

# deepin 集结

## 【深度·春秋】

02 15.10 研发心得  
新的旅程、新的开始

## 【深度人·在说】

28 用户体验之上，产品设计之下

## 【行业·观察】

40 绝大多数开源软件代码  
不受美国出口限制

## 【深度·社区】

50 统计用 deepin 的人都是做什么工作的

## 【深度·讲坛】

56 使用 Vulns 对 linux 系统进行  
漏洞扫描

59 得到系统调用的函数签名

68 开发 DDE 的屏幕保护程序

14 期  
2019年06月

## 【深度·案例】

52 申威容器云平台方案

P02

深度操作系统

# 15.10

安全稳定  
精细入微



P44

【特别策划】

## 2019

## 首届深度 软件开发大赛

### 决赛入围名单公布

## 【深度·伙伴】

76 玩转 Remmina，远程办公 So easy！

78 在 linux Deepin 深度系统安装多媒体电子教室 Veyon

81 忘掉 Windows 吧！ Deepin 操作系统深度体验



深度操作系统公众号  
www.deepin.com  
内部资料 免费发放

deepin作为一款由中国研发团队打造的Linux发行版，已经多次登上国际Linux发行版排行前十名，并一直活跃在国内外各大社区。自主研发的桌面环境（DDE）目前已经被移植在Arch、Manjaro、Ubuntu、Fedora、OpenSUSE等国际主流发行版，现在，在开源世界，也出现了基于deepin的发行版——ExTiX Deepin。

在国内，deepin被更多地用户称为“最符合中国人使用的Linux发行版”，在深度商城，用户可以非常便捷地找到、查看、下载安装自己需要的应用软件，常用的QQ、微信、百度网盘、网易云音乐、360浏览器都已集成其中。便捷的控制中心，操作体验和智能手机一样简单，只需找到对应操作的图标，便可立即设置，比传统的控制面板操作节省了很多操作步骤。

为了让更多的开发者加入到开发deepin项目中来，已经将代码移到了Github上，大大方便了讨论交流和意见提交。深度软件开发大赛是deepin拥抱开发者的另一个举动，deepin希望更多有才华、有能力的开发者贡献其中，并为国产CPU的软件生态添砖加瓦。

deepin，因理想而生，为梦想而长。

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 DEEPINJIIE

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  
秦娣 Qin Di  
编辑 Editor  
王 棣 王明栋 Wang Di Wang Mingdong  
采编 Assistant Editor  
李会会 黄锦敏 Li Huihui Huang Jinmin  
美术设计 Art Editor  
云云 Yun Yun

网站 Website  
<http://www.deepin.com>  
邮箱投稿 Contribution  
[deepin-magazine@deepin.com](mailto:deepin-magazine@deepin.com)  
市场推广 Marketing  
[account-marketing@deepin.com](mailto:account-marketing@deepin.com)

武汉联络处 Wuhan Office  
地址 Address  
武汉市光谷大道77号  
光谷金融港B18栋6楼  
邮编 430223  
电话 +86-27-87805607

北京联络处 Beijing Office  
地址 Address  
北京市西城区新街口外大街28号普天德胜B座603室  
邮编 100088  
电话 +86-10-62669499

上海联络处 Shanghai Office  
地址 Address  
上海市长宁区愚园路绿地商务大厦1601室  
邮编 200052  
电话 +86-21-62669499

承印单位 武汉金港彩印有限公司  
出版日期 2019年06月  
发放对象 公司员工、用户及合作伙伴、Linux爱好者

本刊为非盈利性杂志，用于传播企业文化，交流学习技术经验  
版权所有，未经同意不得转载。

## 02 深度·春秋

- 02 深度操作系统 15.10—安全稳定，精细入微
- 08 15.10 研发心得，新的旅程、新的开始
- 12 ExTiX Deepin 19.4 发布，基于深度操作系统 Deepin 15.9.3
- 14 Fedora 30 正式版发布：引入 Deepin 桌面环境（DDE）
- 15 WPS Office 2019 For Linux，上架深度应用商店
- 16 深度科技参加关键信息基础设施自主安全创新论坛
- 18 为爱而走，深度集结 用脚步为爱助力
- 20 深度科技参与广东省云浮市“省市共建信息技术应用创新产业基地”工作
- 23 国内首款国产固态硬盘控制芯片发布
- 24 世界知识产权日，深度科技在行动，加强知识产权建设 推动软件正版化
- 26 深度科技携手华为生态共建行动第一

## 28 深度人·在说

- 28 用户体验之上，产品设计之下

## 36 行业·观察

- 36 《中国正在说》听倪光南院士说说我国操作系统与网络安全
- 40 绝大多数开源软件代码，不受美国出口限制
- 43 360 压缩火力全开，全面支持 Linux 操作系统！

## 48 深度·社区

- 48 单一 deepin os 用了 5 个月，很满意
- 49 貌似老外发的关于 deepin 的视频介绍和评价比国内的还多
- 50 统计用 deepin 的人都是做什么工作的

## 52 深度·案例

- 52 申威容器云平台方案

## 76 深度·伙伴

- 76 玩转 Remmina，远程办公 So easy！
- 78 在 linux Deepin 深度系统安装多媒体电子教室 Veyon
- 81 忘掉 Windows 吧！Deepin 操作系统深度体验

## 44 深度·策划



# 2019 首届深度 软件开发大赛 决赛入围名单公布！

- 1 作品征集阶段  
2019 年 3 月 1 日 - 6 月 1 日
- 2 作品初选阶段  
2019 年 6 月 1 日 - 7 月 1 日
- 3 国产软硬平台支持阶段  
2019 年 7 月 1 日 - 9 月 1 日
- 4 通过初赛作品的终审  
2019 年 9 月 1 日 - 10 月 1 日  
.....
- 5 颁奖阶段  
2019 年 10 月 - 11 月

## 56 深度·讲坛

- 56 使用 Vulns 对 linux 系统进行漏洞扫描
- 59 得到系统调用的函数签名
- 62 质量管理体系
- 68 开发 DDE 的屏幕保护程序
- 70 electron 初探



# 深度操作系统 15.10

桌面版

## 安全稳定，精细入微

**深**度操作系统是一款致力于为全球用户提供美观易用、安全可靠的 Linux 发行版。

深度操作系统基于 Linux 内核，以桌面应用为主的开源 GNU/Linux 操作系统，支持笔记本、台式机和一体机。深度操作系统（deepin）包含深度桌面环境（DDE）和近 30 款深度原创应用，及数款来自开源社区的应用软件，支撑广大用户日常的学习和工作。

另外，通过深度商店还能够获得近千款应用软件的支持，满足您对操作系统的扩展需求。深度操作系统由专业的操作系统研发团队和深度技术社区（[www.deepin.org](http://www.deepin.org)）共同打造，其名称来自深度技术社区名称“deepin”一词，意思是对人生和未来深刻的追求和探索。

深度操作系统 V15.10 修复了已知的各种问题，

并对现有的部分功能进行了优化和改进，深度操作系统 V15.10 基于 Debian stable 仓库全新构建并发布，我们将对 unstable 版本提供为期三个月的维护周期。

与上一版本相比，新增了桌面文件自动整理、壁纸轮换、时尚模式托盘拖出等功能，基于 Debian stable 发布，大大提升了系统底层的稳定性和安全性，希望能给用户带来更稳定、更高效和更美观的操作体验。

### 新增的功能

#### 基于坚若磐石的 Debian stable 仓库全新构建

深度操作系统 V15.10 由 Debian stable 全新构建并发布，与原 Debian unstable 版本相比，可获得更为及时的安全更新，并且 stable 更为稳定，迁移至 stable 后稳定性和安全性将大幅提高，为您提供一个更为稳定的操作环境。

深度操作系统 V15.10 会面向 stable 和 unstable 版本同时提供更新。stable 版本的用户可以通过 V15.9.2Beta 版升级或安装本次发布的 ISO，unstable 版本的用户可通过系统升级更新至 V15.10 版本。我们推荐所有用户安装本次发布的 stable 版本 ISO，unstable 版本将于 2019 年 7 月停止技术支持。



### dde-kwin 正式发布

深度操作系统 V15.10 选用 dde-kwin 作为默认窗口管理器，dde-kwin 内存占用更小，性能更好，为您带来更流畅、更美观的使用体验。

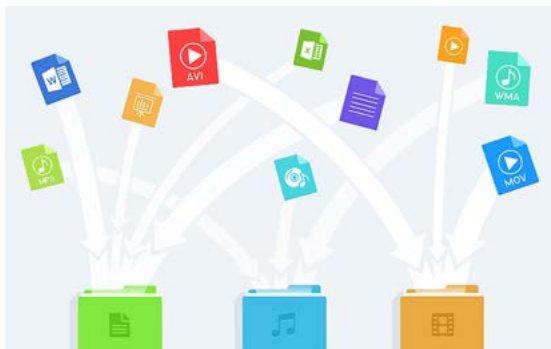
dde-kwin 在 V15.10 将作为 stable 版本的默认窗口管理器，unstable 版本将沿用之前的 deepin-wm。





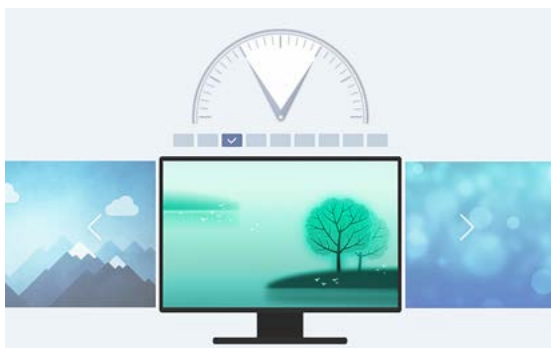
### 桌面文件自动整理——告别杂乱，井然有序

自动整理桌面文件，当桌面右键菜单中的自动整理选项被勾选，文件将会按照视频、音乐、图片、文档、应用、其他等类型分别存放，帮助您桌面图标排列井然有序。



### 壁纸轮换——美观大方，自出新意

多张壁纸自动轮换，当壁纸设置中自动更换壁纸选项被勾选时，用户可自行设定壁纸更换时间，为您的工作和生活带来时刻新鲜的视觉体验！



### 音效设置调整——音效开关，音随我动

对控制中心中的音效（如开机、关机、注销等）设置进行了详细划分，使用户可以通过音效设置控制某个的音效开关状态，使您对 dde 的掌控更加完善。



### 修复的缺陷与改进的功能

#### 控制中心：

- \* 新增屏幕旋转的超时还原的功能，避免黑屏后无法操作；
- \* 修复 2d 下屏幕旋转没有适配高分屏；
- \* 修复无法使用 ESC 键退出屏幕旋转界面；
- \* 新增网络 ipv4 的子网掩码的有效性检查；
- \* 优化无线列表的排序；
- \* 修复多屏缩放相关问题；
- \* 修复无限投屏相关问题；
- \* 修复显示模块在多屏环境下相关设置没有显示；
- \* 修复通知的音效示例效果和实际播放的效果不同；

#### 任务栏：

- \* 新增关机按钮右键菜单休眠选项；
- \* 新增回收站插件支持禁用；
- \* 在高效模式下 dock 右键菜单中不再显示回收站插件；
- \* 修复在 overlay 状态下没有通知提醒；

- \* 修复点击预览窗口后无法将其置顶；
- \* 修复预览窗口经过拖拽进入操作后无法再使用鼠标 hover；
- \* 修复在多屏环境下设置多屏缩放后 dock 位置不正确；
- \* 新增网络连通性提示错误；
- \* 插件相关配置由后端保存不再使用配置文件；
- \* 修复 stable 版本部分插件配置无法正确得保存到后端；

