大多都没能形成自己的能力，而且在商业上也并不成功，一些公司甚至只能依赖政府输血过日子。

举例来说，与 VIA 成立合资公司已逾 4 年，在十二五期间还承接了核高基 1 号专项获得资金数十亿元人民币，但在自主创新上却乏善可陈——ZX-A 是 VIA Nano 的马甲，ZX-C 的 CPU 核与 ZX-A 的 CPU 核一致，都是 VIA 的以赛亚，而且至今依旧没有设计出自己的 CPU 核，始终在使用 VIA 的技术穿

马甲。由于 CPU 核是决定 CPU 性能、安全、功耗、成本的最重要因素，在设计 CPU 的过程中，80% 以上的功夫都用在 CPU 核上，日常我们所说的双核、四核、八核芯片，其实是将 2、4、8 个 CPU 核的接口互联并集成到一片硅片上。

苏州中晟宏芯发布的 CP1 也是 IBM Power8 的马甲，原本计划于 2016 年完成的 CP2 至今依旧处于“PPT 状态”，并被推迟到 2018 年。另外，在中晟宏芯在今年 3 月还爆出欠薪事件。

很显然，这些公司在获得境外技术后的 3-4 年后，依旧尚未吃透技术，设计出自己的 CPU 核，只能拿别人的产品穿马甲，原因何在？因为 CPU 设计是一项非常复杂的工程，要在指令分支预测、寄存器重命名、多指令通路、乱序发射、功能部件、访存性能等方面下功夫。即便获得国外公司提供的全部源代码，但如果没有设计文档和吸收消化能力，也是很难吃透的，更何况国外公司的技术授权还有所保留。

操作系统和 CPU 同样是一项非常复杂的工程，这里以 Linux 操作系统为例，Linux 已经开源 20 余年，国内多个单位也基于 Linux 开发出中标麒麟、银河麒麟、深度等国产 Linux 操作系统，但 Linux 内核基本上是国外程序员在维护，本土品牌操作系统厂商至今仍然不敢说掌握了系统的所有细节。

微软与中国电子科技集团成立的合资公司会步上述公司的后尘吗？值得深思。



深度操作系统

Windows 能多大程度上取代国产操作系统

相对于还处于成长期的国产 Linux 操作系统，Windows 在技术上更成熟，在用户体验上也更好，特别是在软件生态上远远优于国产 Linux 操作系统。那么，相对于银河麒麟、中标麒麟、深度等国产 Linux 操作系统，Windows 在党政军市场的竞争力到底有多强，能多大程度上取代上述国产 Linux 操作系统呢？

在讨论 Windows 与国产操作系统竞争之时，首先要明确一点，那就是这并非两个操作系统之间的竞争，而是整个 Wintel 联盟和国产 CPU+ 国产 Linux 之间的竞争。

目前，党政军办公电脑中存量最大的还是 X86 芯片，X86+Windows，x86+Linux 占据了大部分市场，而龙芯 + 中标麒麟，申威 + 睿思，飞腾 + 银河麒麟也占据一定的份额。对于 x86+Linux 的情况，Windows 不存在任何替代的技术障碍。但因为 Windows 不支持龙芯和申威，对于运行于龙芯和申威 CPU 的国产 Linux 操作系统，Windows 则无法撼动。

银河麒麟的情况则更为特殊，银河麒麟由国防科大研制，早期版本采用 FreeBSD 内核，现在的版本采用 Linux 内核，由于飞腾 CPU 也由国防科大研制，因此在客观上已经形成了飞腾 + 银河麒麟的组合。加上国防科大的背景，飞腾 + 银河麒麟大多被用于军队——日前，中国航天科工二院 706 所自主研发的“通用机架式服务器”和“台式计算机”以总分第一的成绩成功中标中央军委装备发展部“新一代自主可控计算机”项目，该项目使用的就是飞腾 ARM+ 银河麒麟。

另外，虽然像深度等国产 Linux 操作系统也是支持 ARM 的，但由于飞腾不对其他操作系统提供支持，因而很难很稳定地运行，据业内人士分析，内部硬件可能有 watchdog，非麒麟系统差不多每 30 分钟

就自动重启。正是飞腾+银河麒麟高度捆绑的特点，对于银河麒麟操作系统，Windows 也无法取代。



中标麒麟操作系统

阻碍我国信息安全产业的自主发展

信息技术产品与一般的工业产品不同，信息化产品软硬件之间互相依存，构成了一个庞大的信息产业生态体系。而这个体系中的任何一个点技术（芯片或软件）都不是目的，而是控制产业生态的手段。单纯地采取国外品牌本土化策略，容易造成国外 IT 生态对国内产业的进一步垄断，反而不利于我国信息安全产业的自主发展。

自 2000 年以来，中国在基础软硬件方面奋发图强，实现了从无到有的根本转变，填补了很多该领域的空白——在 CPU 方面有了龙芯和申威；在操作系统方面有了中标麒麟、银河麒麟和深度；在数据库方面有了达梦、金仓；在办公软件方面有了 WPS……已经初步具备了自主研发软硬件的能力，虽然没有 Wintel 好用，但也可以满足党政军最基本的办公需求。

若在此时，放 Windows 进入党政军市场，国产操作系统相较于 Windows 如同是让成年人与幼儿同场竞技，最好的结果也是国产 CPU+ 国产 Linux 只能守住现有阵地，最差的结果则是国产 CPU+ 国产 Linux 落败于 Wintel 联盟——由于软件和硬件的高度一体化，龙芯和申威跑不了 Windows，这必然导致

自主基础软硬件在 Wintel 技术联盟面前全面溃败，将会使刚刚初具雏形的自主基础软硬件被扼杀于萌芽状态。

事实上，马甲产品在一些人的操作下，已经成功打进党政军市场——根据科技部官网的文章《核心电子器件、高端通用芯片及基础软件产品科技重大专项交流材料》显示：采用 ZX-C 的联想 II 代整机应客户要求于 2015 年底正式交付 5800 套产品，并将在军队和政府保密系统中进行上千套规模的试点应用。

这对于国内自主企业的发展造成了不利的局面，打乱我国在相关安全领域发展自主可控产业的已有布局，也对真正自主企业的发展造成了市场、舆论以及人才培养方面的极大干扰。

结语

我国自主软硬件厂商经过十几年的积累，已掌握了部分核心技术，建设了初步的产业生态体系，具备了在部分关键领域进行国产化替代以及在部分市场上参与竞争的能力，可以说正处于持续发展的艰难爬坡期。

在这个关键时期，本土国产厂商需要国家对自主产业强化政策和资金扶持，在与国家信息安全相关的敏感部门及关键行业优先考虑采用自主产品，通过财政和税收等手段支持这些企业的科研投入。而穿上国产化马甲的国外产品，则很有可能凭借其技术成熟度和背后的势力抢占党政军市场，扼杀来之不易的自主技术。因此，在笔者看来，一系列合资公司的本质是国外 IT 巨头打入中国的特洛伊木马。d

（作者微信公众号：tieliu1988）



面对境外 IT 巨头的合资潮 可以采取三项措施应对

● 铁流 / 文

近年来，随着网络安全和自主可控等政策的出台，在催生了规模可观的国产化 IT 市场的同时，也重创了国外企业级 IT 厂商的在华业务，为此，IBM、Intel、AMD、惠普、思科、微软、高通、EMC、VIA 等境外 IT 巨头纷纷积极寻找中国本土的战略合作伙伴。

有观点认为，通过和境外 IT 巨头合作，有可能通过这样的合作帮助其掌握核心技术，借力 IT 巨头的成功生态圈加速管理变革并与国际接轨，同时也能够顺应中国政府“保障信息安全、核心技术自主可控”的科技发展战略。

不过从长远发展角度分析，这种做法对中国的信息技术水平提升，虽有一定作用，但对整个国家的信息安全来说，却存在挑战和隐患。

举例来说，近期微软和中电科的主要合作内容，并没有涉及核心技术和知识产权，也不涉及 Windows 系统内核开发能力培训、以及技术共享和转让。而微软 Windows 和设备集团副总裁 Yusuf Mehdi 也表示，“微软将保留所有关于 Windows 10 的技术知识产权”。另外，据小道消息称，具体合作方案为中方每年向微软支付 2 亿美元授权费，合作期 10 年，且每年授权费增加 5000 万，每年还要交 500 万的服务费，每套 license 交 50 美元。如果

事实真的如此，那这种合作压根不涉及任何技术转让，仅仅是给微软一个马甲而已。

因此，这种做法虽然对外资有利，但对中国未必有利可图。虽然部分源代码和技术的开放，为我们提供了加快技术发展的机遇，但目前的这种浅层次的合作授权模式，难以做到期望的核心技术消化吸收，难以改变核心技术、信息安全受制于人的现状。

更为重要的是，如果没有合理有效的监管和引导，还容易造成国外 IT 生态对国内产业的进一步垄断，无助于国内产业发展与技术能力的培养，影响我国信息安全产业的自主发展进程，动摇自主产业格局和生态。

习近平总书记曾多次强调信息安全和产业发展要统筹兼顾，指出“安全和发展是一体之两翼、驱动之双轮。安全是发展的保障，发展是安全的目的”、“没有网络安全就没有国家安全，没有信息化就没有现代化”，明确提出中国“要有自己的技术，有过硬的技术”，号召建设“战略清晰、技术先进、产业领先、攻防兼备”的世界一流网络强国。

为确保国家信息安全战略的实施和我国信息产业的持续发展，可实施以下措施：

一是加强国家对国企和地方政府引进国外 IT 核心技术的思想引导和统筹规划，建立健全审查审批制度，事先审查，避免以经济效益为唯一指标盲目、重复引进。

信息产业的发展不仅仅需要考虑到经济因素，也要考虑到国家信息安全战略因素。因此，在企业和地方政府引起国外 IT 核心技术、与国外厂商成立合资公司的问题上，国家宜加强统筹规划和指导，兼顾安全与发展、市场效益与本土保护，避免企业和地方政府盲目、重复引进，造成安全隐患，甚至对本土信息产业的健康发展造成冲击。

二是建立健全自主可控测评认证标准，对已引进或引进后的 IT 项目进行分级认证，严格确保党政机关、关键行业的信息安全。自主可控是我国信息化发展的重要目标，同时也是对相关企业、科研院所自身能力和进入党政机关、关键行业市场资质的重要评估标准。自主可控包含的范围很广，除了关键核心技术自主可控外，还包括技术能力自主可控、发展主动权自主可控、供应链自主可控等。

最基本的，在党政机关和关键行业，需要从本土或合资厂商对体系架构修改能力、核心技术掌握程度、核心技术修改能力、关键模块自主替换能力等方面做定量评估。建议有关部门深入探讨，以自主可控、安全可信为基本原则，兼顾生态建设、产业发展，制定相关测评认证标准，在政府机关和关键行业建立严格的审查制度，强化国家信息安全战略得到有效实施。

同时，该测评认证标准可以倒逼引进国外技术、与国外厂商成立合资公司的本土厂商加强自身能力建设，做好国外先进技术的承接和真正落地。在具体实施过程中，各厂商也要做好规划，可以按照安全透明、安全可控、自主安全的步骤实施。

其中，安全透明是对引进的设计从实现原理上进行分析，确保没有不掌握的黑盒子和恶意后门。安全可控是持续消化吸收，进行针对性地安全分析，如果有安全问题能够自主修改。自主安全是在坚持自主发展原则的基础上，引进消化吸收国外先进技术所要实现的最终目的，全面形成自主设计能力，融入自主的安全技术和安全标准。

三是加强本土信息产业生态体系的建设与扶持力度，确保行业话语权，最终建成自主可控、健康健全的信息产业生态体系。

信息产业在宏观上表现有明显的生态性，如产品生态、应用生态、开发环境生态等等，其健康有序发展不是仅仅依靠几项核心技术的突破就能保证的，而需要从产业链上下游协作和生态体系建设等方面实现。

因此，我国信息化发展过程中要积极布局产业生态体系的建设，尤其是政府有关部门要从政策制定、产业引导、资金投入上积极部署，逐步摆脱对国外核心知识产权与生态系统的依赖。加大技术创新投入力度，在国际和国内市场争取更多的行业话语权。d



制定统一标准 做大国产操作系统

● 政府采购信息网 / 文

“在中国市场，微软操作系统的市场份额超过了90%，而国产操作系统的份额只有3%左右。”中国工程院院士倪光南表示，“实际上，国产桌面操作系统在技术和体验方面，已经与国外产品水平相当，与国外的差距主要体现在生态系统方面。”

10月9日，习近平总书记在中共中央政治局第三十六次集体学习时强调，加快推进网络信息技术自主创新，朝着建设网络强国目标不懈努力。“加快推进国产自主可控替代计划，构建安全可控的信息技术体系。”倪光南表示，中国现在是网络大国，但还不是网络强国。一个重要区别是网络强国必须

能靠国产自主可控的技术设备来实现信息化。反之，如果只能依赖别人的技术设备实现信息化，那就好比在别人的墙基上砌房子，再大再漂亮也可能经不起风雨，甚至会不堪一击，也就不可能成为网络强国。

倪光南指出：“我国信息通信业几十年的实践已经证明，真正的核心技术是买不来的，是市场换不到的。中国发展到了现在这个阶段，连比较重要的技术人家都不会给你，更不要说核心技术了。”因为这类技术关系到一个国家的核心竞争力，被所拥有的国家奉作“定海神针”、“国之重器”，不能随意开放、随意买卖。所以最关键最核心的技术必须靠自己研发、自己发展。

操作系统就是这样的核心技术，也是我国亟待突破的重要领域。目前，全球各式各样的智能终端已达百亿量级，桌面终端是其中之一，桌面终端（电脑）操作系统是信息核心技术之一，目前全世界的桌面操作系统市场几乎都被 Windows 所垄断。桌面操作系统与网络安全、信息安全有重大关系，例如谁掌握了桌面操作系统，谁就能够很容易地掌控亿万桌面电脑所产生、存储和传送的海量数据；而且它又是信息系统的主要组成部分，在很大程度上决定了系统的安全。

倪光南认为，桌面操作系统是很好的突破口。桌面操作系统主要用于办公等特定场景，目前国产桌面操作系统在技术和体验等方面与国外产品相比已无明显差距。现在某些较大规模的示范应用表明，国产软硬件构成的信息系统不仅已“能用”，而且已基本“好用”了，真正的差距体现在生态方面。“Wintel 联盟的 Windows 之所以能



够垄断操作系统市场，就是因为他们推出了统一的标准，几乎所有的外设、应用程序都能做到即插即用、即装即用。而我国芯片厂商有 6、7 家，操作系统不到 10 家，大多各自为政，这样一来排列组合有几十种软硬件方案，很难实现外设和应用程序的兼容。”

“所以我们需要打造自己的 Wintel 联盟来统一标准。”倪光南表示，由于起步较晚，我国很难像 Wintel 联盟一样，由两家巨头企业来引领行业发展，为此我国“产学研用”各方共同发起成立了“中国智能终端操作系统产业联盟”（以下简称“联盟”），共同推动国产操作系统的发展。他透露，联盟正在着手制订一个统一的“桌面操作系统参考平台”标准（以下简称“参考平台”）。参考平台与那些约束性较弱的标准（如 POSIX 标准等）不同，是一种约束性很强的标准。例如，通过前者符合性测试的软硬件未必能彼此兼容，而通过后者符合性测试的软硬件一定能彼此兼容，正如通过 Wintel 平台符合性测试的软硬件一定能彼此兼容一样。参考平台有望在今年年底出台。

倪光南同时强调，我们提出的自主创新，不是关起门来搞研发，一定要坚持开放创新，但对于技术合作事宜要从严把关审批。对于类似这样的“技术合作”，他建议要评估是否能真正获得“先进技术”，是否会因此受到控制。美国陆军参谋长高级顾问 Maren Leed 曾表示：“网络武器可以在多个时间点发起攻击，包括针对早期开发的过程。”历史上曾有超级大国入侵他国工业研发体系，进行网络战场预制（使其便于超级大国进行网络攻击）的先例，我们应当对此引起高度重视。d

深度科技致力于为用户提供友好、美观、易用的操作系统，为了改善 Linux 平台上应用软件生态，除自己研发了深度影院、深度截图等应用，还建立深度软件商店为用户提供更多应用软件，努力打造 Linux 软件生态。深度科技作为国产操作系统生态打造者，在完善自有软件的同时，近年来联合国内外软硬件企业，开发了多款符合中国用户需求的应用软件。

深度科技在桌面操作系统生态多年的运营经验和努力，将会给广大开发者及用户带来更多、更好的应用软件，Linux 平台上的软件生态也会越来越完善。



航天福昕阅读器

福昕阅读器是福昕公司推出的 Foxit Reader 首款简体中文版本，它是一款免费的 PDF 文档阅读器和打印机，具有令人难以置信的小巧体积，启动迅速且无需安装。

深度科技早在之前就已经开始做福昕阅读器的适配开发，此次双方正式签署战略合作协议，旨在更好的促进双方的合作发展。在后续开发过程中，将分为两个版本：

社区发行版：即预装在基于 x86 平台的、以面向全球个人 / 家庭市场为主体的航天福昕阅读器深度版；

专业发行版：默认集成航天福昕阅读器深度版，根据项目需求为客户装机。





有道词典

外语翻译是用户在日常工作和生活中经常遇到的，而 linux 桌面系统中始终缺乏一个用户在应用中得心应手的翻译软件。在早前开源系统中有一些翻译软件，但在交互性及本地化方面都存在诸多问题。

有道词典是由网易有道出品的全球首款基于搜索引擎技术的全能免费语言翻译软件，为全年龄段学习人群提供优质顺畅的查词翻译服务。有道词典囊括互联网上的流行词汇与海量例句，集成中、英、日、韩、法多语种专业词典，新增的图解词典和百科功能，提供了一站式知识查询平台，能够有效帮助用户理解记忆新单词！



双方经历了长达 5 个月的联合开发，在易用性和操作界面及智能精准等方面不断地改进，针对于早期 linux 下星际译王版本还需要自行添加词典档配合使用，有道词典 linux 版则更方便易用，下载与当前系统对应版本的安装包后，双击运行安装即可为用户提供准确、高效的翻译。同时，有道词典 linux 版本也是中国第一个基于 linux 下的互联网商业翻译软件。



网易云音乐

网易云音乐是一款由网易开发的音乐产品，依托专业音乐人、DJ、好友推荐及社交功能，在线音乐服务主打歌单、社交、大牌推荐和音乐指纹，以歌单、DJ 节目、社交、地理位置为核心要素，主打发现和分享。2015 年 7 月 14 日，网易云音乐宣布其用户数突破 1 亿，累计产生 1.2 亿条乐评和 2 亿次的音乐分享。



由国内操作系统团队武汉深之度科技有限公司与网易公司联合开发的网易云音乐 Linux 版于 5 月 25 日正式发布，这也是国内在线音乐应用中的首个 Linux 版！用户可以享受和其他多系统（Windows/iOS/Android）环境下同样的账户体系和服务。



CodeWeavers

CodeWeaver 是一家专门销售 wine 商业版本的公司，其产品 crossover 可以通过 wine 技术实现 Windows 应用在 linux 上运行。CodeWeaver 公司的产品包括 crossover mac 版本和 Linux 版本，其具体产品包括高级版，标准版，游戏版，和服务器版。crossover 游戏包括诸多本地和网络游戏，同时支持补丁的快速在线更新。在和深度的联合开发过程中，不仅双方提供了针对中国大陆的游戏支持，更新增了针对中国本土软件的支持。尤其对于 QQ 在深度桌面版上的运行是一个前所未有的突破，实现了 CodeWeaver 在中国的落地。



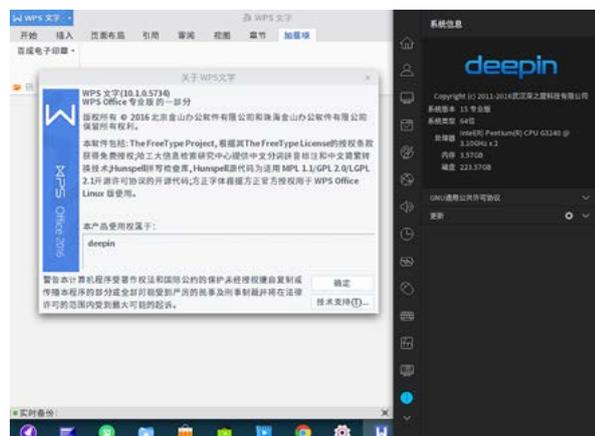
深度科技优化运行了更多常用的 windows 软件，如 Photoshop、Microsoft office 2010 等常用软件，给用户提供了更多选择的同时，在桌面生态建设上奠定了坚实的基础而针对 CodeWeaver 商业版本的使用上，所购买版权费用完全由深度科技支持，面向社区和所有用户都是免费的。

未来，随着合作的深入和技术的成熟，我们会支持更多更新的非 linux 版本软件，以此来解决用户的实际问题。



金山办公版

WPS Office 是由金山软件股份有限公司自主研发的一款办公软件套装，可以实现办公软件最常用的文字、表格、演示等多种功能。具有内存占用低、运行速度快、体积小巧、强大插件平台支持、免费提供海量在线存储空间及文档模板、支持阅读和输出 PDF 文件、全面兼容微软 Office97-2010 格式(doc/docx/xls/xlsx/ppt/pptx 等) 独特优势。



深度科技与金山软件联合发布深度桌面操作系统 V15 金山办公版，用户在购买深度操作系统的同时享受 WPS 的企业版支持，双方在销售渠道中给用户完整提供办公和系统的解决方案。深度桌面操作系统 V15 金山办公版的推出经过完整的测试和优化，不仅方便用户使用，也提供了更为便捷的购买方式。

作为一个用了很久 deepin 系统的中国人，不能不为这么好用



hxy258852 / 2016-9-23 11:41

浏览: 322 / 回复: 22

其实我用 deepin 系统很久了，从去年的四系统 (win7+deepin+centos+ubuntu) 慢慢变成三系统 (win7+deepin+centos)，直到最近，变成了单系统 deepin

用了两年多的 deepin，一直没有注册过账号，也没在 deepin 论坛和各位网友进行过交流，现在，新版本的 deepin 可以说是完美无暇了，我果断把 win7 也扔了，自己新买的电脑只有一个 deepin，推荐大家安装 15.3 版的 deepin !!!

相对于 ubuntu 麒麟来说，deepin 用他八条街不止，就是从全世界来看，deepin15.3 也是位于前三名的 linux 衍生版，中国人的骄傲啊。

现在，我把办公室的电脑也安装成 deepin 系统了，在同事们使用后正在大力向周围办公室推荐。

今天，我实在不忍再这样偷偷的用着咱们国产的系统了。

所以果断注册账号来这里表示下对 deepin 的感谢，deepin 真的是中国 IT 界的骄傲！

当然，最好感谢 deepin 的办法还是把 deepin 推广出去，目前对于办公来说，deepin 已经完全可以取代 windows 了。

感谢 deepin 公司的努力和奉献！！

—————2016.9.23 于阜平县政府办公室



weijuwei 发表于 3 小时前 来自移动端 | 只看该作者

谢谢 deepin 团队的努力！



399913628 发表于 2 小时前 | 只看该作者

谢谢 deepin 团队的努力！

努力加油。。。。。。
我的电脑就只有你了。



hxy258852 发表于 1 小时前 | 只看该作者

不过即便 deepin 要是决定与政府合作，记得要两条腿走路，不能成了官方定制版，哪怕分开做，一面做官方定制，一面坚持做 free world 版，毕竟目光要放长远，有时候只做受人控制的版本会断送走向世界的未来，这个需要权衡，但是，要相信，未来的世界不属于微软也不属于苹果，而是 linux 的未来。这点我想 deepin 团队早已心知肚明。



sevenlinux 发表于 2016-9-24 13:24:24 | 只看该作者

现在 linux 已经不再像我们以前印象中的那么“难搞”了。每个国家都有自己的 linux 发行版，去适应每个国家各自国民的操作习惯。deepin 在这方面确实做的越来越好。从系统安装的简单化，到系统的操作习惯简单化，以及现在日趋丰富的生态圈。将来再加上中国几个比较大的特有的软件商和一些外接输入输出设备的 linux 版，在 deepin 上开发应用。那么在不久的将来，不说一定要从 windows 迁移，从此不再 windows (那样太极端)，最起码让我们每个人都有个自己的选择，其实也是不错的！



johnny123 发表于 2016-9-24 15:13:10 | 只看该作者

顶楼主，我的办公用机也是 Deepin。

对 deepin 第一印象



mrdeng / 2016-11-9 22:18

浏览: 517 / 回复: 28

今天在百度首页看到这个系统,然后下班回家就安装了。给我第一印象相似 MAC OS 系统,但玩了一会又有点惊讶(后来查询了一些来明,在国内竟然还有这个系统存在,虽不是真正的属于国内的也不是一个真正类似 MAC OS/Win 的系统,玩了会我就感觉这是一个真正系统了。个人感觉“流畅度”超过了 Win,页面设计仅次于 MAC OS。可能是我不是很会玩(或玩时间短)还没有遇到 bug,目前感觉唯一的缺陷就是软件方面,photoshop/LOL 等一些 win 系统平时用的玩的软件游戏不能用。对于没钱买 MAC 的来说这是一个很不错的系统体检,但是感觉为什么很少人用,很少人知道的呢?如果今天我不是在百度首页看到了,都不知道还有这个存在,比起那些拿国家资金开发的系统好一些!!!也感谢 deepin 团队,能让我不用在折腾黑苹果就能体验了(纯属小白印象 /// 见解)



fuyao 发表于 2016-11-9 22:55:39 来自移动端 | 只看该作者

看来得感谢百度一下了,默默地打开了百度网盘。。。



mrdeng 发表于 2016-11-10 17:33:45 | 只看该作者

可能有些人用惯了 win 不想尝试其他系统。至于我 如果平时用不出现什么大问题的话,我想我会用这个系统的,如果要工作或者是玩游戏可能要过 win 了!!! (有时候我比较喜欢折腾,哈)



Wintersweet 发表于 2016-11-10 18:32:58 | 只看该作者

一般不墨守成规的人都会很快上手,乐于接受和爱上新事物,新鲜体验!打不破的是心灵的魔咒!对于大多数人来说日常办公丝毫没有影响除了特定行业需要特殊应用工具办公外!



chaojun56 发表于 2016-11-11 14:45:47 | 只看该作者

双十一购物都正常



joyk3000 发表于 2016-11-11 15:16:35 来自移动端 | 只看该作者

喜欢就好

deepin 做得很好了



brucegl / 2016-11-13 21:26

浏览: 491 / 回复: 11

对于一些喜欢折腾的人来说，任何 linux 版本都是不稳定的。作为一个日常使用的操作系统。deepin 做得很好了。应用的整合很全面。桌面优化不输 ubuntu。我是从 debian 过来的。debian 是很干净的系统。干净到什么都是自己来装。现在用 deepin，大部分都装好了。而且应用的体验也很棒。deepin 的开发进度也是很稳定的。我表示没必要折腾 linux 版本了。对于桌面用户 deepin 几乎是优于 ubuntu 的选择。至少应用的整合甩了 ubuntu 几条街。而且本土化的应用生态圈 deepin 会有很大的发展空间。一统国内 linux 桌面端是早晚的事了。



aurthur 发表于 2016-11-13 22:02:59 | 只看该作者

文承武德，一统天下。



ginanony  发表于 2016-11-16 09:14:08 | 只看该作者

一统天下，唯我 deepin



moonlimb 发表于 2016-11-16 09:49:22 | 只看该作者

相比 fedora，ubuntu，mint 等发行版（各有特色），deepin 是个全能选手。



nebhao 发表于 2016-11-17 10:59:15 | 只看该作者

看起来 deepin 很稳定。

Deepin 的盈利模式是什么？



leonardo520 / 2016-10-26 19:58

浏览: 1215 / 回复: 32

deepin 靠什么盈利？以后会不会在系统里植入广告？像 miui 一样？还是像微软一样卖软件？



wangyong 官方管理员 发表于 2016-10-26 21:47:10 来自移动端 | 只看该作者

我们做的是产品，情怀老罗做的最好，我们活的很好，谢谢大家关心



leonardo520 发表于 2016-10-26 21:51:55 来自移动端 | 只看该作者

希望你们把 deepin 做的越来越好，希望最近两年能把系统质量超过 ubuntu



eleveni386 发表于 2016-10-27 00:25:08 | 只看该作者

老实说，大家关心 Deepin 的盈利模式，我觉得更多的是担心，因为企业没盈利，而导致该系统无法很好的继续迭代。而让大家少了一个不错的桌面系统，太可惜了。毕竟开箱即用且符合国人软件特色，也只有 Deepin 做的好了。不管是 Linux 老手，还是新手都能融洽。不过，这里吐槽一下 Deepin 的电池管理，确实感觉挺费电的。通过 powertop 和 ltp 等工具进行优化，之后还是没明显效果。这点在用 Ubuntu 时没有出现过。



ssiaw12345 发表于 2016-10-27 10:10:44 | 只看该作者

支持 deepin，以前一直再用 Ubuntu，对比之下真的觉得 deepin 真是太棒了



flydeepin1979 发表于 2016-10-27 17:32:44 | 只看该作者

你们活得很好。我们就放心了。这是真心话。真的怕你们赚不到钱，然后。。。



sevenlinux 发表于 2016-10-27 21:00:15 | 只看该作者

我还记得今年年中时候，deepin 搞了一个捐正模式，好几个模块项目。我作为个人虽然贡献少的可怜，但也是每个项目都捐了一点。其实，如果每个人在使用的这个系统的话，或多或少会有一些支持。积沙成塔。再加上政府大方向的支持，和一些大企业的支持，deepin 就一定不会是我们所担心的烂尾项目，而是越做越优秀，可能暂时不能和 ubuntu 那样广泛，但是只要是国人心目中最好的是绝对可以的。毕竟，deepin 在国产定制的 linux 中，目前体验绝对最好的（当然仁者见仁，可能有些人无法认同）。

而且每个人的观念都会随着时代的发展而改变。就像以前，我们用客户端或者网页听歌下歌，看电视看电影，都是免费的，那些客户端其实很早以前都在收费，但是真正花钱的还是占少数部分。但是现在，可能每个人都会有一个爱奇艺，或者腾讯等视频会员，或者酷狗，网易云等会员。最起码对于我这种很少看电影电视的人来说都会有一个电视电影会员。这其实也是一种改变，一种对别人给我的日常生活带来便利的观念改变。也是对作者一些尊重吧。

不过话说回来，现在完全进入一个互联网+的时代，一个大数据的时代，如果 deepin 只是靠我们这些人普通人捐款生活，那也是绝对不可能的。他们有他们的生存模式，当他们把系统做的越来越稳定，越来越优秀的时候，作为终端使用 deepin 的我们，做到我们该做的一些支持，也是情理之中的。



brucegl 发表于 2016-11-14 00:23:17 来自移动端 | 只看该作者

你们活的好我们作为长辈就放心了



2016 年 9 月

某专用安全操作系统研究型项目

为了应对现代化国防需要，进而提升我国国防领域信息系统的安全防御能力和作战效能，就必须加快相应国产基础软件的研制。高安全高可靠的专用安全操作系统的研究就成为了当务之急。

从目前现状来看，我国在操作系统定制与裁剪方面已经积累了一定的用户界面的定制能力，主要集中在桌面环境的样式、功能、使用习惯等方面。但是对于系统底层，包括内核参数、内核代码、内核配置、内核模块等的定制与裁剪，仍然基本处于空白。但是这些差距是可以集中攻关的方式逐渐跟进和追赶上国外水平的。

一、从操作系统内核定制与裁剪方面来看，会涉及到以下几点内容：

- (1) 内核编译参数定制
- (2) 驱动程序裁剪
- (3) 内核模块裁剪
- (4) 内核运行参数定制
- (5) 内核安全补丁定制

二、从操作系统用户层定制与裁剪方面来看，

由于用户层的规模远远大于内核层，而且可以为普通用户改动，因此裁剪成本更高，会涉及到以下几点内容：

- (1) 性能的优化
- (2) 安全方面的加固
- (3) 系统可维护性的增强
- (4) 对软件兼容性的加强

三、完整可信链的实施技术

在专用的操作系统中，需要提供对完整可信链的支持，实施从 BIOS 起、经 bootloader、至内核、遍及内核模块与驱动程序、到 init、最后抵达用户进程与文件的可信链。

通过实现以上技术方案，满足最小化、可定制、高可靠、高可信、高安全的目标。从而使得系统更为稳定、高效、安全可靠，并满足可信计算的规范，保障从底至上，由 BIOS、外围硬件、系统载入器、系统内核、内核模块、初始进程以至于各进程、库与文件的系统完整性与可靠性。

2016 年 8 月

某市医院

背景：某市医院始建于 20 世纪 70 年代，是一所集临床医疗、科研教学、预防保健和社区服务为一体的国家二级甲等综合性医院。

众所周知，国家当前推行软件正版化工作是我们应履行的义务，加强软件版权保护，对鼓励软件创新、优化市场环境、实施创新驱动发展战略、加快创新型国家建设都具有重要意义。软件正版化对保护知识产权、保护企业信息安全、保障企业的诚信和规范管理也有重要作用。深度操作系统是深度科技自主研发的 Linux 操作系统，友好、美观、易用，且进入中央国家机关政府采购中心操作系统采购名录，无论在成本方面还是在使用上完全满足企业需求。该市医院选择深度操作系统桌面版软件，有效的解决了国家推行的软件正版化工作，并在终端使用中解决了正版办公软件的问题，保障医院和病患信息的安全。

感想：在当前对信息化安全高度集中、密切关系的工作中，有越来越多的企事业单位正在逐步替换本单位的软硬件，实现信息安全，自主可控，这是我们希望看到的，也希望在未来能够得到国家更多的支持和鼓励。

2016 年 10 月

中国人民银行福州市中心支行软件 正版化

背景：在金融十二五行业规划过程中，中国人民银行福州市中心支行已率先实现部分国产软硬件替代，在使用深度操作系统桌面版软件办公的过程中，员工能够很快的适应新的办公界面，而且深度桌面软件与现有硬件兼容良好，能够满足中国人民银行福州市中心支行的基本业务需求，为了更快更好提高中国人民银行某市中心支行信息安全防范能力，中国人民银行福州市中心支行再次与深度科技合作，携手打造金融行业中国产化软硬件替换实现的自主可控领先地位。