#### 启动器：

- \* 修复启动全屏模式时内容会突然向上；
- \* 修复全屏模式拖拽图标时图标过大的问题；
- \* 修复左侧分类列表的字号没有跟随系统字号调整；
- \* 修复任务栏在右边时，启动器小窗口的位置不正确；

#### dde-session-ui:

- \* 锁屏支持密码和指纹的同步验证；
- \* 修复多用户时没有指纹认证；
- \* 修复删除用户后切换用户界面显示不正确；
- \* 修复开启无密码登录后修改密码，锁屏界面无法进入系统；
- \* greeter 支持不同屏幕的缩放比；
- \* 其他 bug 修复。

#### 深度文件管理器

- \* 增加高级搜索功能，允许在搜索状态下对文件进行条件过滤；
- \* 修改地址栏 / 面包屑栏为新设计，以使用户拖拽标题栏；
- \* 面包屑项支持右键菜单“复制路径”和切换到“编辑地址”状态；
- \* 预览文件时支持 Ctrl+C 快捷键复制此文件路径到剪切板（预览图片时复制，会同时复制图片内容及路径）；
- \* 修复在重命名和创建文件时，输入文件名称完成后该文件未被选中的问题；
- \* 修复文件管理器开启磁盘索引后有时无法搜到文件的问题；
- \* 支持打开 ELF 类型文件时提示是否为其添加可执行权限；
- \* 修复 deepin-topbar 会覆盖桌面图标的问题；
- \* 减少文件管理器潜在的崩溃问题；

- \* 修复磁盘属性对话框显示文件数量不正确的问题；
- \* 修复最近文件中排序行为不正确的问题；
- \* 修复最近文件中双击打开文件，列表可能被显示为空的问题；
- \* 修复未知修改时间显示为空的问题。

#### 深度截图 & 深度截图 (Flatpak)

- \* 保存图片到桌面时检查桌面路径是否存在，若不存在则创建。

#### 深度看图 & 深度看图 (Flatpak)

- \* 修复在右下角的图片导航窗口上使用鼠标滚轮无法缩放图片的问题。
- \* 修复复制文件后无法粘贴到 windows 应用（如 QQ）；
- \* 打印预览的默认图像设置更改为“图像适应页面”；
- \* 默认使用彩色打印。

#### 深度编辑器

- \* 支持在状态栏显示文档的字符数量（包含空白字符，如换行符）；
- \* 修复 TAB 字符宽度不正确问题；
- \* 修复编辑器不支持左右分屏。
- \* 调整窗口的最小尺寸。

#### 深度系统监视器

- \* 修复启动 flatpak 应用时崩溃。

#### 深度录屏 & 深度录屏 (Flatpak)

- \* 修复开启缩放后无法在多屏幕下录屏。

#### 深度终端

- \* 增加了分屏时调整各个部分大小的快捷键（Ctrl+Alt+方向键）；
- \* 修复 kwin 下会失去焦点的问题；
- \* 修复终端在高分屏下菜单位置错误；
- \* 修复鼠标三击不能选中全行的问题；
- \* 支持“粗体即亮色”的选项配置（默认关闭）。



## 深度帮助手册

- \* 更新 15.10 相关手册。

## 网易云音乐

- \* 修复在 Ubuntu 系统上无法打开；
- \* 修复打开一段时间后会崩溃退出；
- \* 修复开启高分屏缩放后桌面歌词显示模糊；
- \* 修复无法在锁屏界面设置静音；
- \* 修复保存图片后应用退出；
- \* 支持使用本地文件选择对话框 (deepin 系统环境)；
- \* 修复登录窗口没有支持高分屏缩放。

## 已知但未修复的缺陷

- \* 使用 kwin 时，在触摸屏下无法拖动应用窗口；
- \* 深度商店的用户登录界面始终弹出的位置是在屏幕中间；
- \* 首次启动控制中心时，鼠标 hover 在通知图标上时，图标出现阴影；
- \* 在应用商店中安装软件时，切换到下载页面，页面载入缓慢；
- \* 使用 deepin-wm 窗管时，开启屏幕缩放后窗口标题栏菜单位置错误；
- \* 在 lightdm 中选定登录 kde 后注销，再次点击切换桌面环境按钮，显示的是 Deepin 默认选中。



## ISO 下载方式

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

## 鸣谢

感谢以下人员的支持（排名不分先后）：

Adam, Massimo C., Anna, Trompdoy, Kimmo Kujansuu, Cassiano Molinari, greg, Jason Evangelho, Kristien, P9A3N412, sonichy, SquareFong, SiamSami, youyou\_2608, aiamuzz

## 关于我们

深度操作系统是一款针对普通用户发行的开源 Linux，您可自由下载、分发、修改和使用。

欢迎您关注我们的微博、微信（深度操作系统）、Twitter、Facebook、Github 以第一时间获取最新动态和源代码，同时也欢迎您前往我们的社区论坛，与广大社区成员交流和分享您的快乐。

最后，我们郑重感谢为深度操作系统提供测试、文档、翻译和镜像支持的社区团队与企业，感谢你们的无私的贡献，开源有您更精彩。也要感谢一直支持、理解和等待我们的用户，是你们给了深度操作系统不断前行的动力，和不断自我修正的勇气。

## 下载地址：

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

## 全球排名：

<https://distrowatch.com/table.php?distribution=deepin> **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。

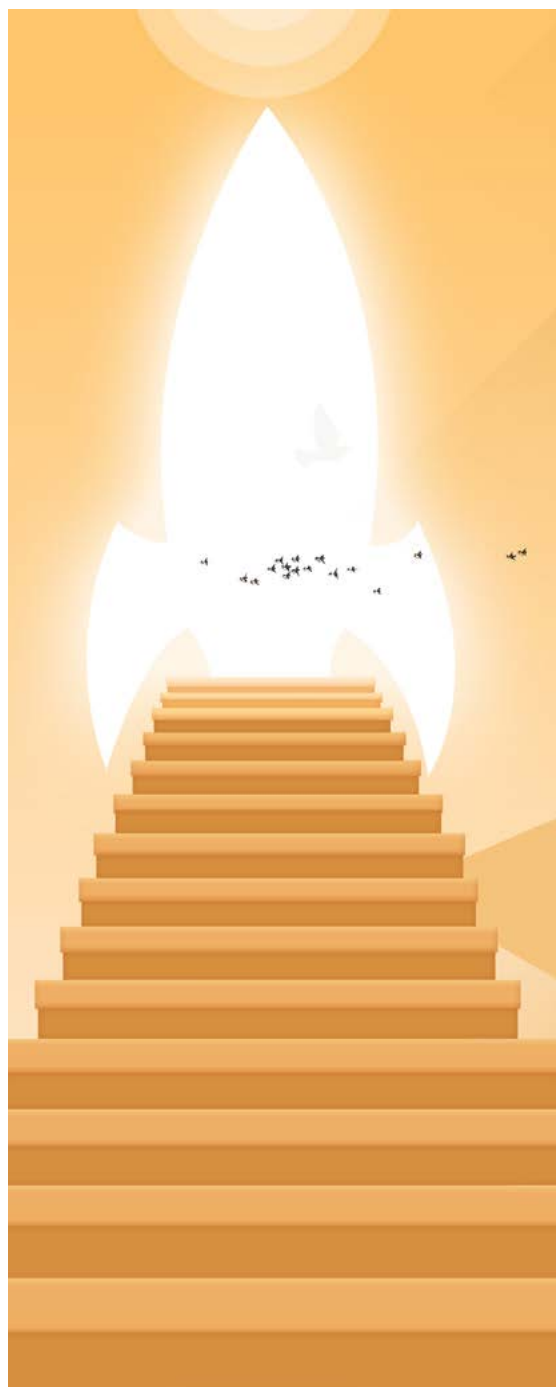
### 捐助定位：

- 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 ¥ 1687.14 已捐助 101 位支持者	 
 驱动中心 ¥ 6545 已捐助 338 位支持者	 
 Deepin Live 修复系统 ¥ 1193.66 已捐助 81 位支持者	 
 Deepin ID 云平台 ¥ 1115.65 已捐助 66 位支持者	 
 深度开发者平台 ¥ 3286.58 已捐助 193 位支持者	 



## 15.10 研发心得

### 新的旅程、新的开始

#### 先水点题外话

正如 @Hualet（深度科技 高级研发工程师）老前辈所言，我接了一口又大又厚的锅——deepin 桌面环境的所有“bug”。自然，最新系统发布之后写研发心得的优良传统也必须要传承下去！

我的昵称叫 @zccrs，说个大实话，看到这个词我也一脸懵逼，不知道该怎么读，索性大家可以叫我“榨(z)菜(c)炒(c)肉(r)丝(s)”，应该会更好记一些😂。

万事开头难啊，第一次写研发心得，居然有些莫名的紧张，另外还掺杂着些小激动。虽然都是敲键盘，但是和写代码时相比，这感觉完全是从二次元到了三次元。不过，幸好年龄大了，脸皮也厚了，写起东西来没以前那么费劲了。别的废话不说，先鼓励一下自己来点自信😁。跟 @Hualet 老哥相比，系统研发方面我还是个新手，虽不知能否像长江的“浪”一样把前浪“拍在沙滩上”，但至少比 @Hualet 长得帅些😁。欢迎大家以后多找我报 bug，多多益善，毕竟 bug 不挑食😁。

谁说写研发心得就非要苦思一个漂亮的开头，我直接就进入主题😁。

#### 进入正题

大家也都知道，从 15.8 开始，我们把项目都迁

移到了 github，方便更多大牛们参与进来贡献代码，也能通过 github 的 issue 提交和管理 bug。与此同时，为了更好的和用户互动，deepin 系统的社区版对外发布也采用了一种新的方式：滚动更新。由之前每三个月一次的系统大版本更新，改成了每三周一次的小版本更新，例如 15.8.1、15.8.2 ...15.8.4 等等。但是总体上来讲系统发布周期是没有变化的，最后一个小版本发布的同时也就是这个阶段大版本完成之时。这样做的好处呢，不用我多说，大家都有自己的感受，更新频率的增加，也就意味着能够更早的“吃到螃蟹”，想想就很开心 😊。

不过，这样做自然也有坏处，别的先不说，就拿大版本更新的发布来讲，新功能早就在小阶段更新中体验过了，大版本更新中完全没有了之前那种强烈的期待感，系统发布文章中描述更新内容时，会给人一种“这不是上个版本就已经有了的功能吗？”的错觉 😊。也就间接导致在研发心得中又少了一个介绍新功能的理由，没错，被大家发现了，说了那么多，我就是想吐槽一下研发心得里不能再介绍新功能了 😊。咳咳，开个玩笑，其实这次也没准备过多的介绍新功能，另外有一些非功能上的改变需要说一下，关于功能上的更新内容，大家可以移步到上篇文章查看详细说明。

### 基于 Debian stable 仓库 (By @hualet)

15.10 最大的变动要数仓库的切换了。从 unstable 切换到 stable，系统组的同学哼哧哼哧地干了两个多月，已然发布了，赞许的声音自然不少，但是不可否认的是社区中各种震惊的帖子也不罕见。我琢磨着，有时候他夜半惊醒，想到那些两三个月前满嘴“你们要是敢切到 stable，我就全盘格了 deepin”的同学们，只能暗自神伤：“这么坑