体会：由于有了初次合作的基础，本次安装、部署进展顺利，客户已正常使用。





2016年 10月

2015年广东省欠发达地区电子政务云平台项目

项目地点：广东省潮州市、韶关市、肇庆市、湛江市、茂名市

项目背景：为促进广东省欠发达地区电子政务建设的发展，贯彻落实《国家电子政务总体框架》、《广东省信息化发展规划纲要（2013-2020年）》、《广东省电子政务发展规划（2014-2020年）》等有关部署，加快推进各地“互联互通”的工作进度，提高信息化建设的服务水平，经过三年分批次支持我省13个欠发达地市5个专题方向的建设，使欠发达地区的政务信息化能力得到显著提升，有效支撑省网上办事大厅、社会信用信息系统、市场监管平台等重大信息化项目的建设 and 应用。一是支持省网上办事大厅地市分厅主要系统建设开发，提升地市分厅与省主厅的互联互通水平；二是支持地市政务信息资源共享平台建设，实现地市各部门的横向连通、地市与省级的纵向贯通；三是支持地市电子政务云平台建设，支撑地市重要电子政务公共基础设施和公共电子政务应用系统；四是支持地市电子政务外网建设，提高网络承载各项政务应用的能力；五是支持地市构建电子政务网络信息安全保障体系，提供安全解决方案及检测预警服务。

按照分期分批、重点优先的原则，根据各地的建设基础、发展情况和配套措施等条件，组织召开了专家评审会，经过论证，2015年将支持潮州市、韶关市、肇庆市、湛江市、茂名市。资金分配方式由省信息中心按照政府采购程序，统一采购、统一建设、统一布置、统一实施。

项目目标：对5个欠发达地市（潮州市、韶关市、肇庆市、湛江市、茂名市）建设电子政务云平台，为其开展重要政务应用提供电子政务云方向的建设基础设施支持及配套支撑管理工作。

建设内容：针对本项目建设需求，采购云计算相关产品和服务，支持5个地市电子政务云平台建设，支撑地市的网上办事大厅、政务信息资源共享平台等重要电子政务公共基础设施和公共电子政务应用系统。

项目建设原则：本项目设备产品原则上建议选用国产并具备自主知识产权的设备（产品软件著作权人须与公司名称一致），只有在无相应国产设备时方可选用经国家主管部门批准的国外设备

项目解决方案：在本项目中采购的25套Linux服务器操作系统全部为深度操作系统服务器版软件V15产品，深度操作系统满足工信部的国产操作系统标准化评测，全面支持包括东方通、中创、金蝶、普元等国产中间件，支持达梦、人大金仓、神州通用、虚谷等国产数据库。深度操作系统在全面满足操作系统平台软件自主可控的基础上，全面支持国产电子政务信息系统的安全可靠建设。

深度工程团队可以提供专业的运维技术支持服务，提供诸如电话、远程、现场等多种服务方式，并可帮助用户定制服务器运维规范，编写应用系统的安装和部署文档等。并在服务器出现故障时，从操作系统层面协助用户分析问题，并给出专业的分析结果。

2016年 10月

武汉市统计局

项目背景介绍:

在当前国家高度重视信息安全和国产化工作推进进展的背景下，武汉市政府及各事业单位积极响应、推动并落实国产化进程，在此过程中，针对硬件和软件的甄选均以国产产品为首选。

武汉市统计局作为武汉市数据上传和交互的重要枢纽，目前正逐步替换现有软硬件产品，向国产化迁移。而操作系统作为支撑软硬件高效运转的基础软件平台，在信息化安全建设中承载着至关重要的作用。

项目建设意义:

1. 将原有应用系统迁移到国产操作系统上运行，从系统底层达到自主安全可控的目的。

2. 有利于用户对资源调配进行统一管理。

项目解决方案:

最初用户选择浪潮服务器和虚拟化软件搭建云平台，深度操作系统服务版软件部署在浪潮云软件上，构建一个全国产化的应用云平台。前期部署过程中，深度操作系统与浪潮虚拟化软件存在些许兼容性问题，经过深度工程师的耐心排查，最终成功解决并部署安装。后期用户根据自身的应用需求，将云平台更换成了VMware虚拟化产品，我司再次安排工程师去现场调配，最终安装部署顺利完成，达到了用户的期许和使用的稳定性。

应用效果:

1. 客户在使用深度服务器操作系统后，提高了系统的安全性和稳定性，便于用户操作和管理。

2. 我司根据客户应用需求，积极响应，安排工程师现场安装部署，得到了用户认可。

3. 我司通过对不同平台虚拟化产品的部署，也为后期产品的适配奠定了一定的基础。





紧急出动——Oracle11g Rac 的挑战

● 北京 工程部 / 文

七月中旬，某客户提出需要在服务器上安装 Oracle11g Rac。我们的服务器操作系统在之前进行过 Oracle11gRAC 的适配，没有出现问题，也有安装 RAC 的技术文档。虽然客户自己的技术工程师是 Oracle 的 OCM 出身，但由于客户应用的一些软件更改了操作系统的默认软件包和环境变量，导致 Oracle 11g Rac 安装过程会和这些包产生冲突，仅按照文档操作无法完成安装，因此紧急要求我们提供技术支持。

为了尽快满足用户上线要求，部门决定派工程师去客户现场进行适配。截止至 7 月末，成功完成了客户需求，我有幸参与了该项目实施的全过程，现在在这里给大家分享一下其中的历程以及成果。

说到 Oracle Rac 的安装，首先需要解释一下 Rac 的工作原理：在一个应用环境当中，所有的服务器使用和管理同一个数据库，目的是为了分散每一台服务器的工作量，硬件上至少需要两台以上的服务器，而且还需要一个共享存储设备。同时还还需要两类软件，一个是集群软件，另外一个就是 Oracle 数据库中的 RAC 组件。同时所有服务器上的 OS 都应该是同一类 OS，根据负载均衡的配置策略，当一个客户端发送请求到某一台服务的 listener 后，这台服务器根据我们的负载均衡策略，会把请求发送给本机的 RAC 组件处理也可能发送给另外一台服务器的 RAC 组件处理，处理完请求后，RAC 会通过集群软件来访问我们的共享存储设备。配置完 Rac 之后可以使数据库达到的“高可用，易伸缩，

低成本，高吞吐等”目的，同时，基于两台同构计算机的相互监控，可以达到负载均衡的目的。

为了完成技术支持，首先，需要制定该项目的执行方案：

第一、与客户方面进行沟通，了解客户需求，询问需要安装的 Oracle 版本号，远程存储的类型，已经安装的软件和版本；派遣工程师去现场查看服务器操作环境，准备回公司复现操作环境。

第二、在公司复现客户方面的操作环境，测试安装遇到的问题，工程部和研发部的兄弟共同探讨解决办法。

第三、测试成功后，进行现场实施。

项目开始实施后，的确是出现了新的技术难点；粗略的概括为以下几点：一、软件的安装环境；二、Oracle 的安装包；三、客户预装的软件影响安装。

在部署安装环境的时候，由于客户使用了第三

Oracle Linux 6 and Red Hat Enterprise Linux 6

The following packages (or later versions) must be installed:

```
binutils-2.20.51.0.2-5.11.el6 (x86_64)
compat-libcap1-1.10-1 (x86_64)
compat-libstdc++-33-3.2.3-69.el6 (x86_64)
compat-libstdc++-33-3.2.3-69.el6.i686
gcc-4.4.4-13.el6 (x86_64)
gcc-c++-4.4.4-13.el6 (x86_64)
glibc-2.12-1.7.el6 (i686)
glibc-2.12-1.7.el6 (x86_64)
glibc-devel-2.12-1.7.el6 (x86_64)
glibc-devel-2.12-1.7.el6.i686
ksh
libgcc-4.4.4-13.el6 (i686)
libgcc-4.4.4-13.el6 (x86_64)
libstdc++-4.4.4-13.el6 (x86_64)
libstdc++-4.4.4-13.el6.i686
libstdc++-devel-4.4.4-13.el6 (x86_64)
libstdc++-devel-4.4.4-13.el6.i686
libaio-0.3.107-10.el6 (x86_64)
libaio-0.3.107-10.el6.i686
libaio-devel-0.3.107-10.el6 (x86_64)
libaio-devel-0.3.107-10.el6.i686
make-3.81-19.el6
sysstat-9.0.4-11.el6 (x86_64)
```

方软件，为了支持这些软件的运行，导致很多软件包不是操作系统的默认版本，而且该软件版本高于 Oracle11gRac 识别的最高软件版本；我们只能根据客户的操作系统环境进行软件包的降级或者工作。

从软件源来说：Oracle 官方为了其商业用途，发布的版本都是源码包或者针对特定 RHEL 系列的操作系统发布的 rpm 包，这就使得我们需要更改源码包进行安装，工作量不小。

```
11.2.0.1.0 版
即时客户端程序包 — Basic: 运行 OCI、OCICLI 和 JDBC-OCI 应用程序所需的所有文件
  □ instantclient-basic-linux32-11.2.0.1.zip (46,738,674 字节) (cksum -203368094)
  □ oracle-instantclient11.2-basic-11.2.0.1.0-1.i386.rpm (46,299,404 字节) (cksum - 2003986695)
```

而 Oracle 的源码包默认的安装路径以及调用的环境变量都是根据 Oracle 的 OEL 操作系统来设计的，相比 Oracle 服务器 OEL 系统而言，不同操作系统上很多配置文件以及环境变量的存放位置都不一样，会产生写入错误或调用错误。

发现了所有问题之后，就要开始思考解决办法，安装 Oracle 的 OEL 操作系统，在操作系统上运行 Oracle 的安装程序，查看安装过程中的需要的依赖包，以及调用环境变量、写入配置文件等信息，研发的同事们各司其职，只用了两天的时间就成功的完全分解了 OracleRac 的源码安装过程。

分解完成后，首先从软件包入手，根据 Oracle 安装时需要的软件包版本，查看客户方面的操作系统中安装的软件版本，对比差异，找出有差异的包，使用合适版本的系统包进行替换。

依赖包调整结束之后，就开始了 OracleRac 安装的重头戏 – 更改配置文件，配置文件大致分为两部分，用户环境变量的配置文件，系统配置文件。Oracle 服务器这方面提供的公开资料有限，需要进行单步跟踪对比两边差异，进行修改。

由于需要赶上用户的时间进度，参与这个过程



的几位兄弟废寝忘食的工作，仿佛已经没有了时间的概念。基本都是两三天没有回家，就是为了早日完工，不把时间扔到路上。加班的这段时间里，大家都是竭尽全力的工作，困了就冲杯咖啡继续干，实在撑不住了就躺在沙发上睡一会儿，睡前还不忘和其他同事说一声：有问题马上叫我！大家还没回应，轻微的鼾声就已经响起了。

大家的努力是有回报的，从最开始的不能正常安装，到安装读条到达 100% 的时候，所有人有松了一口气，测试没有问题之后，安排进行现场实施，问题完美解决。

在适配过程中，深度的研发团队以及工程团队展现出了雄厚的技术实力 – 强大分析能力，对系统各层面软件和配置的熟悉；展现出了勇攀高峰，不畏险阻的研究精神，这就是深度的精神！



龙芯优化的战斗小结

● 北京 工程部 / 文

在某大型保密项目入围测试中，深度科技与国内某航天院所共同组成了深度龙芯版桌面操作系统的测试组合，并且在其中为了满足测试需求，我们基于龙芯平台移植了研发团队长达半年多精心研制的文件管理器。

今年九月份基于某大型保密项目上根据用户提出的需求，我们需要优化文件管理器拷贝性能，因为根据某大型保密测试大纲的需求来看要对 10G 混合文件进行拷贝测试，来满足评分的要求。大家可以看下，10G 文件里面主要包含了文件夹 6 层，21280 个文件，（400mb 文件 1 个，20MB 文件 4 个，2Mb 文件 6 个，1Mb 文件 3 个，20Kb 文件 20 个，4Kb 文件 1000 个）×20，从这数据里面可以看到其中不仅仅涉及到大文件，里面还有小文件以及目录层次的要求。经过测试我们的 10G 混合文件拷贝经过 6 层目录拷贝之后我们的时间是 450 秒，在与用户的沟通过程中我们了解到必须把拷贝速度优化到 90 秒左右。根据我们了解到的信息，在硬件平台相同的情况下其他同类操作系统的拷贝速度在 480 秒左右，顿时我心里就在想这有点不现实啊，怎么完成呢？450 秒到 90 秒足足 5 倍的差距，而且在竞争对手也处于 480 秒拷贝时间的情况下，怎么达到 90 秒啊。

刚开始我一直在想拷贝时间提升 5 倍，从

450 秒提升到 90 秒是一件不太可能完成的事情，后来回到公司跟研发以及上级领导经过多方面的探讨之后发现，如果想提升拷贝性能那我们只能从硬件以及软件这两个方面来做出相对应的调整。在硬件方面我们只能基于硬盘方面来做出调整，由于之前我们测试出来的数据全部是基于机械硬盘，所以我们在想是不是可以通过更换固态硬盘来达到优化效果。软件方面我们只能从系统软件、系统性能、以及内核参数等方面来进行性能的优化。经过多方面讨论之后我们决定从这几个方面下手，尝试是否可以进行大幅度的优化来达到 5 倍的提升速度。为了测试我们商量出来的方案是否有效，我跟研发部门曹大神决定去现场对商讨出来的这两种方案进行优化测试，看是否可以达到我们最终想要的的数据。

基于硬件方面，经过多方面测试之后发现机械硬盘的读取速度最大速度在 160MB，固态硬盘的读取速度可以达到 260MB。也就是说从理论上讲机械硬盘的拷贝时间可以达到 125 秒左右，而固态硬盘的理论数据拷贝时间可以达到 77 秒左右。由于 cpu、文件大小等多方面的影响 125 秒以及 77 秒这些理论数据在现实中并不能达到，这就需要我们去测试硬盘在实际情况下拷贝的极限速度来进行优化。

经过多重测试之后，发现固态硬盘的拷贝时间

```
Terminal 终端 - deepin@deepin-pc: ~/桌面/组合测试样本
文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)
./10G/4KB-981.zip -> ./testcontent/4KB-981.zip
./10G/4KB-982.zip -> ./testcontent/4KB-982.zip
./10G/4KB-983.zip -> ./testcontent/4KB-983.zip
./10G/4KB-984.zip -> ./testcontent/4KB-984.zip
./10G/4KB-985.zip -> ./testcontent/4KB-985.zip
./10G/4KB-986.zip -> ./testcontent/4KB-986.zip
./10G/4KB-987.zip -> ./testcontent/4KB-987.zip
./10G/4KB-988.zip -> ./testcontent/4KB-988.zip
./10G/4KB-989.zip -> ./testcontent/4KB-989.zip
./10G/4KB-98.zip -> ./testcontent/4KB-98.zip
./10G/4KB-990.zip -> ./testcontent/4KB-990.zip
./10G/4KB-991.zip -> ./testcontent/4KB-991.zip
./10G/4KB-992.zip -> ./testcontent/4KB-992.zip
./10G/4KB-993.zip -> ./testcontent/4KB-993.zip
./10G/4KB-994.zip -> ./testcontent/4KB-994.zip
./10G/4KB-995.zip -> ./testcontent/4KB-995.zip
./10G/4KB-996.zip -> ./testcontent/4KB-996.zip
./10G/4KB-997.zip -> ./testcontent/4KB-997.zip
./10G/4KB-998.zip -> ./testcontent/4KB-998.zip
./10G/4KB-999.zip -> ./testcontent/4KB-999.zip
./10G/4KB-99.zip -> ./testcontent/4KB-99.zip
./10G/4KB-9.zip -> ./testcontent/4KB-9.zip
207.113252
deepin@deepin-pc:~/桌面/组合测试样本 $
```

```
Terminal 终端 - deepin@deepin-pc: ~/桌面/组合测试样本
文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)
./10G/4KB-982.zip -> ./testcontent/4KB-982.zip
./10G/4KB-983.zip -> ./testcontent/4KB-983.zip
./10G/4KB-984.zip -> ./testcontent/4KB-984.zip
./10G/4KB-985.zip -> ./testcontent/4KB-985.zip
./10G/4KB-986.zip -> ./testcontent/4KB-986.zip
./10G/4KB-987.zip -> ./testcontent/4KB-987.zip
./10G/4KB-988.zip -> ./testcontent/4KB-988.zip
./10G/4KB-989.zip -> ./testcontent/4KB-989.zip
./10G/4KB-98.zip -> ./testcontent/4KB-98.zip
./10G/4KB-990.zip -> ./testcontent/4KB-990.zip
./10G/4KB-991.zip -> ./testcontent/4KB-991.zip
./10G/4KB-992.zip -> ./testcontent/4KB-992.zip
./10G/4KB-993.zip -> ./testcontent/4KB-993.zip
./10G/4KB-994.zip -> ./testcontent/4KB-994.zip
./10G/4KB-995.zip -> ./testcontent/4KB-995.zip
./10G/4KB-996.zip -> ./testcontent/4KB-996.zip
./10G/4KB-997.zip -> ./testcontent/4KB-997.zip
./10G/4KB-998.zip -> ./testcontent/4KB-998.zip
./10G/4KB-999.zip -> ./testcontent/4KB-999.zip
./10G/4KB-99.zip -> ./testcontent/4KB-99.zip
./10G/4KB-9.zip -> ./testcontent/4KB-9.zip
88.508044
deepin@deepin-pc:~/桌面/组合测试样本 $
```

可以达到 330 秒左右的时间，但是远远没有达到我们理想的数据，这也就意味着我们还可以在这个方面跨出一大步。但是不同的固态硬盘性能也有差异，选择合适的硬盘型号是能够继续提高。经我们在拷贝速度上面有了很大的进步，但是我想到了必须达到 90 秒的时间之后，我又陷入了深深的忧伤之中。

基于软件方面，经过公司内部讨论之后，让我协助曹大神到现场进行测试，前期主要是基于曹大神写了一个可执行 demo 来进行测试 10G 混合文本拷贝，通过代码的方式来实现对 IO 缓存和调度的优化，经过多天的测试以及曹大神根据测试结果频繁的修改 demo 代码来测试系统的 10G 混合文件拷贝速度发现拷贝速度得到了几十秒的提升。

在公司后援北京研发部负责人基于内核参数以

及文件系统两个方面分别进行了优化。经过五天左右的测试之后，拷贝性能已经完全可以满足用户的需求，对于内核虚存进行了多方面的优化之后拷贝数据已经达到了 88 秒的时间。

经过长达十天的时间里我跟曹大神经过了从硬件到软件方面多方面、多层次的调试之后，目前深度龙芯版桌面操作系统已经达到了 88 秒的拷贝时间，并且领先竞争对手将近 30 秒的拷贝时间。看上图就可以看到这么长时间的努力终究是有回报的。

虽然最终测试这个项目没有列入评分标准。但是，在这次测试准备工作中，我们已经向合作伙伴证明，深度的技术能力是能够做到最好的。d



一些冷门的 C 语言知识

语法篇 (C99)

● 北京 研发部

1. C 语言的 declaration

○ 估计没多少人知道 typedef 是可以写在类型中间的，比如

```
int typedef unsigned uint32_t;
```

在 C99 的语法里，typedef 在语法范畴上是属于 storage class specifier。如果你去看它的产生式你会发现其实它写在类型的任何地方都可以。同属于 storage class specifier 的还有 extern static auto register。比如：

```
int static adder(int, int);
ssize_t extern write(int fd, const void *buf, size_t nbyte);
```

但是这个并没有什么用。。。。

○ 关于 typedef 引起的歧义

如果我定义

```
1 typedef unsigned int * handle_t;
2 handle_t h1 = 0;
3 const unsigned int * h2 = h1;
4 const handle_t h3 = h2;
```

大家都知道 h1 的类型是指向 unsigned int 的指针。问题是 h2 和 h3 的类型是不是一样的呢？答案是不一样，不信可以拿编译器去跑一下。h3 的类型其实是 unsigned int * const，也就是一个指向无符号整形的常量指针。为什么会是这样？简单的说，typedef 定义使得 unsigned int* 成为一个独立的类型。const 是一个 type qualifier，用来修饰一个类型的。在 h3 的例子下，就是修饰 handle_t 这个类型的，而不是修饰 unsigned int 的。h2 里类型是 unsigned int，所以 const 修饰了 unsigned int。关于 Pointer，它通常属于 declarator 的组成部分（参考下面关于 const 的说明），但是这里由于 typedef

被包含在了新类型 handle_t 里。typedef 不是也不同于 #define，不是做字面量替换。

○ 关于 const

很多人觉得 C 的变量声明语法混乱，可读性差。于是有人发明了左右读的方法。如果你仔细的去读 C 的语法声明规范，就不会感觉那么难了。其实大致上去分的话，C 的声明分两部分，由类型限定 (declaration specifiers) + 声明符 (declarators) 组成。const 是 type qualifier，它可以出现在 declaration specifier 里，也可以出现在 declarator 里。出现在 declaration specifier 里是用来限定类型的，比如 int const 或者 const int（像之前说的，storage class specifier 可以在类型的任何地方出现一样，type qualifier 也可以放在在类型之前或之后）代表只读的整数。const 放在 declarator 里是用来修饰指针的。* 属于 declarator 的一部分，所以只要 declarator 里有 * 就代表是一个指针，比如 * const p1 表示一个名称为 p1 的只读指针。所以你会发现有时候 const 修饰它左边的元素（在 * const p1 里），有时候修饰右边的元素（int const）。如果你按照我上面的说明，你就能明白为什么了，并不需要死记。最后再来个例子。

```
unsigned const char const * const * const ppuc
```

我们可以将声明一分为二，unsigned const char const 和 * const * const ppuc。前半部分代表只读的无符号字符类型（其实编译器会警告有第二个 const 是重复的）。后半部分说明 ppuc 是一个指向只读指针的只读指针。连起来就是 ppuc 是一个只

读指针，指向一个指向只读的无符号字符类型的只读指针。

2. 从 duff's device 说 switch

第一次看到这个结构的时候很吃惊，感觉像是完全重新认识了 C 语言。

```
1 int bar(int n) {
2     int ret = 0;
3     int r = (n+3) / 4;
4     switch (n % 4) {
5     case 0: do { ret++;
6     case 3: ret++;
7     case 2: ret++;
8     case 1: ret++;
9     } while (--r > 0);
10    }
11    return ret;
12 }
```

后来看了 C99 的规范后，发现其实 C 还有很多语法上的潜力可以挖掘。只是平常不会有人闲的没事去看一个语言的规范。如果学会从语法结构上去看，这个就并没有什么特别的地方。要很容易的看出这种貌似奇怪的代码结构，最核心的一点就是要知道 case 与 switch 并没有必然的关系。唯一的限定就是 case 语句只能出现在 switch 语句的 statement 里。case 的语法结构是

case constant-expression : statement, switch 的语法结构是 switch (expression) statement 。

所以其实任何语句 (statement) 都可以作为 switch 的作用范围。最常见的就是复合语句块 (compound statement), 也就是 {} 里面的语句序列。而 case 语句的唯一作用其实就是作为类似 goto 语句的目标而存在。switch 根据 expression 的求值来决定跳转到与 constant-expression 求值相匹配的 case 目标处。所以下面这个也是合理合法的。

```
1 switch (i)
2     case 2: puts("hello");
```

```
1 switch(key)
2     for (int i = 0; i < 10; ++i) {
3         key = i;
4         case 2: printf("case 2: key is %d\n", key);
5         case 4: printf("case 4: key is %d\n", key);
6     }
```

另外，注意上面 case 语句的语法中，最后是 statement，也即是说 case 后面跟只能跟一个语句。那么下面这个我们常常看到的明显正确的代码是什么情况呢？

```
1 switch (i) {
2     case 1:
3     case 2:
4     case 3:
5         puts("small");
6         break;
7     case 4:
8     case 5:
9     case 6:
10        puts("middle");
11    case 7:
12    case 8:
13    case 9:
14        puts("big");
15 }
```

一般教 C 语言的书籍会说这叫做 passthrough，其实从语法角度讲，用缩进来描述它的正确的结构是这样的：

```
1 switch (i) {
2     case 1:
3     case 2:
4         case 3:
5             puts("small");
6         break;
7     case 4:
8         case 5:
9             case 6:
10                puts("middle");
11    case 7:
12        case 8:
13            case 9:
14                puts("big");
15 }
```

看明白了吗，为什么是这样的？因为 case 后面只能跟一个语句。break 语句其实就是一个目标指向紧跟 switch 作用范围之后的位置的跳转语句。所有 switch 里的语句顺序执行，所谓的 passthrough 能起作用的原因无非就是一直执行，直到遇到了最近的 break 语句而已。

如果我列出编译器编译后的 AST 结构，这个事实就很明显了：

```
SwitchStmt 0x7fbfe2062610 <line:6:5, line:20:5>
  |-ImplicitCastExpr 0x7fbfe20625f8 <line:6:13> 'int'
  <LValueToRValue>
    |`-DeclRefExpr 0x7fbfe20625d0 <col:13> 'int' lvalue Var
```



```

0x7fbfe2062538 'i' 'int'
  `--CompoundStmt 0x7fbfe2062cb0 <col:16, line:20:5>
    |--CaseStmt 0x7fbfe2062658 <line:7:9, line:10:33>
      |--IntegerLiteral 0x7fbfe2062638 <line:7:14> 'int' 1
      | `--CaseStmt 0x7fbfe20626b0 <line:8:13,
line:10:33>
        |--IntegerLiteral 0x7fbfe2062690 <line:8:18> 'int' 2
        | `--CaseStmt 0x7fbfe2062708 <line:9:17,
line:10:33>
          |--IntegerLiteral 0x7fbfe20626e8 <line:9:22> 'int'
          | 3
          | `--CallExpr 0x7fbfe2062820 <line:10:21, col:33>
          | 'int'
          |   |--DeclRefExpr 0x7fbfe2062740 <col:21> 'int
(const char *)' Function 0x7fbfe201c550 'puts' 'int (const char
*)'
          |   | `--StringLiteral 0x7fbfe20627a8 <col:26> 'char
[6]' lvalue "small"
          |   |   |--BreakStmt 0x7fbfe2062880 <line:11:9>
          |   |   |--CaseStmt 0x7fbfe20628a8 <line:12:9, line:15:34>
          |   |   |--IntegerLiteral 0x7fbfe2062888 <line:12:14> 'int' 4
          |   |   | `--CaseStmt 0x7fbfe2062900 <line:13:13,
line:15:34>
          |   |   |   |--IntegerLiteral 0x7fbfe20628e0 <line:13:18> 'int'
          |   |   |   | 5
          |   |   |   | `--CaseStmt 0x7fbfe2062958 <line:14:17,
line:15:34>
          |   |   |   |   |--IntegerLiteral 0x7fbfe2062938 <line:14:22>
          |   |   |   |   | 'int' 6
          |   |   |   |   | `--CallExpr 0x7fbfe2062a40 <line:15:21, col:34>
          |   |   |   |   | 'int'
          |   |   |   |   |   |--DeclRefExpr 0x7fbfe2062990 <col:21> 'int
(const char *)' Function 0x7fbfe201c550 'puts' 'int (const char
*)'
          |   |   |   |   |   | `--StringLiteral 0x7fbfe20629f8 <col:26> 'char
[7]' lvalue "middle"
          |   |   |   |   |   |   |--CaseStmt 0x7fbfe2062ac0 <line:16:9, line:19:31>
          |   |   |   |   |   |   | |--IntegerLiteral 0x7fbfe2062aa0 <line:16:14> 'int' 7
          |   |   |   |   |   |   | `--CaseStmt 0x7fbfe2062b18 <line:17:13,
line:19:31>
          |   |   |   |   |   |   |   |--IntegerLiteral 0x7fbfe2062af8
          |   |   |   |   |   |   |   | <line:17:18> 'int' 8
          |   |   |   |   |   |   |   | `--CaseStmt 0x7fbfe2062b70
          |   |   |   |   |   |   |   | <line:18:17, line:19:31>
          |   |   |   |   |   |   |   |   |--IntegerLiteral 0x7fbfe2062b50
          |   |   |   |   |   |   |   |   | <line:18:22> 'int' 9
          |   |   |   |   |   |   |   |   | `--CallExpr 0x7fbfe2062c50 <line:19:21,
col:31> 'int'
          |   |   |   |   |   |   |   |   |   |--ImplicitCastExpr
          |   |   |   |   |   |   |   |   |   | 0x7fbfe2062c38 <col:21> 'int (*) (const char *)'
          |   |   |   |   |   |   |   |   |   | <FunctionToPointerDecay>
          |   |   |   |   |   |   |   |   |   |   |--DeclRefExpr
          |   |   |   |   |   |   |   |   |   |   | 0x7fbfe2062ba8 <col:21> 'int (const char *)' Function
          |   |   |   |   |   |   |   |   |   |   | 0x7fbfe201c550 'puts' 'int (const char *)'
          |   |   |   |   |   |   |   |   |   |   |   |--ImplicitCastExpr
          |   |   |   |   |   |   |   |   |   |   |   | 0x7fbfe2062c98 <col:26> 'const char *' <BitCast>
          |   |   |   |   |   |   |   |   |   |   |   |   |--ImplicitCastExpr
          |   |   |   |   |   |   |   |   |   |   |   |   | 0x7fbfe2062c80 <col:26> 'char *' <ArrayToPointerDecay>
          |   |   |   |   |   |   |   |   |   |   |   |   |   |--StringLiteral
          |   |   |   |   |   |   |   |   |   |   |   |   |   | 0x7fbfe2062c08 <col:26> 'char [4]' lvalue "big"

```

3. C 语言标识符的名字空间

所有的 C 程序员都理解这个语句的作用：

```
typedef struct Tree Tree.
```

但是不是每个人都问过为什么 Tree 这个标识符在这里可以使用两次，不会有歧义吗？第一个 Tree 其实前置声明了一个名字为 Tree 的结构体，而第二个 Tree 是定义了一个自定义类型的名称。即使它们出现在同一个作用域范围内也没问题。下面这个杜撰的变态例子是完全合法可编译的：

```

1 int node()
2 {
3     struct node {
4         int node;
5     } node;
6
7     goto node;
8
9 node:
10    return node.node;
11 }

```

其实 C 语言的标识符是有独立的名字空间的，同一个作用域里不同名字空间里的名称可以相同。具体来说，C 语言有以下几个名字空间：

- 标签（label）一般作为 goto 语句的目标。
- 结构体，联合，枚举类型的名称（C 里叫 tag）
- 结构体，联合的成员拥有独立的名字空间。
- 所有其他标识符的名字（ordinary identifiers）

4. 关于初始化表达式

C99 对于数组和记录（结构体或联合）支持下面这样的选择性初始化，其他为指定的部分会被编译器设置默认该类型的 0 值。

```

1 int a[10] = {
2     [2] = 5,
3     [4] = 6
4 };
5
6 struct node {
7     int val;
8     struct node *left, *right;
9 } n = {
10    .val = 2,
11 };

```

5. 关于嵌套结构体声明

先上代码：

```
1 {
2     struct Event {
3         int type;
4         struct MouseEvent {
5             int x, y;
6         } me;
7         struct KeyEvent {
8             unsigned key;
9         } ke;
10 };
11
12 struct MouseEvent me = {10, 20};
13 printf("%d, %d\n", me.x, me.y);
14 }
```

上面这个看起来诡异的代码片段其实是完全合法可编译的。事实上编译器会将嵌套的记录类型（结构或联合）声明进行提升，让他们处在同一作用域范围内。注意，不要拿 C++ 编译器去编译，我上面所有的例子都是指 C，C 和 C++ 是完全两个不同的语言，尽管 C++ 兼容很多 C 的特性。

6. sizeof 到底是什么

发现很多人纠结 sizeof 死关键字还是操作符，后面需不需要加括号，甚至还经常出现在面试题。从 C 的语法规范去看，sizeof 就是一个关键字，也是一个操作符。作为关键字，说明你不可以定义同名的标识符，不能改变其含义（别跟我说宏可以。。）。作为操作符，它代表一个语句。sizeof 语句后面可以跟两类元素，一种是一个表达式，一个是类型表达式。

```
1 sizeof unary-expression
2 sizeof ( type-name )
```

这就是为什么有时候需要括号，有时候不需要括号的原因。另外，一般来说，如果 sizeof 后面跟随的是 unary-expression，一般是不会做 evaluation 的，只要能静态计算出大小即可，这在 C99 之前都是成立的，一般 C 语言的书上都这么写的。但是由于 C99 支持可变长度数组（VLA, variable length array），如果表达式中包含 VLA，那么就需要进行运行时的求值。

语义篇

1. implicit type conversion

C 语言的隐式类型转换规则还是比较复杂的，会发生在表达式求值、函数参数求值（当函数没有原型或者是可变参类型）等地方。我通过一个例子简单描述下通常所说的 usual arithmetic conversion 过程。我们的问题是，下面第四行代码有多少次类型转换？

```
1 char c = 'b';
2 short si = 5;
3 long l = 10L;
4 unsigned short usi = si - c * 2 + l / 2;
```

我们按可能的求值顺序一个个来说：

首先，si 会提升为 int，然后 c 会提升为 int，`si - c * 2` 的结果是一个 int，然后 `l / 2` 的常量 2 会提升为 long，计算结果也为 long。然后 `si - c * 2` 的计算结果被提升为 long，参与 + 的运算，最后因为赋值表达式的左侧是 unsigned short 类型，所以 `si - c * 2 + l / 2` 被转换为 unsigned short。用代码来描述大致这样的：

```
1 unsigned short usi = (unsigned short)((long)((int)si - (int)c *
2) + l / (long)2);
```

有时候你可能会觉得有些转换是无意义的，比如：

```
1 char c1 = 'A', char c2 = 2;
2 char c3 = c1 + c2;
```

`c1+c2` 运算中有没有类型转换呢，答案是有，一共三次，用代码描述就是：

```
1 char c3 = (char)((int)c1 + (int)c2);
```



为什么这样？因为两个 char 类型相加的结果可能超出一个 char 的值域，所以提升为 int 计算后再进行类型截断才能保证结果的正确。不理解类型转换的规则，就可能犯错误，比如：

```
1 #include<stdio.h>
2 #define TOTAL_ELEMENTS (sizeof(array) / sizeof(array[0]))
3 int array[] = {9, 7, 1, 6, 2, 4, 5, 8};
4
5 int main() {
6     for(int d=-1;d <= (TOTAL_ELEMENTS-2);d++)
7         printf("%d\n",array[d+1]);
8     return 0;
9 }
```

这个程序不会输出任何东西，因为 sizeof 的结果是一个 unsigned int 类型，d 会被转换为 unsigned int 进行比较，因此循环体永远不会被执行。这里还有一个有趣的例子，说明类型转换无处不在：

```
1 main(){
2     char a = 0;
3     short int b = 0;
4     return sizeof(b) == sizeof(a+b);
5 }
```