我，你们的良心真的不会痛么……”

玩笑归玩笑，经过这么一段时间的把玩和使用，我觉得这次仓库切换还是值得的。毕竟 stable 仓库可以说“人”如其名——行走江湖多年，就靠一个稳字。这也是我们切换到 stable 仓库的初衷了。除此之外，也很少出现以前那种装一个软件动不动就有可能出现依赖冲突等问题了。作为一个开发者，深感欣慰。

这时候有些“好事”的同学一定会问，商店里面的软件不够新怎么办？别着急，底层系统的稳定不代表上层应用不够新，我们已经为把商店应用更新准备了“激进”的方案，就差一个程 (da) 序 (bao) 员 (zhe) 了。

----- 硬广分割线 -----

为了满足 deepin 社区各位小伙伴对应用更新速度的极致需求，诚聘一名对开源和系统构建有足够兴趣的“准”大牛（到深度以后一定会成大牛的），有兴趣的请发简历到 hr@deepin.com。

----- 硬广交差线 -----

### DDE KWin 现在 & 将来

使用 dde-kwin 代替之前的 deepin-wm，这可能是这次系统发布后大家所争论最多的点。这个锅我背了 😊。

Updated 15.10 : SOS !!! ... How to switch back to deepin-wm ? #1121

如这个 issue 中所言，被吐槽最多的点就是



KWin 的界面。的确，我也觉得，它有些地方的确长的不够精致😓，像是一个👤。

DDE 环境中窗口管理器的界面主要分为五个部分：

- 窗口标题栏
- 屏幕热区触发
- Alt+Tab 任务切换
- Win+W 窗口切换
- Win+S 工作区切换

dde-kwin 的开发分成了两个阶段：

第一个阶段主要是在做一些看不到工作，增加对 DDE 环境的兼容性，例如支持窗管相关的所有快捷键设置、热区设置、窗口主题设置、窗口缩放跟随系统、窗管特效开关、工作区切换等功能，另外还包含了两个 deepin 风格的窗口标题栏主题插件。在 deepin 系统上，为了让 KWin 更好的“入乡随俗”，我们从 kwin fork 出了 deepin-kwin 项目，用于实现一些只能通过修改代码才能支持的功能，比如“\_GTK\_FRAME\_EXTENTS”窗口属性的支持。另外，因为其它发行版中没有 deepin-kwin，所以在修改 kwin 代码的同时，我们也会在 dde-kwin 中通过其它方式实现一个尽可能类似的效果，以保证 DDE 在其它系统上的兼容性。

第二个阶段则是着重处理 UI 方面的问题，KWin 和 deepin-wm 这两个项目有着本质上的差别，deepin-wm 中已完成的窗管界面无法直接在 KWin 中复用，一切工作都需要从头开始。不过，我们不会因此而妥协，为了带给大家一个🔥的 KWin，这一步无论多么艰难都要迈出去👊。😓一下子没把持住，是不是有点太煽情了(:手动斜眼:)😓。

额..... 刚刚才发现，说了那么多废话，居然没

有介绍为什么要那么青睐 KWin🐱，下面就要真的进入到正题了😓，这不是在为 KDE 背书，这不是在为 KDE 背书，这不是在为 KDE 背书——重要的事情先说三遍😓。

### 从产品发展的角度

在众多 X11 的窗口管理器项目中，毫无争议，KWin 绝对是最强大的存在，先丢一个官方的介绍。KWin 提供了“自定义窗口规则”、“窗口特效插件”等一系列扩展支持，以及强大的可配置性等等...但这一切还不是最重要的！接触过 Linux 的人都知道，生态环境是多么的重要（说多了都是泪😓），KWin 在窗口管理器的生态环境中有着绝对的优势，从 KDE 商店中可以看到。

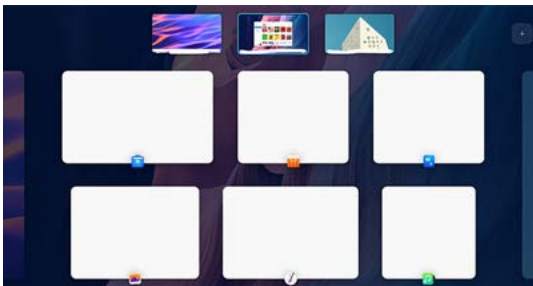
All	
Aurorae Themes	258
Beryl/Emerald Themes	1829
Cursors	919
deKorator Themes	172
Emoticons	129
Icon Themes	1305
KDE App-Addons	2147
KDE Game-Addons	104
Kvantum	53
<b>KWin</b>	<b>79</b>
Kwin Effects	11
Kwin Scripts	57
Kwin Switching Layouts	11

光 KWin 本身的插件就有将近 80 个，窗口标题栏主题 Aurora Themes 更是有几百个之多，各式各样的插件，能满足酸甜苦辣各种口味需求😓。另外，KWin 原生支持 2D/3D 模式切换，一个人能代替 deepin-wm 和 deepin-metacity 两大兄弟，也就意味着，使用 DDE 时，窗管再也不用装一送一了。

既然都已经把 KWin 表扬到这般地步了，那自

然也得再捅 deepin-wm 几刀 (@sonald 不要打我 😊)。从上游 fork 出 deepin-wm 之后，改了包名，而且所有的功能实现都是直接在原代码基础上进行的修改，导致现在的代码拉出来做 DNA 鉴定，都看不出和 mutter 有半毛钱关系 😊。时隔几年，deepin-wm 已经无法再跟随上游的脚步，同步不了最新代码，也就是说它现在已经变成了一潭死水，遇到新的产品需求时，在上面耗费了大量的时间也无法达到人民对它的期望。但也不可否认，deepin-wm 服役这些年获得了无数的好评，也得到了大家的认可。不过，“设想英雄垂暮日，温柔不住住何乡”，现在的确是到了它该“解甲归田”的时候了，江山代有才人出，接下来应该给 dde-kwin 一个证明自己的机会，我相信它一定能领风骚数百年！

deepin-wm 的 UI 已经定型，无论如何它就只能是目前这个样子了。但 KWin 它还可以是这样的：



甚至是这样的（千万不要被设计师看到，会被打死的 😊）：



### 从开发者角度


DDE 全家桶中，除了“深度终端”，其它所有的产品都是基于 Qt 开发，选择 KWin 自然也有这方面的理由。未来，我们希望窗管这个项目是 deepin 所有开发人员都能参与进去的，人多力量大，让这个项目一直都能有新鲜血液的注入，永远保持着年轻的活力。因此只有和 DDE 项目有着同样的技术基础，才能更好的帮助我们实现这个目标。

写到这个点眼睛都有点累了，稍微休息了一下，把前面写的捋了一遍。看完了关于 KWin 这部分后，我感觉说了那么多其实就概括为一句话：“俱往矣，数风流窗管，还看 KWin”。正如标题所言，这是一个新的旅程，deepin-wm 已经定格在了这里，但是，dde-kwin 的未来还有着无限的可能。

### 后记

书到用时方恨少啊，偷偷瞄了几眼前辈们写的“研发心得”，文体幽默而又不失风趣，心得感悟深而不腻，正好印证了一句话：姜还是老的辣啊 😊。想来想去，也不知道怎样才能写个漂亮的结尾，所以就到这里吧，省下点时间多修几个 bug 绝对没有错 🙌。

虽然 15.10 没有做到最好，但是我还是希望大家能够尝试一下，提出更多的问题。

宝剑锋从磨砺出，新的选择也将会打造出新的 deepin。 



# ExTiX Deepin 19.4 发布 基于深度操作系统 Deepin 15.9.3

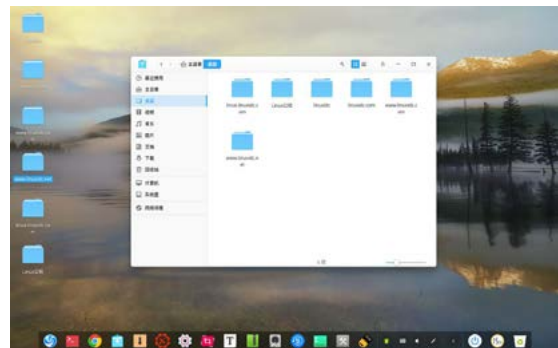
● Linux 公社 / 文

GNU/Linux 开发人员 Arne Exton 发布了他的 ExTiX GNU/Linux 发行版的新版本，该发行版基于来自中国的深度科技团队开发的 Deepin Linux 发行版。

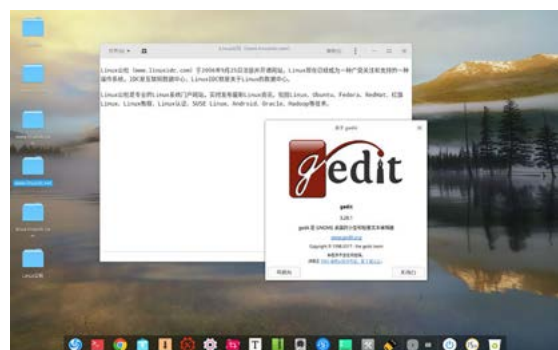
关于 ExTiX Deepin 19.4 发行版最大的消息是，它是第一个基于即将发布的 Deepin Linux 15.9.3 发行版的 GNU/Linux 发行版，该发行版目前处于 beta 开发阶段，尚未由 Deepin Technology 正式发布。



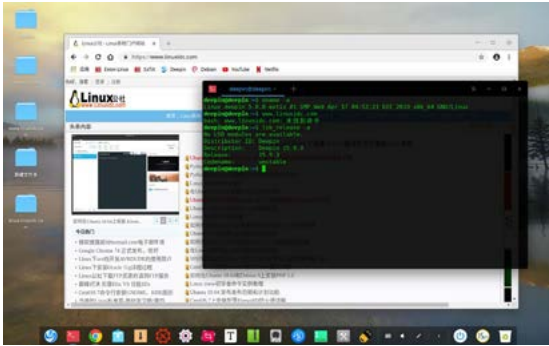
此外，ExTiX Deepin 19.4 附带了 Linux 5.0.8 内核，以获得最好的硬件支持，这使得 ExTiX 成为名副其实的“终极 Linux 系统”。对于那些想要制作自己的 ExTiX Deepin 19.4 live 系统，以及 Spotify 和 Skype 应用程序的用户，默认情况下也包括了最新的 Refracta Snapshot。



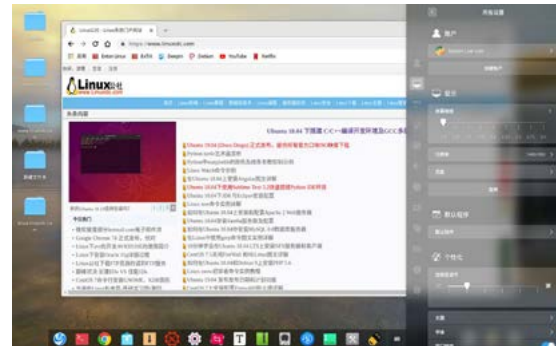
ExTiX Deepin 19.4 Build 190419 截图



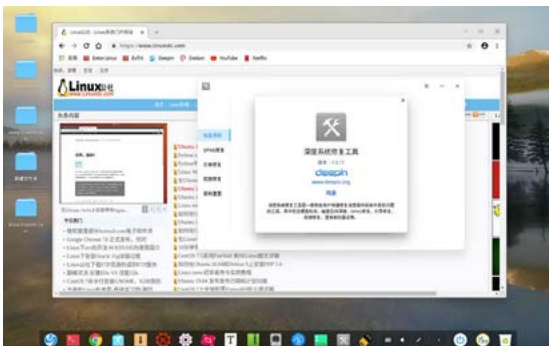
gedit 文本编辑器



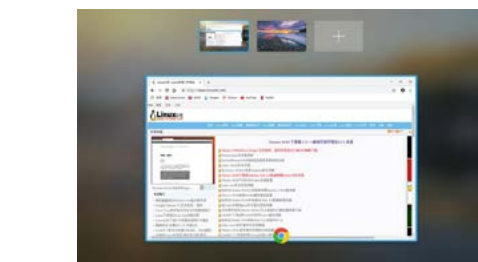
ExTiX Deepin 19.4 Build 190419 截图



ExTiX Deepin 19.4 Build 190419 截图

ExTiX Deepin 19.4 Build 190419 发布，基于 Deepin  
15.9.3 和 Linux 5.0

深度系统修复工具



ExTiX Deepin 19.4 Build 190419 多任务视图

Arne Exton 表示“多年来，我一直把自己不同版本的 ExTiX 称为‘终极 Linux 系统’。对于 ExTiX 19.4 Deepin，我觉得这是特别合理的。一切都很正常。”

使用 ExTiX Deepin 19.4，live ISO 映像也从 1.8GB 缩小到 1.5GB，这样您就可以更容易地从 RAM 运行 ExTiX，而无需在计算机上安装任何东西。此外，此版本附带了 Reborn 版本的 Deepin 安装程序，以及在进入活动会话之前选择语言的功能。

像往常一样，如果要在个人计算机上安装 ExTiX Deepin 19.4 Build 190419。如果您想知道 ExTiX Deepin 19.4 中的内容，查看已安装软件包的完整列表，请查看开发人员网站上的安装说明。<https://www.extix.se>



# Fedora 30 正式版发布： 引入 Deepin 桌面环境（DDE）

● OSCHINA 重庆大侠 / 文

Fedora 是一个由 Red Hat 赞助、Fedora 项目社区支持的独立 Linux 发行版。在 Fedora 30 这一版本中引入了 Deepin 桌面环境（DDE），以后可以选择 DeepinDE 体验 Fedora 了。

## 版本 / 变体

Fedora 拥有诸多版本（Editions），分别契合不同的特定应用场景。这一概念起始于 Fedora 21 版本，社区的需求在此后一直不断变化。在 Fedora 30 中，Fedora Server 版本针对云及服务器的应用场景。关注于容器的 Fedora Atomic Host 版本由 Fedora CoreOS 替代。Fedora Workstation 版本仍旧致力于带来最新的开源的桌面工具。

除了这些 editions 之外，Fedora 项目还通过 Spins（提供替代的桌面环境）和 Labs（特定领域的软件集合）覆盖了更多的不同需求或应用场景，其中也包含适用于 IoT 场景的版本。Fedora 也未曾忘记对其它架构的支持——在常用指令集架构之外，

Fedora 还支持 ARM AArch64、Power 及 S390x。

## 变化

此版本的 Fedora Workstation（工作站）版本搭载了 GNOME 3.32 桌面环境。GNOME 3.32 是 GNOME 桌面环境的最新版本，包含对视觉元素（如用户界面、图标及桌面本身）的更新。


Fedora Server 引入了 Linux System Roles 特性。这是一系列由 Ansible 执行的角色和模块，用以帮助 Linux 系统管理员配置常用的 GNU/Linux 子系统。

此版本还包含诸多软件包更新，其中较为显著的有：GCC 9、Bash 5.0、Fish 3.0 及 PHP 7.3。这一版本还引入了 Deepin 桌面环境（DDE）及 Pantheon 桌面环境。

## 新的论坛

此外，随着 Fedora 30 版本的发布，该项目正在将“Ask Fedora”支持论坛迁移至基于 Discourse 的新平台。这一新的论坛非常简明，且为包含简体中文在内的诸多非英语语言设有独立板块。

Fedora 主页：<https://fedoraproject.org/>

下载（请注意版本）：<https://getfedora.org/> 





# WPS Office 2019 For Linux 上架深度应用商店

● 深度科技 市场部 / 文

近日，WPS Office 2019 For Linux 新版本正式发布，版本号为 11.1.0.8372（以下简称 WPS 2019）。除延续 WPS 2016 版相同体验外，WPS 2019 界面更灵活、内容更精彩、视图更丰富、系统更兼容。

WPS 2019 较前一版本，进行了多方面的完善与优化，进一步提升了操作体验，更加符合国人的使用习惯。目前，WPS 2019 已经在深度应用商店上线，欢迎下载体验。



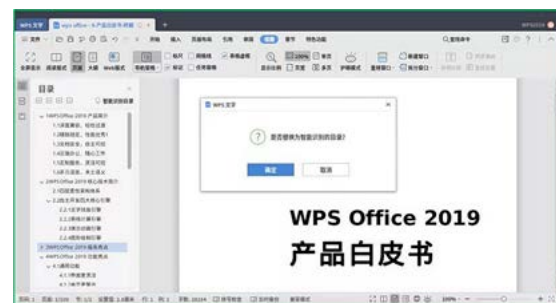
## 深度应用商店 一键安装 / 更新

在 WPS 2019 推出后，深度应用商店及时上线了最新版本，深度操作系统个人用户可以直接在应用商店进行安装或者更新，操作过程简单明了，一键完成。

## WPS 2019 亮点功能

WPS 2019 不仅支持全新的 2019 界面、高分屏、内置浏览器、方框打勾、导航窗格、智能填充等功能，同时优化了文字、演示、表格的整体性能，修复了批量的 bug。

1. 全新的新建页面，整合最近打开文档、本地模板、公文模板、在线模板等。
2. 支持智能目录更新。



3. 自动识别文档中的方框，支持打勾或取消打勾。
4. 支持多种判断规则，实现智能填充。
5. 支持单区域、多区域数据对比和提取；合并多个工作表或工作簿。
6. 支持 97 种智能图形的插入和编辑。

你心动了吗？赶快去深度应用商店安装吧！ [d](#)



## 深度科技参加 关键信息基础设施自主安全创新论坛

● 深度科技 市场部 / 文

### 重温 4·19 讲话

“互联网核心技术是我们最大的“命门”，核心技术受制于人是我们最大的隐患。”

“核心技术是国之重器，最关键最核心的技术要立足自主创新、自立自强。市场换不来核心技术，有钱也买不来核心技术，必须靠自己研发、自己发展。另一方面，我们强调自主创新，不是关起门来搞研发，一定要坚持开放创新，只有跟高手过招才知道差距，不能夜郎自大。”

“关键信息基础设施是经济社会运行的神经中

枢，是网络安全的重中之重，也是可能遭到重点攻击的目标。”

2019年4月19日，为纪念习近平总书记“4·19”讲话发表三周年，落实总书记指示精神，加快网信核心技术自主创新，《网信自主创新研究报告》编委会举办的第二届关键信息基础设施自主安全创新论坛在北京万寿宾馆举行。

来自网络安全和信息化主管部门领导，能源、交通、金融、医疗、军工等重要行业领域信息化主管领导，国产基础软硬件领军企业领导，国内优秀



网络安全企业代表等参与了本次会议。

论坛围绕习近平总书记关于网信工作的一系列讲话精神，探讨自主创新的网信技术和产品对关键信息基础设施保护的支撑作用，促进优秀产品和解决方案在重要行业领域的应用推广，会议同期发布了《2018 网信自主创新调研报告》。

深度科技参与了关键信息基础设施自主创新成果展，搭载深度操作系统的国产龙芯、申威笔记本在现场进行了展示。

深度桌面操作系统拥有自主研发的桌面环境，具有安全稳定、简单易用的特点，可提供基于 x86、龙芯、申威等硬件平台的操作系统替换方案。

深度操作系统与国产硬件平台进行了深层的优化适配，在日常办公场景下的用户体验与 Wintel 下

基本一致，可完全满足用户日常办公、学习和生活的应用需求。

深度科技作为国产操作系统生态的打造者，不但与各芯片、整机、中间件、数据库等厂商结成了紧密合作关系，还与 360、金山、网易、搜狗等企业联合开发了多款符合中国用户需求的应用软件。

深度科技的操作系统产品已通过了公安部安全操作系统认证、工信部国产操作系统适配认证、入围国管局中央集中采购名录，并在国内党政军、金融、运营商、教育等客户中得到了广泛应用。

核心技术是国之重器，最关键最核心的技术要立足自主创新。深度科技始终坚持自主研发与创新，秉持合作共享精神，愿与合作伙伴一道，共同为促进我国自主基础软件发展，提升关键信息基础设施安全保障贡献自己的力量。d



## 为爱而走 深度集结 用脚步为爱助力

● 深度科技 市场部 / 文

4月20日，第八届“为爱而走”在北京奥林匹克森林公园举行，以行走的力量为罕见病儿童发声，为教育公平助力。“为爱而走”奥森5KM慈善健走发起于2012年，每年春季在北京奥林匹克森林公园举行，是北京地区规模和影响力最大的跨国企业体育公益项目。

活动当天，虽然早晨飘着小雨，但是依然没有

阻止大家的热情，深度科技北京公司员工与 deepin 热爱者共 18 人参与了本次慈善健走活动。

出发前，主办方举办了血友之家儿童合唱团表演、爱心义卖等公益活动，将慈善健走活动推向高潮。

深度科技已经连续3年参与“为爱而走”慈善健走活动，以行走的力量为爱发声，希望更多的朋



友可以关注罕见病儿童，拥抱绿色健康的生活。

大家在健走的过程中不仅完成了公益活动，也加强了感情，成为了朋友。

最终，deepin 小队全部人员都完成了全程健走活动，前4名的小伙伴获得了纪念奖牌，每一个完

成全程健走的小伙伴都得到了一张捐赠证书。

5KM 很短，公益之路很长，每一个人都应有公益之心。deepin 将继续坚持公益之路，继续前进，希望更多的小伙伴参与进来。小小公益之火，可以燎原。d



## 深度科技参与广东省云浮市 “省市共建信息技术应用创新产业基地”工作

● 深度科技 市场部 / 文

为贯彻落实习近平总书记关于必须切实提高关键核心技术创新能力的指示精神，深入推动“省市共建信息技术应用创新产业基地”工作，推动信息技术应用创新产业加快发展，促使广东省信息技术应用创新等信息产业和数字经济发展持续走向全国前列。

4月26日，在广东药科大学云浮校区会议中心，广东省工业和信息化厅、云浮市人民政府联合举办了“云浮·云端·云产业，创新·自主·迎未来。省市

共同推进信息技术应用创新数字经济发展大会”。活动期间还举办了省市共同推进信息技术应用创新大会，信息技术应用创新及云计算大数据发展论坛，以及数字经济产业发展展示。

出席会议的有云浮市委书记、市人大常委会主任黄汉标，中国工程院院士倪光南，及工信部领导，信息技术应用创新领域专家，省工信厅，省直有关部门，各地市工信部门领导及有关企业代表等800多人。



深度科技作为国内优秀的国产操作系统代表之一参加了此次盛会，并参与了省市共建信息技术应用创新产业框架协议的颁布及合作伙伴框架协议的集体签约仪式。

### 盛会看点一

是我国广东省信息技术应用创新和数字经济发展的第一次专业盛会

本次盛会是深入贯彻落实党的十九大关于加快建设数字中国、网络强国的战略部署和习近平总书记关于“紧紧牵住核心技术自主创新这个‘牛鼻子’，抓紧突破网络发展的前沿技术和具有国际竞争力的关键核心技术”等重要指示精神，通过省市共同推进信息技术应用创新产业聚集，带动信息技术应用创新与数字经济融合发展，促进省市信息技术应用创新水平提升和技术应用创新办公终端产品普及应用。