这段代码的返回值是 1 还是 0 呢？

要想具体了解 C 语言的类型转换规则，最好去看参考手册。

2. restrict keyword

指针别名 (pointer aliasing) 对于优化是很重要的。尤其是在 C 这种显示支持指针并且不同类型之间几乎可以随意强转的语言中更重要。比如我们常用的 memcpy 函数，其函数原型是 `void *memcpy(void *dest, const void *src, size_t n);`，man 手册里明确说明了 dest 和 src 不能重叠。这个事实在 c99 里可以通过 restrict 来表达，即我们可以把 memcpy 声明为 `void *memcpy(void *restrict dest, const void *restrict src, size_t n);`。其实这就

是 memcpy 在 c99 标准里的声明。restrict 是码农对函数的使用者以及编译器的一种提示，编译器可以通过合适的指针分析来优化。比如上面 c99 的 memcpy 函数声明，这就表明编译器可以假定 dest 和 src 肯定不会相等或指向的内存区域重叠，这样 memcpy 实现时就不需要进行类似的安全检查，编译器也可以很安全的减少数据被加载的次数。当然，这并不能限制码农传递同一个指针给 dest 和 src，但是这个行为是 undefined behaviour。要全面了解 restrict 的语义以及对优化的影响，可以看这里 (<http://cellperformance.beyond3d.com/articles/2006/05/demystifying-the-restrict-keyword.html>)

3. block 中的 external

下面这个程序如何修改才能输出全局变量 x 的值 (10)。

```
1 int x = 10;
2 int main() {
3     int x = 20;
4     {
5         printf("%d\n", x);
6     }
7     return 0;
8 }
```

意思其实就是如何才能在这种情况下引用全局变量的 x，答案很简单：

```
1 int x = 10;
2 int main() {
3     int x = 20;
4     {
5         extern int x;
6         printf("%d\n", x);
7     }
8     return 0;
9 }
```

extern 是一个 storage-class specifier，说白了就是决定了他的值存储在何处。定义了 extern 的量可以被其他模块或其他库引用。全局变量默认是

extern 的。一般我们在引用其他动态库里的全局符号时都知道声明前要加 extern。在同一个模块里访问全局符号其实是一个道理。

4. 序列点 (sequence point)

我估计很多人没听说过 sequence point。但是对于 C 的副作用 (side effect) 应该不陌生, 比如 postfix increment。其实 sequence point 和 side effect 是紧密相关的, 这涉及到编译器的实现。简单的来说, 序列点之前的所有副作用会在下一个序列点开始前全部执行完成。言外之意是, 在两个序列点之间的所有副作用的执行顺序和时机是由编译器的具体实现决定的, C 标准并没有规定。好的 C 语言教材都会告诉我们函数参数的求值顺序, 表达式各个操作数 (子表达式) 的求值顺序都是不确定的, 我们不能假定一定是从左到右或者从右到左。编译器出于优化目的, 在不改变语义的前提下, 会将两个序列点之间的操作进行合理重排序。关于序列点的具体规定可以参考这里 (http://en.cppreference.com/w/c/language/eval_order), 最著名的序列点应该就是 ;, 这是每个 c 语句的结束符。所以通常的建议都是不要在一个表达式里对同一个变量多次使用带有副作用的操作。

5. void bar(int x[static 10]);

这个看起来奇怪的声明是 c99 的新特性。它的含义是对 bar 的每次调用, 实参如果是数组类型, 其不能为 NULL, 并且有至少 10 个元素。但这只是一个建议, 编译器会做有限的提示, 并没有运行时的保证。比如下面的代码:

```
1 int foo(int a[static 2])
2 {
3     return a[0] + a[1];
4 }
5
6 int main() {
```

```
7 int a[1];
8 foo(a);
9 foo(0);
10 return 0;
11 }
```

clang 会给出两个警告:

```
test.c:10:5: warning: array argument is too small; contains 1
elements, callee requires at least 2 [-Warray-bounds]
foo(a);
^ ~
test.c:3:13: note: callee declares array parameter as static
here
int foo(int a[static 2])
^ ~~~~~~
test.c:11:5: warning: null passed to a callee that requires a
non-null argument [-Wnonnull]
foo(0);
^ ~
test.c:3:13: note: callee declares array parameter as static
here
int foo(int a[static 2])
^ ~~~~~~
```

6. int x = '\0CBA';

这玩意没什么用, 但是它的确合理合法。

7. macro

c 语言的宏展开有一个微妙之处, 就是在包含 stringify 的时候的处理逻辑, 下面这个例子足以说明问题:

```
1 #include <stdio.h>
2
3 #define f(a,b) a##b
4 #define g(a) #a
5 #define h(a) g(a)
6
7 int main() {
8     printf("%s\n",h(f(1,2)));
9     printf("%s\n",g(f(1,2)));
10    return 0;
11 }
```

8. e1[e2] 和 e2[e1] 是等价的。

这揭示了一个事实: 在 C 语言里, 没有真正的数组类型, 它其实是指针的别名。编译器会在合适



深度·讲坛 征稿啦

是什么让你豁然开朗？是什么让你灵感迸发？又是什么让你百思不解？

这里汇聚了深度大神，专门凑在这里搞技术，从今往后，这里专治各种疑难杂症，你可以来提问，可以来传授技术，总之，这里都是爱凑热闹的技术控。

分享专业知识，方便学习交流，内刊“深度·讲坛”栏目长期向各位技术控们征集技术稿件啦。

字数：1000+，图文并茂

内容：必须为原创

投稿邮箱：deepin-magazine@deepin.com

注明“部门+姓名”



的地方做 ArrayToPointer 的转换。比如作为参数的 arr:

```
1 int size(int arr[10]) {
2   return sizeof arr;
3 }
```

再比如作为操作数:

```
1 int foo() {
2   int arr[] = {1,3,5,7};
3   return *(arr+2);
4 }
```

函数也有类似的转换过程:

```
1 typedef void (*callback)();
2
3 static void handler()
4 {
5 }
6
7 int main() {
8   callback cbs[] = {
9     handler
10  };
11
12  (****cbs[0])();
13  return 0;
14 }
```

第九行的 handler 是一个函数类型，在语法解析后会被转换为 &handler。在大多数地方 function 都会被转换为 &function。这样第 12 行的表达式才能合理并成立。cbs[0] 是一个 callback 类型的函数指针，*cbs[0] 解引用后是函数类型，然后编译器又将其转为函数指针类型，相当于 *&cbs[0]，于是又可以对其进行解引用。

结语

这里所说的都是以科普为主，如果想更深入的了解 c 语言，应该去看看 c 语言规范或者参考手册。d



志愿者装机活动

深度操作系统的发展离不开社区的支持，同时我们也会及时响应社区用户的需求。为了更好的服务、推广、宣传深度操作系统，让更多的社区用户参与线下互助和传播，通过社区用户反馈的建议，我们推出了免费装机地图活动。

本着奉献、友爱、互助、进步的志愿者精神，志愿者们利用业余时间，帮助希望了解 deepin 操作系统的爱好者们，免费上门安装系统，过程中不仅丰富了自己的专业知识，也结识了更多 Linux 爱好者。

活动开始至今，已覆盖全国 31 个省市，参与志愿者近 300 余人。

捐助渠道

深度操作系统的发展离不开社区的支持，为了进一步完善深度操作系统社区生态环境的建设，深度科技对外正式开通捐助渠道。

捐助秉承着完全自愿的原则，不管你是个人还是组织、来自何处、捐助金额多少，都是对社区的发展贡献出自己的力量，我们对你的捐助深表感谢。

捐助定位：

- 深度操作系统是一个致力于为全球用户提供美观易用、安全可靠的 Linux 发行版。
- 深度操作系统的发展离不开社区的支持，为了进一步完善深度操作系统社区生态环境的建设，深度科技内部通过决策，对外正式开通捐助渠道。
- 捐助秉承着完全自愿的原则，不管你是个人还是组织、来自何处、捐助金额多少，都是对社区的发展贡献出自己的力量，我们对你的捐助深表感谢。

捐助目的：

- 调查各个项目参与捐助的金额和人数，决定项目开发的方向和优先级
- 持续和深入的开发
- 社区活动和建设（例如：内测活动奖品、志愿者奖励以及社区礼品）

 Deepin Talk ¥ 810 已筹		 驱动中心 ¥ 2377.5 已筹	
 DeepinLive修复系统 ¥ 244.89 已筹		 DeepinID云平台 ¥ 413.87 已筹	
 深度开发者平台 ¥ 1236.86 已筹			



debian 系发行版 initramfs 调试方法

● 北京 研发部

简介

在进行 debian 系统开发时，如果开发工作涉及对 initramfs 定制修改，就需要对 debian 的 initramfs 有一个详细的了解。本文希望通过对于 linux 系统普遍使用的 initramfs 和 debian 系统如何自动生成使用 initramfs 的讲解，来减轻读者在 debian 系统及其衍生版本上开发调试 initramfs 时的麻烦。

什么是 initramfs ?

为什么需要 initramfs

首先需要弄清楚，initramfs 是什么东西，要回答这个问题，我们要知道 linux 系统启动时的一般流程：

系统加电 -> bios 中的固件 -> bootloader -> kernel -> 真正的系统

上述过程看起来顺理成章，但实际上从 kernel 到真正的系统启动这一过程中有许多困难，最主要的问题是：根系统可能位于不同的硬件，不同的文件系统之上，而由于内核的模块机制，相应的硬件或者文件系统驱动当时可能并不存在于 linux 内核的二进制文件中，因而这时在 bootloader 加载内核之后，并不能正确识别出根文件系统。

因此在这个地方我们需要一个机制来帮系统从 kernel 正确过渡到真正的根文件系统。这也正是 initramfs 发挥作用的地方，也就是说：initramfs 解决了这个问题。

initramfs

initramfs 是一个压缩过的 cpio 文档，在 debian

系统中，它一般位于 /boot/ 目录下，文件名为 /boot/initrd.img-kversion，在系统的根目录下还有一个 /initrd.img 文件，这个文件是指向系统中安装的最新 initramfs 的软链接。

在使用 debian 系统时，initramfs 由 initramfs-tools 程序自动生成，因此你一般不需要手动修改这个文件，在系统启动时，内核会自动加载这个文件到内存，解压出其中的内容，然后把 initramfs 当作 rootfs 来挂载，挂载之后，会启动 1 号用户空间程序 /init，接下来 /init 完成直到挂载真正的根文件系统之前的所有工作，并最后挂载根文件系统到 /root，这样，系统就正常过渡到了真正的系统。

与真正的 rootfs 的不同

可以看到，initramfs 是系统启动时第一个被挂载的 rootfs，但是它与真正的 rootfs 有很多不同，下面是一些总结：

- initramfs 是完全存在于内存中的一个内存文件系统，而真正的 rootfs 往往都是存在于硬盘上的硬盘文件系统

- initramfs 在体积上做了很多裁剪，包括使用适合嵌入式设备的 klibc 和 busybox 来替换真正系统中的 glibc 和许多应用

修改 initramfs

前面说了，debian 中的 initramfs 主要是 initramfs-tools 在管理，因此在修改调试 initramfs 时，最好还是能直接使用 initramfs-tools 提供的现成的方法来实现。当然，如果发现有些地方直接使

用 `initramfs-tools`，那此时可以考虑修改 `initramfs-tools` 的脚本来实现想要的功能，但是一般不要尝试完全手工去产生一个 `initramfs` 文档。

安装 `initramfs-tools`

`initramfs-tools` 是 `debian` 系统自带的工具，所以不需要手动安装。

配置文件

`initramfs-tools` 的配置文件位于 `/etc/initramfs-tools` 目录下，进入该目录查看一下内容：

```
$ cd /etc/initramfs-tools
$ ls
conf.d/ hooks/ initramfs.conf modules scripts/ update-
initramfs.conf
```

如果要对 `initramfs` 进行定制修改，需要对 `hooks` 和 `scripts` 目录下的文件和 `modules` 进行操作，`modules` 文件中定义了在建构 `initramfs` 时将会被包含的内核模块，每个模块占用一行，可以使用模块参数，`#` 开头的行为注释行。

注意：在使用 `update-initramfs` 时，`-k all` 命令会忽略 `modules` 文件而默认将所有内核模块包含到 `initramfs` 中。

hooks/

可以在 `hooks/` 目录下添加自定义的脚本，该脚本在 `mkinitramfs` 命令执行时会被自动调用，创建的自定义脚本文件内容需要遵守一定的规则：每个脚本文件的起始处必须包含以下几行：

```
#!/bin/sh
PREREQ=""
prereqs()
{
    echo "$PREREQ"
```

```
}

case $1 in
prereqs)
    prereqs
    exit 0
    ;;
esac

. /usr/share/initramfs-tools/hook-functions
# Begin real processing below this line
```

包含这几行的主要目的是用来确保脚本能以一定的顺序执行，比如你需要确保这个脚本在执行之前，`lvm hook` 脚本先被执行，那么你需要将 `PREREQ=""` 替换为 `PREREQ="lvm"`。

`/usr/share/initramfs-tools/hook-functions` 脚本提供了许多在写 `hook` 脚本时很有用的函数，具体信息可以参考 `man initramfs-tools`。

scripts/

`scripts/` 下有很多子目录，这些子目录分别代表了 `initramfs` 执行的不同阶段，可分别在这些子目录下创建自定义的脚本，在 `initramfs` 被挂载后，自定义脚本会在相应的阶段被执行，从而达到定制 `initramfs` 行为的目的。

- `init-top` 这个目录下的脚本在 `initramfs` 挂载了 `sysfs` 和 `procfs` 后将首先被调用，这个阶段也会调用 `udev` 脚本来初始化 `/dev` 目录下的设备文件
- `init-premount` 当 `hooks` 脚本和 `/etc/initramfs-tools/modules` 文件中指定的内核模块加载后被调用
- `local-top` 或者 `nfs-top` 这些脚本被调用后，真正的 `rootfs` 此时已经可见或者 `nfs rootfs` 已经可用
- `local-block` 这个子目录下的脚本会和相应的块设备文件一起被调用，之后应该可以正确识别相应的块设备文件，如果 `local-top` 或者 `local-block` 调用失败，则相应的脚本会被定期调用重试



- local-premount 或者 nfs-premount 这两个子目录下的脚本在真正的 rootfs 的一切状态都已被验证后被调用。

- local-bottom 或者 nfs-bottom 当真正的 rootfs 被正确挂载后被调用

- init-bottom 这个子目录下的脚本在最后被执行，相应的脚本执行完成之后，sysfs 和 procs 将被真正的 rootfs 下，系统控制权将被移交到真正 rootfs 下的 /sbin/init，initramfs 中的 udev 也将停止运行

在上述子目录下创建的自定义脚本如果想要使其依据一定的顺序来执行，也需要在自定义脚本之前添加 hooks scripts 相似的头部。

```
#!/bin/sh
PREREQ=""
prereqs()
{
    echo "$PREREQ"
}

case $1 in
    prereqs)
        prereqs
        exit 0
        ;;
    esac
```

调试 initramfs

1. 进入 rescue shell 在修改了 hooks scripts 和 boot scripts 之后，如果想要进一步确认文件和功能的正确性，可以通过在添加 break= 内核启动参数来使 initramfs 在特定的阶段暂停执行，并且提供一个命令行环境，此时你可以通过该命令行环境来检查 initramfs 中的各项是否正常

2. debug log 在内核启动时添加 debug 启动参数，initramfs 会输出一个详细的 log 文件，并将其保存在 /run/initramfs/initramfs.debug 中，可以通过查阅该文件来定位问题

一个小例子

本节以一个小例子来讲解 initramfs 的具体使用方法。

情景：假设在安装系统的时候，你选择了使用全盘 luks 加密的方式来安装系统，那么在每次系统启动时候你都会被要求输入一个密码来对系统盘进行解密，然后才能正常进入系统。这么做是处于安全的角度来考虑，但是每次开机都输入两次密码（一次解密硬盘，一次用户密码）会让人觉得非常繁琐，因此我们可以对 initramfs 动一些小手脚，让它来帮我们自动完成分区的解密。

首先需要创建一个 hook scripts，位于 /etc/initramfs-tools/hooks 目录下。

```
#!/bin/sh
PREREQ=""
prereqs() {
    echo "$PREREQ"
}

case "$1" in
    prereqs)
        prereqs
        exit 0
        ;;
    esac
. "${CONFDIR}/initramfs.conf"
. /usr/share/initramfs-tools/hook-functions
if [ ! -f "${DESTDIR}/lib/cryptsetup/scripts/loadkeyfile.sh" ];
then
    if [ ! -d "${DESTDIR}/lib/cryptsetup/scripts" ]; then
        mkdir -p "${DESTDIR}/lib/cryptsetup/scripts"
    fi
    cp /lib/cryptsetup/scripts/loadkeyfile.sh ${DESTDIR}/lib/
    cryptsetup/scripts/
fi
if [ ! -d "${DESTDIR}/root" ]; then
    mkdir -p ${DESTDIR}/root/
fi
cp /root/keyfile ${DESTDIR}/root/
```

其中 keyfile 是你用来加密系统分区的二进制文件，以上 hook script 的作用是将该 keyfile 和用来进行分区解密的脚本放入 initramfs 中。

然后创建 loadkeyfile.sh 脚本，

deepin

因理想而出生 为责任而成长

We do we change...