### 盛会看点二

展示欠发达地区推动信息技术应用创新的有效模式

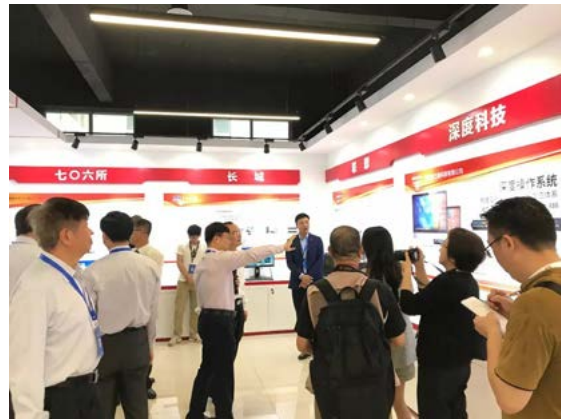
此次盛会，展示了云浮作为欠发达地区成功推

广信息技术应用创新办公终端应用的模式及“应用+研发+产业”的发展思路，在广东省工信厅等上级部门的关心支持下，云浮市于2015年承担基于信息技术应用创新软硬件的复杂办公系统应用研究与示范工程。经长时间的应用、试错、攻关、升级完善，系统稳定性和用户体验有了质的提升，由一开始的“不能用、不好用”进入“可以用、基本好用”阶段，2018年上半年，云浮市率先成为全国政务外网最大最好的信息技术创新终端应用场景，得到了国家工





深度科技总经理刘闻欢  
参与合作伙伴框架协议集体签约仪式



领导参观深度科技展台

信部、科技部、省工信厅等上级部门领导和专家学者的高度肯定，为广东省及全国铺开信息技术创新办公终端应用提供了成功经验。

### 盛会看点三

#### 打造信息技术项目合作新平台

在信息技术产业方面，广东省是全国经济强省，拥有高度发达的信息产业，云浮市是广东省内唯一承担国家信息技术应用创新项目和电子公文系统应用项目的地市，通过几年的应用推广，取得示范成效和先发地位。此次落成的云浮云计算数据中心，未来将成为华南地区水平领先的数据中心集群。同时，广东省工信厅也将和云浮市政府签署合作共建信息产业基地项目，该项目主要依托我市信息技术应用创新成果和产业基础，发挥省工信厅在政策、资金、人才和管理等资源和优势，深化信息技术应用创新合作，推动信息技术相关产业更多落户云浮发展。

深度科技将发挥自己在国产操作系统领域的优势，与广东省云浮市和各合作伙伴协同努力，助力国家级全国产化解决方案的顺利实施，促进产业优化升级，走出一条欠发达地区借助信息技术应用创新和数字经济产业实现跨越式发展的特色路子，并实现高质量发展。

在云浮云计算数据中心，深度科技展示了搭载龙芯 3A3000 的深度操作系统龙芯版软件，该操作系统是一款符合自主可控要求的国产操作系统软件，内置深度软件商店，WPS，360 安全浏览器等办公常用应用，通过云浮市省市共建信息技术应用创新产业基地工作的进行，深度操作系统将不断优化升级。

未来，深度科技将发挥自身优势，努力打造出更好的操作系统软件，积极扩大应用生态，与产业上下游合作伙伴共同努力，为实现关键核心技术安全可靠而努力。d



# 国内首款国产固态硬盘控制芯片发布

● 深度科技 市场部 / 文

4月，国科微与龙芯中科战略合作签约暨国内首款国产固态硬盘控制芯片发布仪式在北京人民大会堂举行。深度科技作为产业合作伙伴，受邀参加了此次活动，并与国内优秀的软硬件生态伙伴共同见证了“国内首款国产固态硬盘控制芯片发布会”。

作为国科微与龙芯中科的首个战略合作成果，国科微发布全新的GK2302系列芯片，搭载龙芯嵌入式CPU IP核，成为国内首款真正实现全国产化的固态硬盘控制芯片，为我国信息安全保驾护航。

来自工信部、中国工程院、安可工委、中国电子标准化研究院、国家集成电路产业投资基金股份有限公司等有关部门的领导和专家，代表国产化生态力量的企业家和技术专家，以及咨询机构和媒体等近200人齐聚一堂，一起见证我国在关键核心技术领域取得的这一重要成果。

本次发布的GK2302系列芯片采用高速SATA 6Gbps接口与主机通讯，单芯片容量最大支持4TB，读写速度达到500MB/s。相比上一代产品，GK2302 CPU性能提升15%，功耗降低6.5%，全盘性能提升18.4%，达到国际先进水平。未来，国科微将以搭载GK2302的固态硬盘形态为客户提供高性能、高安全、高可靠的存储解决方案。

作为首个全国产的固态硬盘控制芯片，GK2302系列重新定义了国产化的4个标准：第一，搭载国



产嵌入式CPU IP核；第二，从芯片设计到流片再到生产封装等各个环节全部在国内完成；第三，与国内整机品牌实现全面适配，第四，集成国密加解密算法，安全可靠。无疑，GK2302将成为推动国家网络信息安全建设的一项标志性成果，开启存储国产化的新里程。

核心技术，国之重器。今年的政府工作报告指出，加大基础研究和应用基础研究支持力度，强化原始创新，加强关键核心技术攻关。当前，人才、生态与时间已经成为制约我国关键核心技术取得快速突破的最重要因素。唯有企业间加强合作，实现产业链高效联动，兼收并蓄，才能加快推动核心技术的国产化之路，助力网络强国建设。

深度科技在与众多国内优秀的软硬件生态伙伴共同见证此次发布会的同时，也为国内首款国产固态硬盘控制芯片的诞生感到骄傲。深度科技也愿与更多国产软硬件厂商建立长期稳定的合作伙伴关系，并携手打造关键信息基础设施国产化生态建设，为国产化替代和国家信息安全贡献力量。d



## 世界知识产权日，深度科技在行动 加强知识产权建设 推动软件正版化

● 深度科技市场部 / 文

为庆祝第十九个世界知识产权日，中共云南省委宣传部、云南省版权局于4月25~26日在云南师范大学呈贡校区开展主题为“严格版权保护，推进创新发展”的版权集中宣传活动。

在云南师范大学呈贡校区贝壳广场，举行了版权执法、著作权社会服务、国产软件成果展示等活动。深度科技、金山、永中、360等优秀国产软件厂商向与会观众展示了最新研发成果。

深度桌面操作系统拥有自主研发的深度桌面环境（DDE），具有安全稳定、美观易用的特点，可

完全满足用户日常办公、学习和生活的应用需求。

活动现场，深度科技展出了搭载深度操作系统的龙芯桌面计算机，现场观众可以实际使用、体验深度操作系统。QQ、微信、360浏览器、WPS办公套件等常见的软件在深度操作系统上使用体验与Windows下面基本一致。

尤其是深度应用商店，可以实现常用软件的一键安装与卸载，普通用户无需担心与Windows系统的使用习惯不一致的问题。深度科技还与360、金山、网易、搜狗等国内知名企业联合开发了多款



符合中国用户需求的常用应用软件，深度科技自研的深度录屏、深度截图、深度影音等应用软件也深受大家喜欢。

### 打铁还需自身硬

深度科技作为一家专注基于 Linux 的国产操作系统研发与服务的商业公司，非常重视知识产权的建设与保护工作，在深度操作系统的研发和推广中，尊重各方的版权并严格遵守相关知识产权法律与协议。

2016 年，深度科技获得武汉市版权示范单位的称号，同年，深度科技作为优秀版权示范企业参加“第六届中国国际版权博览会”。

2018 年，武汉深之度科技有限公司入选“2018 年度中国版权最具影响力企业”，该奖项是中国版权协会设立的重要奖项，旨在鼓励在版权创造、运用、保护和管理等方面作出贡献的优秀企业。

### 推动软件正版化

深度科技除了自身高度重视知识产权建设与保护外，还积极地参与推动软件正版化的工作，并在湖北省正版化工作中取得优异成果。

2018 年，深度科技完成了“云梦县国家级正版化软件试点项目”的验收工作，深度科技在项目中为全县 48 个党政机关共 528 台电脑，安装国产操作系统软件 488 台，满足实施安装率 92.42%，深度操作系统优良的操作体验得到了云梦县党政机关试点用户的认可与肯定。

知识产权是一个企业乃至国家提高核心竞争力的战略资源，深度科技在加强自身知识产权建设和保护的同时，还将继续参与推动软件正版化工作，为我国信息安全和正版化工作的推进贡献力量。d



## 深度科技携手华为生态共建行动第一

● 深度科技 市场部 / 文

4月26日，深度科技受邀参加了华为智能计算大会，本次大会上，基于华为自研芯片鲲鹏920的TaiShan服务器解决方案正式发布，TaiShan解决方案可应用在大数据、分布式存储、ARM原生、高性能计算和数据库等领域。

### TaiShan ARM OS 使能

深度科技作为华为TaiShan平台的操作系统合作厂商，在TaiShan ARM服务器平台上已经完成相关适配工作，为华为ARM平台生态的建设贡献了深度力量。

在TaiShan技术分论坛上，深度科技技术总监张木梁发表《TaiShan ARM OS 使能》的演讲，介绍了深度科技基于TaiShan服务器开展的工作。

TaiShan服务器解决方案聚焦特定应用场景，充分发挥鲲鹏ARM处理器多核架构、高并发的计算优势，在安全性、低能耗方面更是有着明显的优势。

深度操作系统从2016年开始对ARM平台进



行支持，2018年3月支持 FT1500A，同年7月，完成华为鲲鹏 916 适配，2019年4月，针对鲲鹏 920 平台的通用版本进入终测，6月底将向用户发布。

目前深度科技已经为 TaiShan 硬件平台打造了全线操作系统软件产品：

#### ·深度操作系统 ARM 服务器版软件 v15.2

支持基于鲲鹏 916 的 TaiShan2280 系列服务器，提供合作伙伴和用户试用版下载，可升级为正式版。

#### ·深度操作系统 ARM 服务器版软件 v15.2(SP1)

支持基于鲲鹏 920 的 TaiShan 服务器，向上兼容鲲鹏 916，支持云环境，当前提供技术评估、测试支持。

#### ·深度操作系统 ARM 桌面版软件 v15.5(SP2)

支持基于鲲鹏 916 和鲲鹏 920 的硬件环境，同时支持运行与该硬件平台的虚拟化环境。目前只面向定向合作伙伴提供评估试用。

### 鲲鹏凌云 泰山聚势

华为鲲鹏 920 处理器采用 64 位 ARMv8.2 架构，64 核，7nm 工艺，主频 2.6GHz。

本次大会，华为分享了关于积极推进 ARM 产业发展与生态建设方面的努力，华为通过与合作伙伴紧密协作，完成了 TaiShan 服务器从基础硬件、主流操作系统、云平台及应用软件的使能，在特定应用场景上构建了全栈的 ARM 解决方案能力。