国产操作系统 我们更专注

```
#!/bin/busybox ash

if [ -f /dev/sdb1 ]; then
    echo "mount the device containing keyfile ..."
    mount /dev/sdb1 /root/key 2> /dev/null
fi

KEY="${1}"
if [ -f "${KEY}" ]; then
    cat "${KEY}"
else
    echo "FAILED to find suitable USB keychain ..." >&2
    echo -n "Try to enter your password: " >&2
    read -s -r PASSWD < /dev/console
    echo -n "$PASSWD"
fi

if [ -f /dev/sdb1 ]; then
    umount /ev/sdb1 2> /dev/null
fi
```

需要修改一下 /etc/crypttab 文件，改为如下内容：

```
sda2_crypt UUID=563edceb-392c-49f6-9ace-
e9b8a6b418d2 /root/key/keyfile luks,keysript=/lib/cryptsetup/
scripts/loadkeyfile.sh
```

最后，重新生成 initramfs。

```
sudo update-initramfs -u -k all
```

重启之后，initramfs 就能自动解密系统分区了！ **d**

国际影响

深度操作系统下载量超过5000万次，提供30种不同的语言版本，是全球开源操作系统排行榜上排名最高的中国操作系统产品

更强兼容

支持大量常用Windows应用软件运行，兼容各类国内数据库、中间件与虚拟化产品

专业研发

公司拥有操作系统研发、行业定制、国际化、迁移和适配、交互设计等多方面专业人才，能够满足不同用户和应用场景对操作系统产品的广泛需求

支持服务

深度科技面向商业用户提供专业支持服务。
顾问咨询/远程服务/现场服务/ 培训服务/定制服务

武汉深之度科技有限公司
Service&Consulting/服务与咨询
400-077-9127

武汉：武汉市光谷大道77号光谷金融港B18栋6楼
北京：北京市海淀区知春路锦秋国际大厦B座501室
上海：上海市长宁区愚园路1258号15A01室

www.deepin.com



玩转深度服务器安装部署

● 北京 研发部

概述

面向读者：

- 具有一定的 LINUX 系统经验
- 具备一定的网络基础知识
- 能够编写调试 preseed.cfg 文件

来到深度科技从事服务器版本研发一载有余，结合之前为解决工程团队实际工程部署遇到的问题，将常见系统安装部署的方式，及其优劣差异做一小节，汇总如下！

手动安装

刻录光盘和制作启动 U 盘已经是当前最普遍的安装方式，已经没有什么太多可重复说的了，玩转服务器不要依赖桌面环境，能敲 shell 命令来做的事情就不要使用图形工具来操作，在这里我只想分享两条命令，让你在命令下玩的更轻松！

- 命令行下刻录光盘：

首先要确保系统中 xorriso 软件包被安装，将空光盘放入刻录光驱，执行如下命令完成光盘刻录！

```
xorrecord -v dev=/dev/sr0 speed=8 fs=8m -waiti -multi
--grow_overwriteable_iso -eject padsize=300k deepin-
server.iso
```

- 命令下制作安装 U 盘

当下，U 盘已经完全可以替代光盘，制作好的安装 U 盘可以像使用光驱设备一样来安装系统，使用 `dd if=deepin-server.iso of=/dev/sdX bs=8M` 命

令可以将 iso 文件复制到 U 盘设备。

注意事项如下：

- 制作好的安装 U 盘是只读设备，实际安装过程 U 盘被模拟为一个光驱设备
- 如要恢复 U 盘原有用途，请使用命令 ``dd if=/dev/zero of=/dev/sdX bs=512 count=1``，清空分区表重新格式化即可
- `(/dev/sdX` 请根据 U 盘实际对应的设备名来选择，例如 ``/dev/sdb /dev/sdc ...``)

自动安装

服务器的单机安装已经是 IT 的上古时代了，现在已经是动则是数十台，数百台的批量部署时代，针对工程团队遇到的实际情况，研发团队采用微创新解决了不同场景的安装自动化。

U 盘自动安装

场景描述：在一个新建机房，上架机器不多，十



几台到数十台，没有网络，机器配置不一样，不能拆机硬盘对拷 ... 如何实现快速部署？研发团队和工程团队花了两周的时间，通过不断测试和调整，完成了深度服务器 U 盘自动化安装，具体细节如下：

- 通过微调服务器安装器，让其默认从外部读取 preseed.cfg 文件
- 通过适配深度启动盘制作工具，制作一个可读可写的安装 U 盘
- 将准备好的 preseed.cfg 文件，替换启动 U 盘中的 preseed.cfg，开机从 U 盘启动，就可以进入开始自动安装模式

网络自动安装

场景描述：上架机器几十台到数百台，机器配置相近，所有机器在同一网段，但是不能访问外网。基于上述条件下，搭建 PXE 自动安装部署环境，需要完成如下配置和准备工作：

- 获取 netboot.tar.gz
- 编写 preseed.cfg
- HTTP 服务器的配置
- TFTP 服务器的配置
- DHCP 服务器的配置

配置 HTTP 服务器

执行命令 `apt-get install nginx -y` 完成 nginx 的安装，修改配置文件 `/etc/nginx/sites-available/default` 完成后重启 nginx 服务。

参考配置如下：

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    root /var/www/html;
    autoindex on;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

搭建内网软件仓库

将 ISO 文件内的 `dist` 目录拷贝到 `/var/www/html` 下，进入 `dist` 目录创建链接。

```
ln -sv kui kui-security
ln -sv kui kui-updates
ln -sv kui stable
```

编写 preseed.cfg 文件

在 `/var/www/html` 根目录下创建 `preseed.cfg` 文件。

- 可以通过命令 `debconf-get-selections --installer > file` 获取 preseed 参考配置
- 可以通过命令 `debconf-set-selections -c preseed.cfg` 检查 preseed 文件的语法错误

配置 TFTP 服务

执行命令 `apt-get install tftpd-hpa -y` 完成 tftp 服务器的安装，修改配置文件 `/etc/default/tftpd-hpa`，完成后重启 `tftpd-hpa` 服务。

参考配置如下：

```
TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/srv/tftp"
TFTP_ADDRESS="0.0.0.0:69"
TFTP_OPTIONS="--secure"
```

将 `netboot.tar.gz` 文件解压到 TFTP 服务器的根目录下，修改启动项 `debian-installer/amd64/boot-screens/txt.cfg` 参考配置如下：

```
default Install-Deepin-Server
label Install-Deepin-Server
    menu label "Install-Deepin-Server"
    menu default
    kernel debian-installer/amd64/linux
    append vga=788 initrd=debian-installer/amd64/initrd.gz
    auto=true priority=critical url=http://http_server_ip/preseed.cfg --- quiet
```

以上配置仅供参考，其中 `http_server_ip` 请修改为实际 IP。



配置 dhcp 服务

执行命令 `apt-get install isc-dhcp-server -y` 安装搭建 dhcp 服务的软件包。

- 修改 `/etc/default/isc-dhcp-server`，指定网络设备

```
INTERFACES="eth0"
```

- 修改 `/etc/dhcp/dhcpd.conf`，完成关键配置，指定 pxe 引导文件名称和存放位置，修改完成后重启 `isc-dhcp-server` 服务，参考配置如下：

```
default-lease-time 600;
max-lease-time 7200;
allow booting;

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.100 192.168.1.200;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.1;      # our router
    option domain-name-servers 192.168.1.1; # our router,
again
    filename "pxelinux.0";          # 指定引导文件名称
}

group {
    next-server 192.168.1.108;      # 指定 TFTP 服务器 IP
    host tftpclient {
        filename "pxelinux.0"; # (this we will provide later)
    }
}
```

以上配置仅供参考，其中 `subnet`，`routers`，`next-server` 等配置请根据实际情况修改。

排错和测试

- 可通过观察 `tail -f /var/log/syslog` 来检查服务的运行状态

- 安装过程出错，可以通过 `Control+Alt+F4` 组合键来观察日志输出

preseed.cfg 参考配置

```
### Localization
d-i debian-installer/locale string en_US
d-i debian-installer/language string en
d-i debian-installer/country string CN
d-i debian-installer/locale string en_US.UTF-8
d-i localechooser/supported-locales multiselect en_US.UTF-8,zh_CN.UTF-8, zh_CN.GBK, zh_CN.GB18030, zh_CN.GB2312

### Keyboard selection.
d-i keyboard-configuration/xkb-keymap select us

### Preseed Early
d-i preseed/early_command string kill-all-dhcp; netcfg

### Network configuration
d-i netcfg/choose_interface select auto
d-i netcfg/hostname string deepin

d-i hw-detect/load_firmware boolean true

### Mirror settings
d-i mirror/country string manual
d-i mirror/http/hostname string http_server_ip
d-i mirror/http/directory string /
d-i mirror/suite string kui
d-i mirror/udeb/suite string kui
d-i debian-installer/allow_unauthenticated string true

### Account setup
d-i passwd/root-login boolean false
d-i passwd/username string deepin
d-i passwd/user-fullname string deepin
d-i passwd/user-password password A4H3ljbn
d-i passwd/user-password-again password A4H3ljbn
```

```

### Clock and time zone setup
d-i clock-setup/utc boolean true
d-i time/zone string Asia/Shanghai
d-i clock-setup/ntp boolean false

### Partitioning
d-i partman-auto/method string regular
d-i partman-auto/choose_recipe select atomic
d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true
d-i partman/mount_style select uuid

### Apt setup
d-i apt-setup/non-free boolean true
d-i apt-setup/contrib boolean true
d-i apt-setup/local0/repository string http://http_server_ip/ kui
main non-free contrib
d-i apt-setup/security_host string
base-config apt-setup/security-updates boolean false

### Package selection
tasksel tasksel/first multiselect
d-i pkgsel/include string openssh-server vim
popularity-contest popularity-contest/participate boolean false

### GRUB
d-i grub-installer/only_debian boolean true
d-i grub-installer/bootdev string default

### Finishing up the installation
d-i finish-install/keep-consoles boolean true
d-i finish-install/reboot_in_progress note
d-i debian-installer/exit/poweroff boolean true

```

以上配置仅供参考，其中 `http_server_ip` 请修改为实际 IP。

优劣差异

• 手动安装

优点：最通用的安装方式，安装过程可交互

缺点：只能一台台的安装

• 自动安装

* U 盘自动安装

优点：步骤简单，制作调试方便，不依赖网络

络

缺点：仅仅适用于小规模机器安装部署

* 网络自动安装

优点：大规模部署具有明显的效率优势

缺点：搭建步骤繁琐，严重依赖网络

下载链接和文档参考

• 网络安装压缩包下载地址：

<http://dl.sndu.cn/netboot.tar.gz>

• 深度服务器版本下载地址：

<http://dl.sndu.cn/deepin-server-15-general-amd64-DVD-20161028.iso>

• 深度启动盘制作工具下载：

<https://www.deepin.org/download.html>

• Debian 社区 preseed 参考：

<https://www.debian.org/releases/etch/amd64/apbs04.html.en>



“深度” 的泰国之旅



● 北京市场部 / 文

11月9日，这天本该是一周中最难捱的星期三，但对于深度科技部分同事来说，这是一个开心快乐的开始，因为公司的秋游计划由此开始啦！

先不去理会我们这批是如何从第二批转变到第一批的，反正一颗小心脏都已被兴奋占据。从没坐过夜班飞机的我，从窗口望去，深邃的黑夜仿佛把我吸得很牢，夜空不仅可以看到星星，还仿佛变成了电影院的黑幕，我们几个女生的彩色泰国之旅，突然变得清晰，毫不留恋地告别帝都的霾，下了飞机就只剩阳光、沙滩和笑声。而对于这场浩浩荡荡的集体出行，多希望自己的心能褪去岁月赐给的斑驳，让内心深处的童心在人群中默默地迸发。

披着泰国凌晨的星星，褪去帝都裹来的大衣，上了大巴我们直奔酒店，出行前“小领导”的嘱托时刻催促着我们的脚步，收拾行李、洗漱、早餐一直到大家又重新坐上



大巴，全部都是按了快进键的节奏，只为晚上可以挤出一些时间摆脱旅行团，集体自由行一下。

第一站是鳄鱼湖，其实在中国也见过鳄鱼和鳄鱼表演，但泰国的鳄鱼数量和个头的确是个惊喜。它们冷漠的趴在鳄鱼池里面，我看不清它们的眼睛在注视哪里，只觉得黑压压的一片又一片，它们有的在水里懒洋洋的搜索食物，有的趴在岸上相互依偎、叠压，它们身长有的近 10 米，我不敢想象如果是我掉下去，会不会整个人被吞下去，或者瞬间被撕扯殆尽。

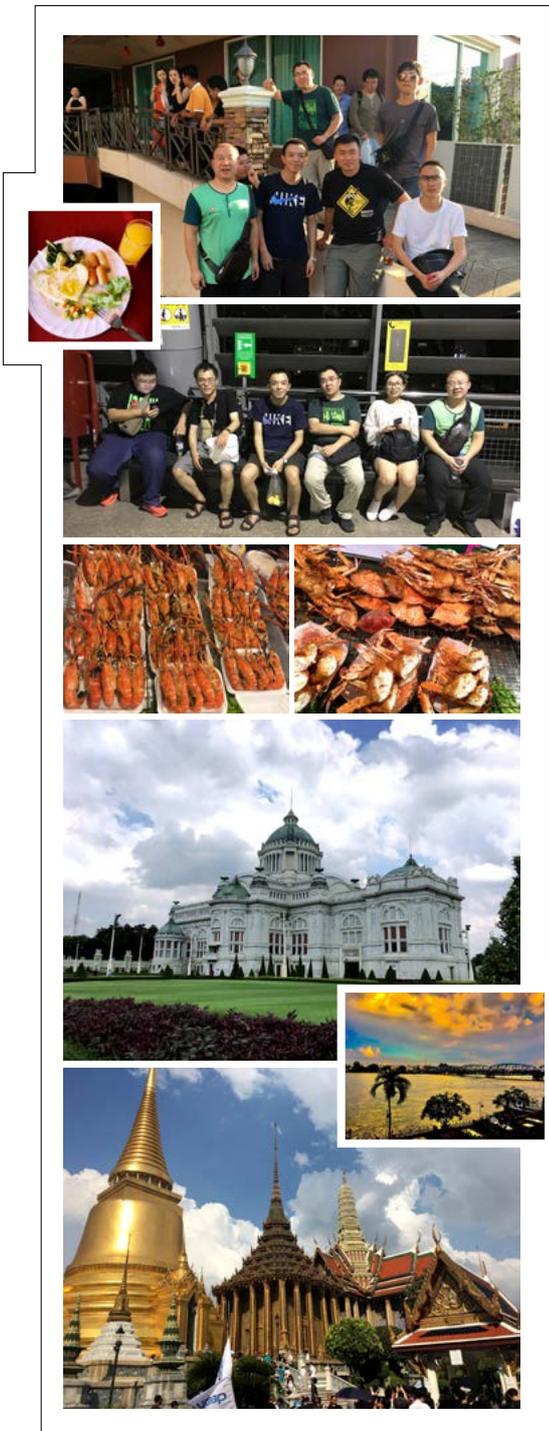
关于鳄鱼表演，首先让我惊讶的是表演者流利的中文，因为这样他们就可以顺利索取小费，以供养他们的生活。表演者将手和头放入鳄鱼的嘴里，我觉得有些残忍，我当然知道在台上他们不会怎样，可又有谁看的到他们表演前的恐惧，不知道他们的家人是否会有担忧，只是突然觉得生活不易，其实在今后的几天，每天都会有强烈的感觉，也许他们没有生活的技能，只能用危险的表演或者身体去换取生活。人生下来，好像不该是这样的。

接下来我们去体验了泰式按摩，行程中按摩时间为 2 小时，之前还担忧按摩师两个小时下来，是

否自己都需要按摩了，还好中途换了一拨。从没有过那么仔细的按摩过，据说泰式按摩是泰国的妃子为取悦国王发明的，后来才传入民间。我感觉这个妃子一定很爱他的国王，每一处的疲惫都被一一消灭，每一寸的肌肉都能得到缓解，每一秒钟都有一种被关心的感觉。慢慢的在昏暗的灯光下，我睡着了，醒来时，真的觉得全身都很轻松，非常感谢按摩师。虽然我们言语不通，但她们的默默的、甜甜的、温柔的微笑深深烙在我的心里。

下午在酒店休息之后，我们开始了自由行，第一次在外国坐地铁，很多人都是一脸的茫然，还好有个有经验的国外旅行小能手在我们的队伍中，买票、上车、转车、下车全都被很好的引导，我们像





一群抱团的小蜜蜂，在这样一个陌生的国度，大家在一起很有安全感，接下来我们逛了曼谷的市中心，很多的小吃和店铺。一路上的步行，感受着泰国每一寸的味道。

第二天我们到了阿南达皇家御会馆和大皇宫，不得不赞叹泰国匠人高超的手工艺。御会馆里面大幅刺绣，宝石和钻石巧妙地镶嵌缝合上去，里面的人物惟妙惟肖，形态各异，特色的植物与文化，被丝线和宝石表现到了极致。皇家的轿撵和船，也全部是黄金和钻石宝石镶嵌而成，可见泰国皇室的富足与奢华。

大皇宫据说只有北京故宫的三分之一大，但含金量绝对超出了几倍不止。来了大皇宫，真的就能看出泰国是多么尊佛的国家，来这里参拜佛像的人都很虔诚，人们要脱鞋进去参拜，在这里我也许下了小小的心愿，只是不知我这类非资深教徒能否得到庇佑。

这里不得不提的是，因为我们这次的游览期间正在国丧期，所以大部分人都是黑衣黑裤黑鞋的扮相，就算有人没有穿了一身黑，也会在胳膊上别一个黑色的结，以表示对逝世国王的哀悼。就是这样齐刷刷的黑色，让我们 deepin 的合影多了一份黑社会的感

晚上我们去了考山路，这里完全是年轻人天堂，不太宽的小路，被商贩占满，卖衣服的、卖小饰品的、卖工艺品的，还有椰子冰激凌、油炸各种昆虫、画纹身、编脏辫，还有数不清多少家的露天酒吧和按摩店，这里鱼龙混杂，身旁走过的都是语言不通的、肤色不同的人，大家哈哈大笑着，完全不用顾虑什么，这里每一寸时光都写着自由与放肆。



第三天在一番购物之后，我们来到了中国台湾偶像剧《流星花园》男主角道明寺的家，导游说这家的公子正在找老婆，于是我们大家把目光都投在了队伍中的单身女生身上，一路上大家都讨论着如果她成了这家的女主人，我们每个人将会被安排什么样的工作，毕竟这是个可以眺望太平洋，有 2000 多名仆人的“大户人家”。这里真是太美了，我想

这家的规划者，一定是一个非常浪漫的人，360 度无死角的浪漫元素，虽然据说他是靠养鸡卖鸡发的家。富丽堂皇的宴会厅，据说只是地毯都是请了 N 多手工艺人在宴会厅不舍昼夜之后做的。广场上的喷泉和雕塑，点点滴滴都要媲美皇家建筑的样子，别具格调的露天广场，唯美梦幻的花园草坪，这都是梦中婚礼的绝佳场地。





时间很快到了快要落日的时候，我站在太平洋的岸边，看着广阔无边的海洋一层又一层地拍打着岸边的大石头，天边由蓝色变成红色，大海的颜色也慢慢暗下来，一对恋人背对着大海，变换着各种亲密的姿势在自拍，真美，真的很美，我闭上眼睛，仿佛看到了他们已经白了头发，又回到这里，此时此刻的样子在自拍，真是一个浪漫的海边。

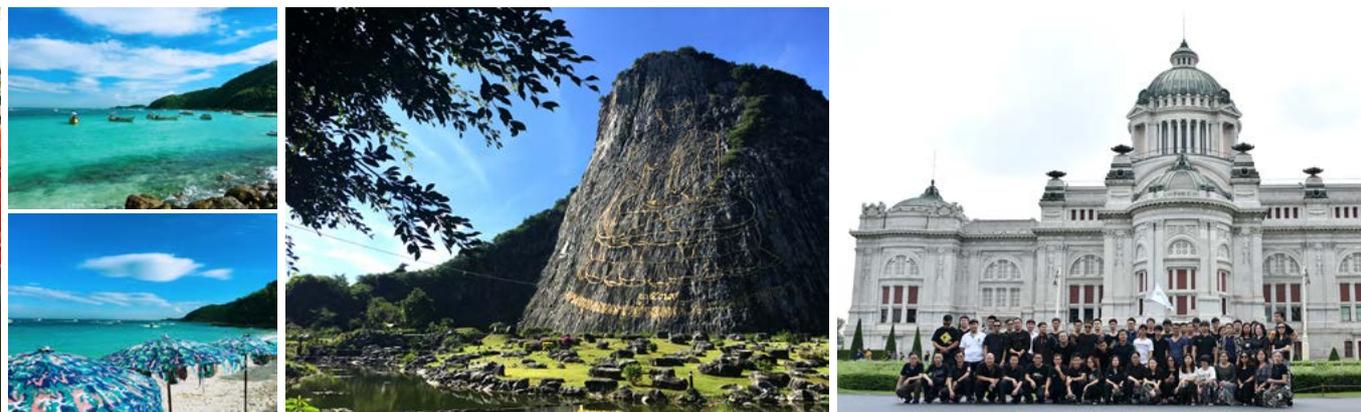
什么时候，人们可以停下飞速旋转的大脑和奔波的双腿，找一处这样安静的岸边，简单地只是看看落日和发脾气的浪花，感受变幻色彩的天空和摸摸浪花溅在脸上的水珠，有多久了，心没有这样的静过，好像世界只有自己在呼吸一样，没有厚重和紧张，不用去整理被风吹乱的秀发，不用在意脸上的妆是否花了。湿润的空气和有点嘈杂的浪花就是全部，难得的放空让我对这里的留恋不止于旅行记忆。

第四天我想可以用阳光和沙滩来形容，哦，对了，还有划伞。在我们女生看来，这已经算是一种极限挑战了，当我被扣上各种安全带之后，面对几步之外的海平面，我有无限的恐惧，我害怕失重的感觉，

害怕手脚无措的感觉，但当我被快艇的速度拉上天空的那一刹那，我的眼里只有蓝天和蓝色的海洋，还有温柔的空气抱着我，远处的建筑一点一点变得渺小，原来世界如此广阔，海天一线的尽头或许是另一种广阔，真的一点都不想下来，多希望自己是一只无拘无束的鸟，可以在没有阻碍的天空中翱翔。

结束了划伞，我们来到一处不是很大的海滩，躺在躺椅上，看着远处游泳、划艇的人们，它们慢慢变小，小到成为一颗颗光斑。明明是很热的天气，可我一点热的感觉都没有，光着脚丫踩在柔柔的沙滩上，这跟我梦里的海滩一样，有蓝绿色渐变的大海，有暖暖沙滩，心里甚至泛起了一丝丝的小幸福，想支起画架，在调色盘上调出这样晶莹的色彩，将它留在画纸上，镶嵌在画框里，将这样的温暖和幸福带回家。其实我们都有一些责备时间太快，还没来得及将阳光刻在皮肤上，就返回了岸边。

我们顶着大太阳逛了其他景点，下午吃了不限量供应的热带水果宴，晚上登上了东方公主号，这里有很多美过女人的人妖，她们奔放热情，表演的舞台在房间的中央，四面环着密密麻麻的桌椅，但



桌上的餐饮根本就不是主题，所有人的目光，都在这些美丽的“姑娘”身上，而坐在第一排的，尤其是男士，便开始了无穷尽的“被骚扰”。

说起“人妖”，这或许是人类历史上最苦难的故事之一，一个出生在充满阳光的热带地区的小男孩，本该在爸妈的呵护中成长为一名顶天立地的男子汉，但却被生活逼迫，以至于在懵懂的年纪或者有了自主选择权后，去过本不该属于他的生活，从此武装变红妆，想要在那样的世界里做到最好，就要忍受各种痛苦，我看他们在台上笑的妩媚，手里还不停的索取小费，突然觉得他们好可怜，如果有下辈子，他们还会做这样的选择吗？

最后一天的行程，在疯狂购买纪念品后，我们来到了美丽的葡萄园和七珍大佛。七珍大佛据说是用了4吨黄金镶嵌而成的，颇有气势。大佛采用镭射激光技术雕刻而成，高度达130米（山的高度180米）。站在远处，如果太阳光的角度刚刚好，整尊佛便会闪着灿灿的金光。在这里又不得不佩服泰国匠人了。

晚餐本是整个行程中最豪华的，但大家拖着疲惫的身体，有些同事已经严重食欲不振啦，候机的时候大家三三两两坐在一起，完全没有来时的精神，然而疲惫中还是留恋泰国的阳光，留恋假期带来的放松感。

旅行赶在国丧期，使这次泰国之行变得没有那么多艳丽的色彩，但却看到了泰国的另一面，他是个落后中国十几年的资本主义国家，在曼谷有六百多万人口，却有七百万辆车，他们的人民很爱他们的国王，以至于满大街只要是稍微有点规模的店铺工厂，都挂着黑白的奠绸，旅游景点和地铁商场里面也都有小小的祭奠台，街上的人装束深沉或者在肩膀带着黑丝带。还有在参观大皇宫时，全身黑装并排队的人群也吸引了我们的注意力，他们顶着大太阳，可能都排了一天队了吧，只为瞻仰一下国王最后的面容，街上有各种煮菊花茶和发矿泉水的，开始导游还强调我们作为游客，不要去索取，但在累的跟哈巴狗的时候，却有两个美丽的姑娘递给我们冰镇菊花茶，即便我们操着中国人的口音，但她们仍示意我们可以的、没问题。嗯，没错，这是个微笑之都…… 



与军队客户的三问三答

● 咕咚（陆军某部）/文

1问

民企供给侧对接军方需求侧
应该注意什么？

答：时下，“供给侧改革”已成为经济领域热词。供给和需求是市场经济内在关系的两个基本方面。没有需求，供给就无从实现，新的供给可以创造新的需求。可以说，二者如同一枚硬币的两面。作为民企，要善于从“供给侧”入手，更好地对接军方“需求侧”，努力实现军民双方互利共赢。

对接军方需求侧，首要的是准确把握部队各级机关和基层官兵的真实需求。以往，军地商务活动曾不同程度地存在供需不对等的弊病：有的公司凭主观想象搞研发，靠自我感觉抓服务，出现了“给的不是官兵想要的”等尴尬；有的公司供给不精准，关注通用共性需求多，满足军队个性需求少，搞千人一面的“粗放式”供给，得不到军方认可。如今，随着部队信息化程度的不断提升，官兵的IT需求已不再是单纯的文字处理、网页浏览，更多的是办公自动化、即时通讯、图像及音视频处理等复杂应用；更多的集中文印、集中存储、虚拟桌面等复杂环境；更加注重安全保密、使用便捷、无缝兼容、界面美观的用户体验。供给方若不能顺应这个趋势，必将南辕北辙，无功而返。

对接军方需求侧，民企要“最大限度满足甚至超越用户需求”。深度科技作为操作系统行业唯

——一家民企，与“国家队”同行相比，深度产品已具备很多局部优势，但整体特色尚不突出，应该多从“供给侧”发力，提高源代码的自主可控水平和配套硬件的国产化率，强化保密性能，努力打造国内领先的安全加强型军用操作系统，形成自己的核心竞争力，以在党、政、军这一广阔市场占得先机。

2问

“民参军”
需要哪些资质和条件？

答：民企参与军队的科研、生产和采购供应，必须获得相应的单位资质和产品资质。我归纳为五证、三网、两名录。

单位资质，首先是大家常说的“军工四证”，包括国军标质量管理体系认证、武器装备科研生产单位保密资质认证、武器装备科研生产许可证、装备承制单位资格证。这四证主要由军队装备部门和国防科工局负责审查，其中，保密认证还需要国家保密局的把关。国军标认证、保密认证是许可证、资格证的前提条件，可以同时启动国军标认证和保密认证，成功后根据需要再办理许可证和资格证。民企具备了上述四证，将有利于其产品列入制式装备，统一配发全军。此外还有一证，是指国家保密局单独审批的“涉密信息系统集成资质认证”，对于地方IT企业，通常有了这个证就能够满足部队在

招投标、集中采购、联合研发等军地商务活动中的保密资质要求。

三网，指的是全军武器装备采购信息网（www.weain.mil.cn）、军队采购网（www.plap.cn）、军队采购网上商城（mall.plap.cn）。前者由军委装备发展部主管，后二者由军委后勤保障部主管。其中，网上商城主要负责 20 万以下的日常通用办公物资零星采购。目前，军方正在逐步加大这三个网站在军内外的推广应用力度。

产品资质有很多种，但军队主要认两个名录，“解放军信息安全认证产品目录”和“解放军军用安全技术防范产品认证目录”。前者侧重网络安全，后者侧重物理安防。

就深度科技而言，我个人认为，首先应申请保密认证或涉密系统集成资质，二者必居其一，才能迈进军方的门槛；其次，尽快加入三个采购网站的供应商库，并定期关注它们发布的信息动态和招标公告；最后，要加快深度科技系列产品办理军队信息安全产品认证的速度，力争获得军 B 以上安全级别的资质。



您能不能就深度科技如何参与军民深度融合，提几条具体建议？

答：我不是研究军民深度融合问题的专家，只能谈一谈个人的几点粗浅认识。

一是找准切入点。前面讲过了，需求是供给的出发点和落脚点，更是供给的着力点和聚焦点。军

队到底哪里最需要安全可靠的操作系统？哪里最讲究国产化替代？这个问题一定要搞清楚，从这个点切入才能事半功倍。目前，深度科技营销团队已在军内保密部门有了个良好开局，我觉得这个切入点找得就很准。

二是直击痛点。有句流行语叫“痛点就是起点”。就操作系统而言，部队最挠头什么、最烦心什么，就是我们应该解决的痛点。例如：保密部门担心病毒木马反复感染，数据安全缺少保障；系统漏洞如何封堵，有无后门远控；信息化部门关心系统是否缓慢经常死机，是否降低安装升级维护工作量；机要部门关注系统能否支持军用加密算法和加密卡等等。这需要深度科技进一步拓展军中人脉，咨询专业人士，才能想客户之所想，急客户之所急。

三是寻求突破点。商场如战场，常常是“一点突破，全线开花”。东部战区某单位采用一款国产软件后不久，南部战区的兄弟单位也采用了同款软件。深度科技可以协调联系团级部队进行小规模试点部署，打造几个能投入实际运转的案例样板，为将来各级部队推介观摩、示范学习奠定基础。

四是制造沸点。当前，很多官兵还不清楚国外操作系统有什么隐患危害，即使了解一点国产操作系统的领导可能也只知“国家队”，不知深度民企队。这就需要深度科技通过参加展会、举办活动来加大宣传力度，制造营销沸点。军内某部门具有巡回保密演示授课的职能，如果深度科技能提供技术支持，在现场呈现操作系统漏洞和后门攻击的巨大危害，那就相当于在广大官兵中为深度科技免费做宣传、打广告，岂不两全其美？



10 年回顾 我的 Linux 之路

● 黄晖 / 文

2006 年我还在上大学，有一次在学校的图书馆里看到了一本介绍 Linux 操作系统的书籍，忍不住打开看了看，一看就看了好几个小时，觉得这个操作系统挺好玩的：可以根据自己的需要定制软件、可以从源代码编译软件，更重要的是，使用了 Linux 操作系统，让自己看起来与众不同。

人在年轻时，总喜欢通过标新立异来刷存在感，就如同东莞电子厂里的小伙子小姑娘们通过各种拉风的发型来证明存在感。理想是丰满的，现实却是骨感的。在刚开始折腾 Linux，总的来说痛苦多于快乐。那时书本上介绍的 Linux 发行版是 Redhat9.0，但这个发行版在当时来说已经有点老，最重要的是它不支持 SATA 硬盘。而在硬件上，2006 年 IDE 硬盘已经逐渐被淘汰，SATA 硬盘成为主流。

有一天下午在图书馆看完书，觉得光看不练假把式，便跑到学校的打印店买了三张 RH9.0 的 CD 盘，放到光驱里去安装。看到书本里面介绍的启动界面变成了现实，内心相当的激动。可是过了没有多久，就出现了错误，提示无法识别硬盘。那时候还没有养成出现问题第一时间通过搜索引擎来找解决问题的方法，于是打电话给在软件学院上学的同学，他没有安装过 RH9.0，他向他的同学请教，他同学说可能是硬盘没有插好，让我检查硬盘的连接线。我检查了过了，没有问题。这才想起来通过百度搜索如何解决这个问题，网上有许多人都碰到了这个问题，解决的办法主要为一是通过 grub for dos，加上新版本的 initrd.img 和 vmlinuz 来进行引导，

二是使用新版本的 Fedora Core。我决定使用第一种方法，通过无数次的失败，终于学会通过硬盘引导来安装操作系统。

通过网络搜索，知道当时红帽已经开始走新的产品路线，即 Redhat9.0 以后，产品分为企业版本 RHEL 系列和社区版本的 Fedora Core 系列。作为个人爱好者，选择 Fedora Core 系列是再合适不过了，而 Redhat9.0 是 2003 年的发行版，对于喜欢折腾系统、追求新事物的我来说，显然有点太老了。于是我的电脑成了双系统，Windows XP+ Fedora。用了 Fedora 以后，发现 Linux 作为日常办公来说，还是有许多不便。没有 QQ，没有网银插件，没有下载软件迅雷，mp3 要加载插件才能播放，视频也要安装插件才能播放。那时候 Windows 平台有一些第三方 QQ，比如珊瑚虫 QQ，可以显示对方 IP，Linux 平台也有第三方 QQ，叫做 LumaQQ，基于 JAVA 开发的。好不容易安装了，但是太卡了。当然壁纸也是要换的，安装完基本系统，第一件事就是将默认的桌面壁纸换成我的女神高圆圆。修改系统自带的 yum 源地址，将地址改为国内地址，加速软件包下载。

用了 Fedora 一段时间，随着时间的推移，新鲜感就过去了，我这个人天生爱折腾，小时候在家里就喜欢自制小刀、自制木梯子、自制弓箭、拉电线等，那一段时间经常泡在论坛上，主要是 <http://linuxsir.org> 和 <http://Chinaunix.net>，知道还有许多其它 Linux 发行版，比如 SUSE，Mandriva，

Debian, Ubuntu, Gentoo, LFS 等。SUSE 源于德国，许多人对于德国制造有良好的印象，因此对于德国产品总体有很好的印象，我也不例外。下载安装了 OpenSUSE，默认界面比较漂亮，包管理为 yast，打包格式为 rpm。毕业后进入一家 IT 公司做技术支持，主要客户为金融行业，其中工商银行和农业银行都是 SUSE 的客户，那几年我安装了无数次 SUSE 系统，每每看着公司业务系统在 SUSE 上运行起来，那一刻内心总有一种成就感。微博问世以后，关注了 SUSE 的母公司 Novell，参加了它举办的“我与 SUSE Linux 的故事”活动中，我成功中奖，获得巨奖一个“OpenSUSE 10 光盘一张”。