### 资源共享 开放合作 共建 TaiShan 生态

目前，深度科技已与达梦数据库、南大通用数



据库、金蝶中间件、金仓数据库、巨杉数据库、金山办公软件、华宇、万户 OA、360 企业安全、慧点科技等合作伙伴开展了适配工作。

深度科技全面开放适配支持，并协调组织“厂商 + 深度 + 华为”的三方兼容评测和认证工作。

深度科技将充分支持合作伙伴基于 TaiShan 服务器的软硬件生态适配工作，向合作伙伴提供远程开发资源和虚拟主机服务，提供支持 TaiShan 服务器的操作系统评估试用版下载，提供 ARM 桌面操作系统试用，合作伙伴可获得深度科技部分迁移、评估、验证后的组件、工具等支持服务。

此外，深度科技还与华为于今年 3 月联合主办了面向华为 TaiShan 服务器平台的深度软件开发大赛，也希望更多的开发者可以参与进来。（详见：“首届深度软件开发大赛”相关报道）

未来，深度科技将投入更多的力量全力支持“深度操作系统 + 华为 TaiShan”平台组合的生态建设，也希望更多的合作伙伴可以参与到适配工作中来，共同壮大 TaiShan 产业生态。d



“

虽然 deepin 有很大一部分用户都是懂技术的开发人员，但我们从一开始的产品定位就是站在普通用户的角度，让他们在使用过程中不会感到困扰，希望每一位没有计算机基础的人都会觉得 deepin 是一个非常简单好用的操作系统产品。

”



# 用户体验之 **上** - - 产品设计之 **下**

● 深度科技 产品经理 王珈 / 文

2014年的夏天，我接到 deepin 的面试通知。当时 deepin 办公地点在一个很偏的园区，有多偏呢？再往前走一点就上高速了。

除了会议室和会客室之外就只剩办公区了，办公区域很像网吧，大家一排一排的坐着，没有格子间，都连在一起，桌子上摆满各种机器和各种线。整个公司都是年轻人，大家的状态看起来比较轻松，而男生大多不修边幅，没什么拘束，每天就是对着电脑敲敲打打的。

在那之前我从事过三年多的产品工作，当时的

想法是希望加入一个有成熟产品团队的公司，能向其他人多学习。但面试官告诉我，我将是公司第一个专职的产品经理，这点和我的预期是挺相违背的。在此之前也从未想过会从事操作系统相关工作，它对我来说既庞大又陌生，很担心自己驾驭不了，抱着尝试的心态我告诉自己有机会那就试试吧。

你好，deepin

我清晰的记得，来 deepin 报道的第一天是个周四，正好在前一天举行了“2014 深度开发者与用户大会”的庆功宴，虽然我完美错过，但是可以感受到那天大家非常愉快还有点自豪。



第一次使用 deepin 时的样子 (deepin 2014)

我接手的第一个项目是深度影院 V2.1。刚开始交接工作的时候一个同事和我说：“播放器这些设置项大多数女生没兴趣的，如果你要做好这个产品经理，就要去了解市面上至少十款视频播放器。”他的话有刺激到我，所以那段时间我每天上下班回家都会体验各种播放器，使用它们的功能，做竞品分析，白天上班再来讨论我们的产品如何设计，版本如何规划，我们在不断探索适合自己的研发流程。

时间久了，新的产品逐渐增多，之前的沟通方式已经不太能支撑工作内容，于是我们开始编写完整的需求文档，每天早上开项目晨会对进度，完善版本发布的计划等等。一边学习别人的经验，一边寻找适合自己的方式，不断磨合，培养默契。

那时，我们的产品还处在填补空白的追逐期，用户更像是朋友一样包容我们的不足，给我们时间去创造，每一次发布都会收到很多的鼓励。

虽然 deepin 有很大一部分用户都是懂技术的开发人员，但我们从一开始的产品定位就是站在普

通用户的角度，让他们在使用过程中不会感到困扰，希望每一位没有计算机基础的人都会觉得 deepin 是一个非常简单好用的操作系统产品。

## deepin 的成人礼

2015 年 deepin 搬新家了，产品部门的同事也慢慢多了起来，有人一起商量讨论的感觉特别好。

刚搬家的那两年发布应用的频率相对凶猛，一年能有十几个发布。就拿深度商店来说吧，因为第一它是推翻 V3.0 从头规划的，看着它从无到有，一点一滴有很强的归属感；第二它和操作系统有非常紧密的联系，也让我对系统本身的认识有了质的飞跃；第三可能是我个人风格偏爱这种类型的产品，对它会比较有兴趣。

在 V4.0 深度商店的产品规划正式开始之前我和我的小伙伴们做了很多准备工作，深度商店除了产品研发本身之外，我们还需要投入大量的精力去运营。我们几个人分工逐个筛选软件，留下质量比较好的，确保新商店内容的高水准。重新定义分类，和启动器分类对应起来。筛选完成之后第一批入选的只有 200 多个。除此之外，持续筛选符合要求的应用，每个应用都需要有专人去做图标、封面图、写描述信息、截图、上架等工作，这些工作从那时起一直持续到现在。

需求和设计启动之后，在 V3.0 的审美环境下，V4.0 设计图一出就是整个操作系统最靓的仔，占了风头：首页轮播图会随着窗口大小变化展示不同模式；应用列表里再也不会会有水平、清晰度参差不齐的图标混在一起，所有图标、封面图都是颜值在线；详情页面每款应用都有多张高清截图，完整的功能介绍；用户的评论划分版本，参考当前版本的评论在不安装的情况下有大概的了解。



我们设计的一切都很美好，可实现起来却很头疼。V4.0的开发时间比计划的时间远远超了一倍，踩了很多坑。

轮播图效果非常炫，但是开发起来很花时间，直到发布之后还有一些小问题。当时安装过程在右上角有一个小圈圈在转动，测试的时候发现每当这个圈圈转起来就会影响帧数，鼠标划过封面图变得很卡，为了不继续延误发布，设计师向现实妥协换上了 gif 图，计划在下一个修复版本再换回来。

除此之外，最大的问题是深度商店的时区选择问题。因为那时深度商店还没有推出自定义选区，我们设定通过‘时区选择’来自动判断商店的区域，而那时，‘控制中心’中的‘选择时区’为了让用户觉得简洁并能更快地找到自己所在时区，只留下了几个最常见的城市，这样虽然用户选择的可能不是自己的城市，但是并不影响他的时区，可是和商店选区联系在一起后导致所有东八区的国家使用商店时全是中国区。

像这样大大小小的问题在每一次开发过程中都会遇到，解决这些问题就像升级打怪一样积累经验。后来 V5.0 的轮播图我们改变了策略，兼具了效果和性能，下载安装列表更明显更便于管理，时区的选择也改成了地图模式，至于商店如何选区在 2019 年的我们又设计了新的方案，应该不久大家就可以体验到了。

### 至高无上的用户体验

在 deepin 深度操作系统 V15 版本发布之后，我们要静下心来想想未来的方向，到底要给用户带来什么？我的小伙伴们在白板前思考、讨论、找灵感很多次之后，得出了一些关键字：

·我们的系统想呈现给人的感觉？

惊艳、轻盈、流畅、清爽、与众不同…

·我们想传递什么理念给用户？

发掘好奇心…

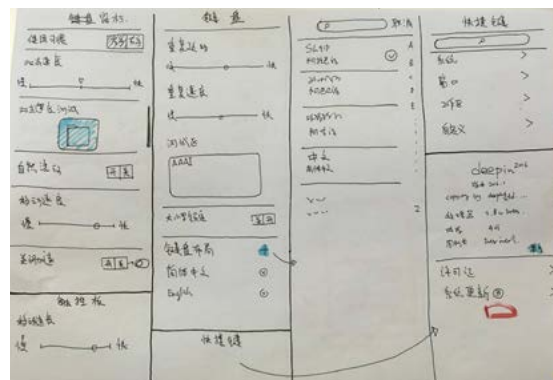
·我们的设计和交互原则？

简单、优美、内容至上、心理预期、触摸情感…

·动效的重要性？

引导性、节奏感、趣味性、物理特性…

### 深度控制中心



深度操作系统 V15.5 控制中心手稿



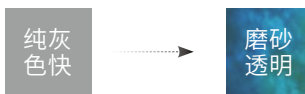
深度操作系统 V15.5 控制中心设计图



深度操作系统 V15.5 控制中心设计图



深度操作系统 V15.8 控制中心设计图



深度控制中心的变化

如果是关注 deepin 好几年的老用户应该知道，控制中心这几年经过了几次比较大的改动。

在以前很长一段时间，控制中心是一整块灰色背景，在我们锁定了“轻盈”“清爽”这些关键字后，深度操作系统 V15.5 的版本里设计师把它调成了磨砂质感，使视觉上更有穿透性。鼠标双击测试增加了小浣熊出没，增加了趣味性，算是一枚彩蛋吧。

到了深度操作系统 V15.8 的设计阶段我们再一次重新思考了控制中心，这个过程中出了很多方案和各种绚丽的设计。最终多次比对后我们选择拿掉和弱化了一些功能，比如天气预报、音乐控制、wifi、蓝牙、vpn、通知信息。虽然有的功能很有用，但是在其他平台或者模块里操作可能会更加方便，留一个纯粹的控制中心是我们在 15.8 做出改动的结论。

### 深度启动器

深度启动器在第一次推出小窗口模式时，用户反响非常不错，因为小窗口的启动器可以在不刷掉整个桌面的情况下启动应用。

按照惯例，我们在推新功能时一般会先推出一个基础版本，发出后会根据用户反馈的情况追加几个强化版本。在小窗口模式的优化过程中，我们希望启动器可以和文件管理器、控制中心、电源快捷的关联起来，这部分没有太多分歧，只是考虑怎样布局更加美观。

顾虑较深的主要是其他几个问题：

- 是否要在有限的区域内显示应用的分类信息？
- 新安装的应用如何有效、明显的提示用户在哪里可以找到？

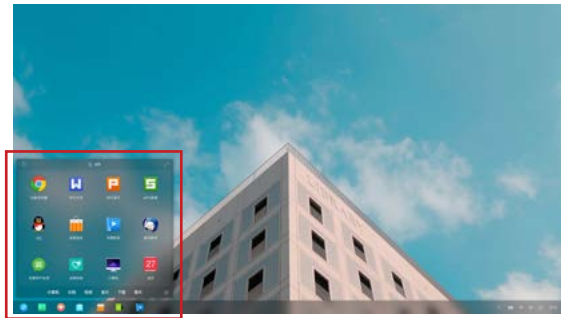
深度启动器中‘新安装应用’增加背景色（如图：改进方案三）的方案首先被 pass 了，不符合深度操作系统的整体风格，在系统内别的地方也没有这样的运用，而且整块区域有些过分强调。

后面出现了五角星（如图：改进方案四）的方案：新安装的应用都显示在最上面，把使用频率高的应用挤下去，当用户点击 n 次以后五角星消失。

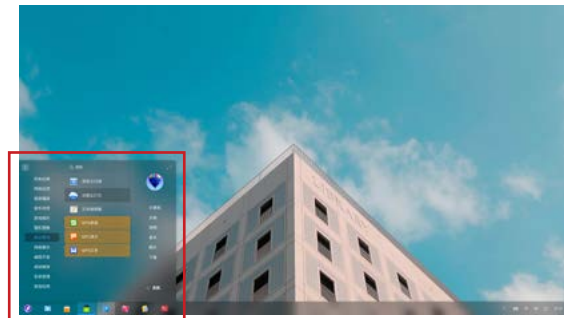
那么，为什么这么和新安装的应用较劲呢？因为有专业版用户提出在小窗口模式中不能快速找到新安装的应用。

在这个方案几乎就要开始做的时候又抛出一个疑问：这样最新安装的应用确实非常显眼，但是在点击 n 次之后呢？它从最上方消失了，用户要去哪里找到他？我们要做的本质其实不是简单粗暴的告诉他：“快看！你要找的就在这里。”而是引导他学会这样一个规律——以后新安装的应用都可以通过这种方式找到（如图：改进方案五）。

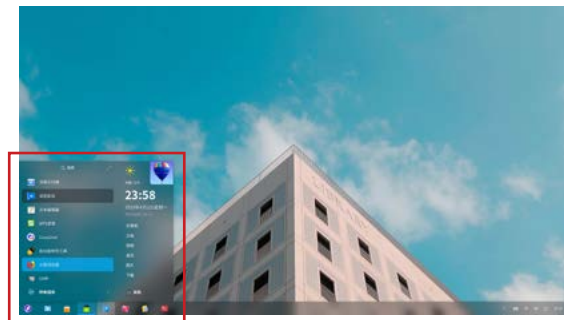
后来经过了 2~3 轮更新后，小窗口模式变得逐渐完善，这些问题也都有了相对一致的方案（如图：发布的版本）。除此之外，使用频率的算法上我们也在不断让 deepin 变得更加聪明，从以前的单一统计点击次数，进化到平均使用次数， $\text{平均使用次数} = \text{总点击次数} / \text{安装应用起到此刻的总时长}$ 。



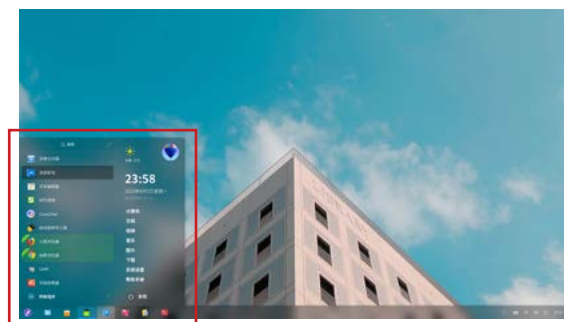
启动器小窗口模式第一次发布



改进方案一



改进方案二



改进方案三



改进方案四



改进方案五



发布的版本

用户体验不仅局限于产品的功能、设计，一切能和用户产生交互的行为都是我们关注的重点：

- 帮助手册的内容是否能解决用户的疑问？
- 用户反馈的问题是否能及时响应？
- 系统内的提示文案对用户的操作是否有帮助？

## 多语言支持

一直以来我们会陆续收到公司内部、外部提出的希望英文翻译准确性能有更大改善，也有不少热心用户主动申请帮助我们提高英文翻译质量。

深度操作系统目前除了英语还支持 32 种其他国家的语言，而这 32 种语言被翻译前的源语言是英文，各国翻译志愿者都是按照英文来翻译成本国语言的，因此英文的准确性、专业性尤为重要。所以在今年年初，我们组建了社区翻译志愿者团队，里面都是英语为母语的活跃用户，我们也设计了一套机制借助他们的力量帮助负责翻译的同事完善英文文案，希望各个国家的用户都能无障碍的使用 deepin。

在国际化方面，我们尤其要感谢同一个地球生活在各个国家的志愿者朋友们，是大家点点滴滴的贡献，使 deepin 深度操作系统迈出国门，并得到世界各国 Linux 爱好者的喜爱和追捧。

## 更多认可，更多努力

如今，deepin 得到了来自世界各地用户的支持和认可，这离不开 deepiner 的执着和坚持。

每一个新功能的发布都需要有多个备选方案，设计师要做很多的不同的设计图供大家讨论、筛选，再去改进，尽管有些永远都不会被大家看到。

思考的过程很烧脑，但幸运的是我们非常有默契，有一致的立场，很多时候一两句话就能明白对方的意思。有分歧的时候，也会激烈的说服对方，但肯定会无条件的信任对方的专业能力。

开发则是一群优秀的小伙子，他们不局限于只

是做安排好的开发任务，当对需求有异议的时候会非常认真的来争取，和我说明为什么觉得不好，给予我很多宝贵的意见。

测试的同事也经常提出我们在需求设计阶段没有考虑到的问题，让我们做到尽早避坑，让开发少走弯路。有时候会遇到因为不了解技术工作卡壳的情况，他们会非常耐心的和我讲解技术原理，帮助我做出正确的判断。

此处不得不插个小故事：有一次我们要做文件管理器打开目录的速度优化，开发的小伙伴说文件排序会很影响加载速度，提议能不能考虑三秒内可以加载的文件就正常排序，三秒后继续加载的文件不进行排序直接在最后依次添加。因为普通台式机三秒内加载的数量已经是万为单位的级别，当超过几万个文件在同一个文件夹的时候用排序去找文件已经几乎是不可能的事情，用的是更多的是搜索。

像上面列举的小故事那样，他们不仅帮忙能找到问题，还能够主动给出解决方案。这样的情况很常见，小伙伴们总是能从他们的角度提出非常有意义的改进意见，大家对 deepin 承担的责任已经超出了自己的岗位。

### deepin, 未来可期

deepin 深度操作系统要同时兼顾社区版和专业版的用户，用户群则在年龄、使用习惯、对计算机了解程度上都有很大差异，我们会尽最大可能满足用户们的需求，不会忘记“为每个普通用户做操作系统”是我们的初心。

每当我们推出一个功能时，内心都是忐忑的，



一部分用户喜欢肯定也会有另一部分用户不喜欢。不管是好的声音还是坏的声音，用户的反馈都是我们宝贵的财富。

这里需要说明的是，用户反馈每天都有专门负责的小伙伴去看，大家会把有效建议搜集起来做优先级划分，有时某些问题响应慢好几个版本没解决并不是我们把用户反馈的问题抛在脑后，而是因为操作系统太庞大了，有限的资源只能优先完成最为紧要的任务，所以大家有任何建议尽管提吧，说不定下一个版本鸣谢名单里就会出现你的名字。

今年我们已经开始深度操作系统 V20 的准备工作了，除了性能、功能的优化之外，还有全新的界面风格大改版。为了保护设计师的劳动果实我只能忍住先不发设计图，但是还是希望大家拭目以待！ **d**



## 《中国正在说》 听倪光南院士说说我国操作系统与网络安全

● 深度科技 市场部 / 文



《中国正在说》本期主讲人：倪光南院士

5月24日，东南卫视《中国正在说》播出了由倪光南院士主讲的关于我国网络安全与信息化技术应用创新的节目。

深度科技总经理刘闻欢、龙芯中科副总经理张戈、《信息安全与通信保密》副总编辑唐莉、中国科学院信息工程研究所博士生研究生陈钧莎做客演播室，与倪光南院士共议网络安全与信息化。

从之前美国对中兴的制裁，到近期美国全力对华为的全方位封锁，我们可以看出，网络安全与信



武汉深之度科技有限公司 总经理 刘闻欢

息化领域的体系建设情况，也代表着一个国家的尊严、独立和自由。

网络信息产业中，芯片被比喻为心脏，操作系统被比喻为灵魂。操作系统在信息产业中的重要性不言而喻。我国操作系统的发展一直备受关注。

### 我国操作系统的发展难在哪？

倪光南院士认为：“突破缺芯少魂的困境有三大关键点，设计、制造和生态，要追赶世界先进水平，设计相对容易，制造比较难，而生态问题则是重中之重。生态是围绕 CPU 和操作系统的软硬件群落。缺少配套的软硬件，CPU 和操作系统就难有用武之地。也不利于功能的持续完善。作为追赶者，我国 CPU 和操作系统，既要突破 wintel 体系在桌面端的垄断地位，还要突破安卓和苹果在移动端的生态垄断。”

武汉深之度科技有限公司总经理刘闻欢认为：“网络安全与信息化不是个人或是一家企业就能做的，而是需要整个国家有更多的人才和企业共同努力，贡献知识力量也好，打造生态也好，用行动来累积。而我们这些已经投入到我国信息技术应用创

新中的企业，则需要严把质量关，做真正能用可用的产品，给用户信心，给国家信心。”

现场观众提问：

感觉我国信息技术发展很困难，为什么会这么难？

针对这个问题，作为现场嘉宾的深度操作系统创始人刘闻欢表示，操作系统是一个超巨型的工程，有两个数据：第一个是美国微软公司在开发 Win7 时，微软在里面长期投入 1000 人左右，做了四年的时间才把 Windows7 开发出来，这还不算他们以前做 Windows95 等操作系统的积累。

第二个数据是我们现在很多操作系统企业，正在基于一个开源 Linux 内核做的相关的操作系统产品。一个完整的通用的基于 Linux 的操作系统它的源代码行数已经达到 1 亿 2 千万行，其中内核就这块就有 2000 万行以上。





而 Linux 内核在过去的二十多年中有将近 2 万名开发人员在里面做过贡献，有数百家企业也在里面提供了贡献。不管从他的投入来说，它的代码行数来说，还是参与者的人数来说，都是非常巨大的一个软件产品。

所以说做操作系统，它的确是一个非常困难的事情，它的投入大，周期长，而且后面还有着生态的挑战。

### 从不可用向可用、好用迈进！

桌面计算机技术体系已达到“可用”

倪光南院士说，无论是能源、金融、交通、通信、教育等领域电脑安全特别重要，在政府办公的层面，也非常重要，根据工信部的评估，我们的桌面计算机技术已经从不可用到可用了，而且正在向可用、

好用发展。

我国操作系统及 CPU 等基础软硬件经过这么多年的发展，已经达到了可用的阶段，龙芯、申威等 CPU 厂商以及深度科技、中标麒麟、普华等操作系统厂商，已经运用在党政军、金融等重要行业，从实际使用效果看，经过深度磨合优化的 CPU 和操作系统在办公场景下已经基本和 Wintel 上的用户体验一致，有些操作体验上可能还更加符合国人的使用习惯，我国操作系统与 CPU 整体性能与体验已经达到了“可用”，正在向“好用”迈进。

### 深度操作系统现场实测

深度操作系统作为我国操作系统的代表在节目现场进行了演示，让大家直观的感受深度操作系



现场微信视频通话





常用场景演示



统良好的用户体验。

节目现场，主持人使用手机与深度操作系统工作人员进行了微信视频通话演示，全程沟通顺畅，链接无碍。

接下来，现场还使用深度操作系统演示了网上购物、游戏操作、视频演示、QQ 登录聊天等日常工作生活中的常用软件，深度操作系统不仅界面美观，易用程度很高，软件也日趋完善。倪院士表示：深度这个系统还是很友好的。

在倪光南院士看来，如今，在网信领域形成自己的核心技术的观念，比以往任何时候都要强烈。网信领域发生的事件对当前的中国来说是一个机遇，我们能够抓住这个机遇，带动我们的技术，带动我们的产业，更好地发展，尽快地从跟跑到并跑，再发展到领跑。d

# 绝大多数开源软件代码 不受美国出口限制

最近在开源界和基础硬件行业，华为被列入实体名单，谷歌宣布安卓手机操作系统对华为断供成为被讨论的热点，下面这样几篇文章流传很广：

《开源界也要注意，Apache 基金会与 GitHub 都受美国法律约束》

<https://www.oschina.net/news/106836/opensource-ourself>

《冷汗：开源软件也受美国出口管制》

[https://mp.weixin.qq.com/s/XV\\_35Qkw1hKmxY8R17OCuA](https://mp.weixin.qq.com/s/XV_35Qkw1hKmxY8R17OCuA)