2006 年，Ubuntu 母公司为了推广 Ubuntu Linux，在全球范围内免费赠送 Ubuntu 光盘活动，听闻有公司免费送东西，我也在其官网填写了申请，两个星期以后，收到了从荷兰阿姆斯特丹还是鹿特丹寄来的光盘。Ubuntu 为了推广也是蛮拼的，光盘没有多少钱，但这个国际快递的成本可不低，那几年其花在免费送光盘行动的费用应该不少，用现在的话来说就是“有钱任性”，现在互联网公司“烧钱”获取用户的方法原来南非人早已经玩过，呵呵。

继 Ubuntu 全球免费送光盘以后，SUN 也推出了这样的活动。我也申请了光盘。申请了以后就放在书架里吃灰尘，继续折腾 Linux 系统。安装过 Mandrive, Debian, Ubuntu 这三个版本，觉得自己的 Linux 水平没有什么进展，论坛上稍微有一点难度的问题就回答不了。怎么突破这个困境，让 Linux 水平得到提升呢？我苦苦思索着。直到后来在 linuxsir 论坛碰到 Gentoo Linux。这个系统的创始人原来是 FreeBSD 的大神，玩腻了 FreeBSD 系统，就转到 Linux 阵营了。在 Linux 阵营中 deb 和 rpm 包管理机制为主流的情况下，他使用 portage 树进

行包管理，同时操作系统采用滚动升级。由于安装比较麻烦，在国内 Gentoo Linux 属于小众发行版本。

安装折腾各种版本 Linux 操作系统，成了我大学生活很重要的一部分，在此期间，也通过论坛认识了少 Linux 爱好者，也成了毕业后的一种回忆。对于男人来说，折腾各种版本的 Linux 就像是与不同的女孩子相处，每种 Linux 发行版都有自己的优点与缺点，有自己的最佳应用场所，通过不断地相处、磨合、了解各自的优缺点，最终找到合适的人生伴侣。

大学毕业前，由于这些经历，找到了一份实习工作，进入一家 IT 公司从事技术支持工作。时间过得很快，转眼已经是 2016 年，距离我初次接触 Linux 已经 10 年了。这 10 年里，我毕业、结婚、生子，也换了工作，但工作内容始终围绕 Linux 开展。

Linux 世界也发生了巨大的变化，2011 年开始，国内 Linux 厂商的深之度的 Deepin Linux 逐渐进入了大众的视野，以其自己开发的友好用户界面、软件种类丰富的应用商店等特色，赢得了大量的国内和国际用户，改变了传统 Linux 操作系统在普通用户的印象，比如界面简单、应用程序太少等刻板现象，没有计算机基础的用户也可以很快上手使用 Deepin Linux。原来很多喜欢折腾 Linux 的朋友，由于工作忙碌等原因，也不再热衷于折腾各种 Linux 发行版，纷纷给自己的电脑安装上了 Deepin Linux 操作系统，自带搜狗输入法、WPS 办公软件、网易云音乐、云打印服务、软件商店等，完全满足日常办公需求。

2013 年“棱镜门”事件以后，国家对于信息安全有了前所未有的重视。政府采购已经不允许采购 Window7 以后的 Window 发行版本，对于 Linux 操作系统来说，迎来了前所未有的机遇，Deepin Linux 也在这几年得到了快速发展。最后，希望 Deepin Linux 越来越好，全世界有越来越多的用户。d



为开源和商业正名

● 李建盛 / 文

灵感来自于一位神交已久的资深 IT 传媒人士的微博，虽然看似稀松平常的一段内容，却体现了时下大多数人对于开源认知的浅薄，使我不得不去专门写一篇文章来为开源正名。不吐不快、如芒在背。

起因

上海出差的路上，打开新浪微博，不经意刚好刷到了这么一条：“听到一种说法，‘没本事才开源，是为了跟领先者竞争才开源’，有点意思，为什么开源？Linux 的诞生是为了以一种区别于传统‘大教堂’模式的‘集市’模式提升软件开发效率以及质量，占领 Unix 市场纯属意外。后来变了，Android 为挑战 iOS 而存在，被 kill 掉的是诺基亚；OpenStack 为挑战 AWS 而存在，受伤的是 VMware。”

下面截图：



类似的话，我听的太多了，我一般懒得理会，此次看到的却是鼎鼎大名的 @葑菲 daffodil，甚至是 @陈沙克竟然也调侃认同，我有点坐不住了，以开源布道师自居的我，必须为开源正名，于是有了此文。至于我对这段话的批判见文章末尾。

观点

- 开源就是开源，和商业无关
- 和商业有关的开源，是被为了蒙蔽不明真相的人们的商人们强行牵扯上的
- 开源拥有领先的技术，不是“没本事”
- Android 不是为了 iOS 而存在，OpenStack 也不是为 AWS 而存在
- 诺基亚的失败，与 VMware 的市场下滑和开源的关系是有，但不大，甚至他们都曾试图利用开源来突围
- 开源是未来，因为以人为本

论述

较真，这个词，在本土已经快成了一个悲情的代名词了。这是个缺乏较真的时代和环境。而这些却每时每刻不是你所处的身边的人而存在的，抬起头来，看下你的四周围，就是找不出一个较真的人来。那么，我的正名的意义也在于此。

历史是复杂的，世界更是如此。你必须去以严谨的学术精神去对待这些误解，然后抽丝剥茧的去寻找说服世人。

开源的意义

追溯开源的历史，其实是一部人类美好品质胜利的历史，是一部从垄断到解放的历史，然而凑巧的是在这个过程中，还成就了一些商业的企业，而且最重要的是让更多的人有了稳定、可靠、不菲的收入。毕竟人类的近代史，其实就是商业所驱动、资本所驱动的历史。

让我们回到上世纪 70 年代，毕业于 MIT 的理查德·斯托曼在贝尔实验室做人工智能相关的工作，当他和同事索要一份打印机驱动程序源代码的时候遭到了拒绝，要知道 Unix 下是 Hacker 精神所驱动，默认情况下，代码是公开和共享的。而这次拒绝彻底的改变了软件的历史。随后，斯托曼发起了 GUN 项目，开启了撰写自由软件的宏伟目标，并在 1982 年成立了自由软件基金会，至今仍在推动着软件的代码应该被全人类共享的伟大目标而奋斗！

自由的代价是昂贵的，向来如此！
——《美国队长 2：冬日战士》

斯托曼以其强大的毅力和崇高的目标，当然也拥有高超的编码能力，开始了一些非常重要的项目，如 GCC、Emacs 等，但是在内核方面却进展不大，时代总是存在某些意想不到的巧合，1991 年，还是赫尔辛基大学的学生的 Linus，发起了一个操作系统内核的项目，随后发布的 0.9 叫做 Linux，Linux 和 GNU 的组合，可以形成完整而强大的操作系统功能，完全可以替代掉处于各大公司垄断的 Unix 操作系统，这让很多人兴奋起来了。GNU / Linux 一发不可收拾，随着互联网时代、云计算时代、大数据

时代、移动互联网时代的来临，Linux 集全球智慧，赫然屹立于世界，运行在我们所处世界的各个角落：智能手机、智能家电、天气预报运算、金融股票市场、汽车……

受 Linux 的成功的影响，引起了一些人 Hacker 们的好奇，最为著名的就是 Eric S. Raymond，开始探索这样的软件开发模式，并在 1999 年，以一篇《大教堂与集市》横空出世，也直接影响了被微软捆绑 IE 浏览器逼到墙脚的 NetScape 开源的真正动力。微软免费的市场策略，让 Navgaitor 渐渐失去了很大一部分市场。而没有人愿意 web 的标准被某一个具体的厂商所垄断。而这才是 Firefox 后来积极发展的真正意义。

开源发展到现在，已经成为了默认的方式，最为经典的故事，莫过于微软：微软在 1998 年，时任 CEO 的鲍尔默痛斥 Linux 是癌症，而今年新上任的 CEO 称：Microsoft love Linux，并开源了 .NET 框架、Azure 容器引擎等等一系列重量级的项目。再看国内的 IT 巨人华为，成立开源能力中心，赞助 Linux 基金会、OpenStack 基金会，在 OSCON 上强势 KeyNote，主导开源项目 OpenSDS、OnLab (SUN) 等等举措。

开源和商业模式

没有哪家公司从开源本身赚到了钱，迄今为止，可以分析下现有的开源的几种商业模式：

- 卖报纸模式，即订阅模式，提供服务与支持。
成功案例：红帽
- 软件即服务模式，软件开源，服务收费。
成功案例：WordPress
- 开放核心，插件 / 外挂付费，类似共享软件模式。
成功案例：cloudera



- 间接赚钱模式，成功案例：IBM、Intel、Google、AWS、FaceBook

再返过头来看，以开源项目为背景的相关技术人员：开发者、架构师、技术支持、技术咨询、布道师等等，受益颇多。首先，随着各家使用开源的项目，个人不会被某个公司捆绑，大大提高了自己的薪资筹码，第二，还造就了很多新的岗位，首席开源官(Chief OpenSource Office)、社区经理等等。

开源本身并不能够赚到钱，因为它让所有人唾手可得。源代码面前人人平等，没有任何公司去也不应该去强行收取过去商业软件的许可费用。

开源的技术是领先的

随意举几个例子来论证这个论点：

- Docker，容器技术或者是操作系统虚拟化的技术发展不是一朝一夕的事情了，商业产品如OpenVZ等，但是Docker绝对是胜利者，因为它已经是事实上的标准。开源的对手有LXD和Rocket，闭源产品是没有人考虑的。

- TensorFlow，Google开源的人工智能机器学习框架，刚刚在前几日度过了它的一周年生日，看看它的成就吧：TensorFlow开源一周年，DeepMind打败围棋世界冠军，真是让它风光了一把。

- Hadoop生态，在大数据领域，很难再找到一个能够和Hadoop生态相媲美的项目或产品了，Spark、Hive、Kylin等等，甚至有人认为是Hadoop成就了今天的Apache社区。

·

Android, OpenStack

Android和iOS都是基于开源的操作系统内核，Android从来不是为了对抗iOS而生的，同样

是智能手机，一个走开放的路线，一个走封闭的路线。本无可厚非，至于商业上的胜负，那也很难说，Google似乎从Android直接的盈利机会不多，赚钱的生意更多看到是三星、华为、HTC、小米之类的手机制造商。

话再说回来，Android也罢，iOS也好，基于的操作系统内核都是开源的，Android基于Linux，iOS基于Darwin(BSD)，而Linux和BSD都是开源的，再说开发语言，Android是基于Java，iOS是基于Swift，后者也是开源的编程语言。

再说回来，Android是2003年开始的项目，而Jobs有iPhone的想法时是2005年。

OpenStack 发展概要

OpenStack最初的发起者是RackSpace和NASA，前者贡献了项目Swift，后者贡献了Nova，但是后来OpenStack的发展模式，已经超越了所有人的预期，成为了既Linux项目之后，最大的开源项目，而其中的一个关键决策就在于成立了OpenStack基金会。

不可否认的是，在公有云市场上，AWS是领导者，也是云计算产品事实上的标杆，但是，私有云市场是另外一番天地，至今还无人能够撼动OpenStack的占有率，尽管这个和虚拟化管理会有一些令人疑惑的地方。但是Gartner对此有明确的解释，大家不妨找出今年的报告看看。

诺基亚、VMware 纯属躺枪

诺基亚手机业务，昨天还是人人持有，Connect People，今天几乎消失殆尽。没有抓住智能手机这个快速发展的契机，这个是可以熊彼特的颠覆性



创新解释的清楚点，或许正是因为 Nokia 在前智能手机时代的占有率，导致他发展缓慢。这个历史上的案例数不胜数：柯达的胶卷相机被数码颠覆、索尼的 CD 随身听被 iPod 颠覆。

另外，诺基亚也在发展自己的智能手机操作系统，收购 QT 所属公司 Trolltech ASA，并在 2010 年发布 Nokia Qt SDK，要知道这是和 Android 一样的框架。很可惜，晚了那么几步。

说到 VMware，不应该在这里出现，无论从技术角度，开源布局、还是商业境况来讲，VMWare 都不应该出现在这里。AWS、OpenStack、VMware 这三个名词放在一起都是非常蹩脚的事情。AWS 是公有云，OpenStack 是开源的 IaaS 平台项目，而 VMWare 是一家成功的虚拟化平台的市场领导者。VMware 布局 Spring 开发框架，后来发展为 PaaS 的 Cloud Foundry 绝对是开源布局的经典案例，值得业内人士、欲拥有战略眼光的人崇拜不已。

说到可能威胁到 VMware 业务的，可能就是近来崛起的容器：以 Docker 为核心的一系列操作系统虚拟化生态。尽管 VMWare 也很早就在 Docker 上、编排工具上布局，但是，仍然难以逃脱熊彼特的颠

覆性创新理论。

开源是未来

上周福布斯发表了一篇文章，Apple 将加大开源的力度，回顾下今年的开源大事件，比如沃尔玛、耐克等传统公司加入开源阵营、微软拥抱 Linux、华为发布 SUN 和 OpenSDS、Linux 基金会新成立了区块链、CNCF 等子组织……其实，今年的信息技术界有一个关键词：开源，几乎每天都有重量级的公司开源重量级的项目：Netflix、LinkedIn、Amazon、Google、IBM……

知识经济时代，企业的竞争归根结底是人才的竞争，技术日新月异，企业得找到最能学习的人才，而不是现在掌握了什么的人才！我想引用一句话，最贴切不过了：

In a world fueled by developers, enterprises must speak their language: open source code.

结语

国人受体制化太过于严重，思维严重受阻，这需要一个历程，但也要心存希望！[d](#)

深度科技 · 制度规范小贴士



1. 产检假使用规定变更：

主要对产检假的使用方法做了补充说明，原规定可休产检假的数量和时间不变，补充说明每次请产检假只能在一天内休完，不能拆分成两个半天。

2. 增加有关加班登记的截止时间的规定：

从今年6月份开始，每个月初大家会收到机器人推送的“加班调休时长统计”邮件，邮件里会给大家提供自己当月可用的调休时长做参考，其数据是依据前一个月的钉钉系统请假单和Tower加班日历统计计算得来。由于数据收集工作是由Tower机器人自动完成的，加班数据收集的截止时间是每个月底最后一天晚上12点，因此如果在次月去补填前一个月的加班记录，此数据并不会被机器人收集上来，也会给大家造成不必要的困扰。

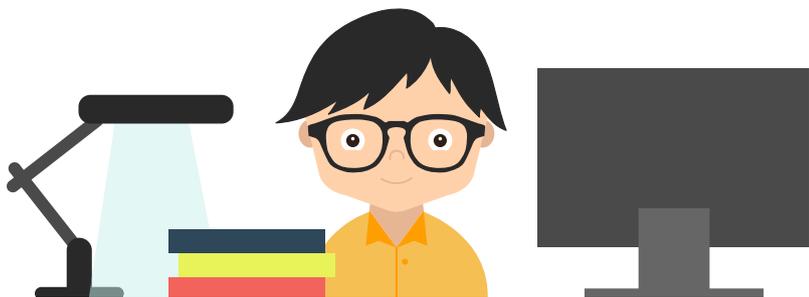
在规则修订中，此次新增每个自然月末的最后一天晚12点是本月加班统计的截止日期，逾期未在加班日历上登记的加班记录，在下月补登记也不会被作为有效记录计入到倒休或者加班费核算中。

3. 对钉钉系统——审批——“付款申请”模块，调整审批流程：

为简化审核手续，加快付款速度。“付款申请”流程中财务经理之后的审批人，由原先的先出纳后王梦境，改为先王梦境后出纳。选择审批人时也不再区分付款金额，一律需添加王梦境。

公司原“够快云库”系统因使用期限原因，以及对其易用程度的综合考虑，已于本月停止使用，新的文档系统正在内部测试过程中，稍后会向大家公布新系统的使用方法。在此之前，对给大家工作中带来的不便深表歉意，也请大家继续关注综合部后续的邮件通知。





deepin 集结 | 征稿启事

《deepin 集结》是深度科技内部刊物，它不仅是记录公司发展历程的一本“画册”，也是公司对外形象和企业建设的窗口。

自 2016 年伊始至今，《deepin 集结》已有三期印刷刊物发放给热爱深度的用户及合作伙伴，并得到了大家的认可。在收集各位读者的反馈意见后，我们将开设新栏目——深度伙伴，主要针对深度员工以外的人员投稿，使《deepin 集结》成为一个企业与用户沟通的刊物，彼此增加交流，分享开源技术。

内部征稿

投稿内容及要求：

形式：摄影作品

Ps. 不要叫你的单反在家睡大觉了，赶紧出来发挥作用吧！

绘画书法

Ps. 你不发出来怎么能知道原来你还这么有才！

诗歌散文、游记、人生感悟

Ps. 知道你原来话不多，但是肚子里面还是有墨水的！

经验技术分享

Ps. 专业达人们，把你们的专业知识拿来 show 一下吧，科普一下啦！

以及能落实到纸上的任何才艺

Ps. 还有多少是我想不到的，快点告诉我！

外部征稿

投稿内容及要求：

1. 内容：a 技术分享

b 用户体验

c 项目评价

2. 要求：a 与行业相关

b 字数 1000-4000

c 文中图片需清晰

深度的同学可以告知身边爱好开源的发烧友积极投稿哦！

投稿邮箱主题需标明：外部投稿（字样）+ 姓名 + 手机号码

要求：我们很看重有图有真相哦，图片和文字说明一样重要。

稿酬：现金（微信红包）或精美礼品

投稿方式：deepin-magazine@deepin.com 邮件名称一定要注明“所在城市 - 部门 - 姓名”

诸位同学有任何问题，都可以立即马上咨询我们：

zhengdan@deepin.com

qindi@deepin.com

deepin

we do we change...