《所以 Apache 基金会不受美国法律约束？》

<https://www.oschina.net/news/106863/what-about-ear-and-asf>

《专家解读：开源软件项目是否会被限制出口》

[https://mp.weixin.qq.com/s/\\_wmBHskWi5CCTjDWu6fMAG](https://mp.weixin.qq.com/s/_wmBHskWi5CCTjDWu6fMAG)

包云岗的微博文章

<https://weibo.com/1686707751/Hv7D1jCAG?type=comment>

《开源是自由的，永远》

<https://tech.sina.com.cn/cs/2019-05-22/doc-ihvhiews3667241.shtml>

为了节约大家的时间，我把上面这几篇文章的核心观点做一下整理：引起大家关注的是 apache 基金会和 github.com 的授权与出口协议，表明其可能会受到美国政府相关法律的要求，限制其产品

和代码出口。

上面的几篇文章在观点上各有不同，对开源软件被禁止出口比较担忧的观点包括：

1. 基于 Apache 协议和托管在 Apache 基金会下的开源软件，受到美国出口限制。

2. 三个主要代码托管平台：github、SourceForge、Google Code 上的代码，无论是否开源，均明确声明遵守美国出口管制条例，并按加州法律解决纠纷，我们很可能无法获得其上全部代码的使用权或者产权。

3. Linux 基金会注册在美国，Linux 内核代码托管在美国服务器上，因此也可能受到美国出口管制

在更多开源界和法律界人士参与讨论后，又有如下分析：

1. 大多数采用 Apache 授权的开源软件，并不属于 Apache 基金会。托管在 apache 基金会的开源软件，才受到出口限制，例如 hadoop，spark，但这个出口限制，也仅限于在其中有需要审查的加解密技术的软件。

2. Github 企业版是 Github 公司的一个产品，而且是一个闭源的产品，这个产品不属于「公开可及 ( Publicly available )」的范围，因此会受到 EAR 管制。确实可能存在无法出口的问题。但实际上，我们完全可以不使用 github 的企业版本，而

且我们也不需要这个产品的源代码。

3. GitHub 的 ToS 只声明 GitHub 的服务本身受美国出口管制，也就是说一个被美国政府限制出口的国家可能无法使用 GitHub 网站，而不是不允许使用托管在 github 上的开源代码。

4. 开源软件中的“开源”和“软件”是两个概念。软件源代码是一种言论自由，并且受美国宪法第一修正案保护，美国政府无权限制软件源代码的发布行为，也不能限制其出口。无论任何许可证，GPL 还是 Apache 或是 MIT，采用它们的代码都不会受到美国管制法律影响。但是分发 -- 尤其是分发编译打包之后的成品，那个叫做软件，就脱离了第一修正案保护的范畴，和大部分真实商品一样，适用于所有管制法律。

5月22日，apache 官方博客发布了如图说明。

该声明同时得到了 Linux 基金会的法律支持，  
<https://blogs.apache.org/foundation/entry/statement-by-the-apache-software>

OSCHINA 发布了一条新的解读：  
<https://www.oschina.net/news/106904/statement-by-the-apache-software>

其核心观点与之前《专家解读：开源软件项目是否会被限制出口》非常接近：

- 依 美 国 EAR (Export Administration Regulation, EAR), 美国人、美国公司将软件出口至美国境外，或在美国境内提供给外国人作为出口的预备行为，必须申请取得许可。

- 但符合「公开可及 (Publicly available)」

## The Apache Software Foundation Blog

= The Apache Software... | Main

WEDNESDAY MAY 22, 2019

Statement by The Apache Software Foundation regarding US Federal Register Notice of non-US affiliates added to Entity List Ruling

Restrictions on exports and reexports to parties named on Entity List specifically apply to activities and transactions subject to the Export Administration Regulation (EAR). [1] Open Source publicly available encryption software source code, as reclassified by the US Department of Commerce, Bureau of Industry and Security (BIS) effective September 20, 2016, is "publicly available" and "published" and is not "subject to the EAR." [2]

Open Source projects involving encryption software source code are still required to send a notice of the URL to BIS and NSA to satisfy the "publicly available" notice requirement in EAR § 742.15(b).

The ASF continues to work with Apache projects and their communities to ensure their notices are up to date and are maintained in the future.[3]

Open Source software, collaboration on Open Source code, attending open telephonic or in person meetings, and providing sponsorship funds are all activities that are not subject to the EAR and therefore should have no impact on our communities.

For more information, visit <http://apache.org/foundation/license-faq.html>

Roman Shaposhnik  
ASF Vice President Legal Affairs

We thank DLA Piper and The Linux Foundation for their legal counsel and collaboration regarding this subject.

[1] <https://www.bis.doc.gov/index.php/documents/regulations-docs/2395-effective-date-of-huawei-and-affiliates-entity-list-rule>

[2] 81 Fed. Reg. 64656, 64668 (September 20, 2016). See also, <https://www.bis.doc.gov/index.php/policy-guidance/encryption/223-new-encryption>

[3] <https://www.apache.org/licenses/exports/>

Posted at 09:59PM May 22, 2019 by Sally in General | [f](#) [t](#)

定义的软件，不在 EAR 管制范围，亦即不需要申请许可；也就是说，多数的开源软件，皆为公开可及并能后续散布，符合这条但书，出口上不需要申请许可。(EAR 734.7 (a))

- 但 EAR 734.7 (b) 同时说明，公开可及软件虽不需许可，但若涉及加解密技术，仍然必须申请许可。

- 除非是这个加解密技术，除了代码本身公开可及外，其加密方式原则上也是公开可及的，那



就可以再主张它在 ECCN 5D002 的列表里，可以采取 EAR742.15(b) 款提供源代码或揭露源代码来源的方式，来登录备查。

一句话总结：即使按照美国法律，除非使用了非公开领域的加解密技术，基于开源协议发布的代码均不受出口管制。

可能还有人会思考，当前虽然没有问题，但是难保以后美国不耍流氓。什么时候增加或者修改一项法律，开源代码也属于限制范围怎么办？这种情况在极端情况下可以关注，但不用过多担心，原因有如下几点：

1. 如果美国修改规定所有托管在美国境内服务器的代码，均受美国政府出口限制。由于代码本身的传播相当方便，那么大量的开源项目，将不会再托管在美国，很快转移到欧洲，中国等地区进行多托管平台的情况一定会发生。我们不用担心无法合法获得源代码。

2. 要修改相关的规定和法律，将会与之前美

国第六巡回上诉法庭和第九巡回上诉法庭的判决冲突：软件源代码是言论自由，受宪法第一修正案的保护。要修改这样的法律，尤其是涉及到美国宪法第一修正案的判例，是一场旷日持久，影响巨大的事件，修改通过的可能性不大。

3. 如果美国真的完成修改法律，暂不考虑其他各国和整个产业界的反应，大量非美国的开源代码和知识产权，甚至包括中国开发的开源代码和知识产权因此被强行管辖，那么保护美国知识产权本身也就没有了基础。到了这样的阶段，传统的知识产权体系基本陷于崩溃，也没必要考虑去尊重美国的国内法和其知识产权的限制。只要我们自己具备能力，相关代码拿过来就用好了。

这次事件在经过各方讨论，尤其是在 apache 基金会、linux 基金会关注和澄清之后，使得行业人士和主管机关，对开源软件、开源软件授权的认识会更加清晰，这对整个软件行业和开源社区的发展非常有益。我们也希望有更多的专业人士关注开源软件技术，积极参与开源社区的工作，建立更多中国自己领导的的开源社区和产品。d

# 360 压缩火力全开 全面支持 Linux 操作系统!

近日，360 压缩在现有功能的基础上已完成针对 Linux 操作系统的兼容性测试，在众多压缩软件中率先实现对中标麒麟、银河麒麟、深度等操作系统，以及龙芯、兆芯、飞腾等 CPU 的支持，全力为主流计算平台提供压缩服务，这也是继 360 浏览器后，360 旗下产品迈向开发者生态的又一重要举措。

## 压缩软件缺乏统一标准及规范

### 360 压缩强势“破局” 打造应用新生态

当前主流计算平台、操作系统、CPU、软件应用的厂商众多，各平台对压缩软件的适配、兼容性程度不一，缺乏统一的标准与规范，不同平台使用压缩软件的操作方式颇有不同，这大大提升了软件的使用门槛，导致多数用户不习惯系统自带的压缩软件。

面对这一困境，360 压缩自 2011 年发布以来，在延续用户习惯的基础上专注于提升用户体验，整合 360 安全能力不断开拓创新，提高操作系统的便捷性、易用性。如今已开创内存扫描，图片预览等广受用户好评的功能，并支持解压主流的 rar、zip、7z、iso 等多达 42 种压缩文件，同时改进了超过 20 项使用细节，大幅简化了传统软件的繁琐操作，以更快更轻巧、兼容性更好、更安全更漂亮、而且终身免费等诸多优势，得到广大用户的一致好评，在市场占有率上持续提高。

## 基于 360 安全能力 打磨六大压缩优势 全面赋能 Linux 操作系统

下面具体了解下 360 这位全力支持 Linux 操作

系统的压缩软件的强大功能优势。

### 优势一：极速智能

能智能地为图片、视频、文档等内容挑选最优的压缩方式，在不牺牲压缩率的情况下，达到压缩速度比传统压缩软件快 40% 的成效。

### 优势二：简洁易用

大幅简化传统压缩软件的繁琐设置，专门为用户配置压缩参数、压缩速度和压缩率，且能让用户随心选择，想怎么压就怎么压。

### 优势三：兼容性好

支持更多的压缩格式，只需安装一款压缩软件，即可轻松解压多达 42 种压缩格式。

### 优势四：云安全引擎


由于很多木马藏在压缩包内进行传播，窃取用户隐私。所以，360 压缩内置了云安全引擎，打开压缩包的同时即可完成自动扫描木马，保证使用者的安全!

### 优势五：RAR 风格主题

双主题快速切换，一键切换到 RAR 风格主题，压缩、解压等一切操作都真实还原 WinRAR，适合喜欢 WinRAR 操作习惯的用户。

### 优势六：永久免费

再也不用担心传统压缩的共享版、40 天试用期、购买许可、破解版、修正版这些问题，360 压缩将会永久免费!

未来，360 公司将积极主动参与国内计算平台的生态及安全建设，竭尽全力支持 Linux 操作系统，从而更好地保证政企及广大用户的网络安全! 



## 2019 深度软件开发大赛 决赛入围名单公布!

6月28日,在“第二十三届中国国际软件博览会”平行论坛中深度科技副总经理公布了2019首届深度软件大赛决赛入围名单。

2019首届深度软件开发大赛是由深度科技发起的,在中国电子工业标准化技术协会信息技术应用创新工作委员会、中国智能终端操作系统联盟的支持和指导,联合国内优秀的软硬件厂商龙芯中科、华为、

成都申威等共同举办的软件赛事,本次大赛面向全球开发者征集原创软件作品,进一步丰富自主软件生态。大赛启动以来,受到了开发者、开源社区以及产业链合作伙伴的广泛关注,期间,开发者通过多种方式报名参与了本次大赛。

3月-6月,大赛完成了面向社会的作品征集,经过三周严格细致的评审,最终遴选出13支优秀作

品入围决赛。在此，感谢所有的参赛者这一路对深度软件大赛的支持，后续深度科技也将陆续联系作者，进行大赛下半程的精彩内容。

本次入围决赛的团队 / 作品在后续的时间中，将基于龙芯平台的终端硬件开发环境以及基于华为鲲鹏处理器的 TaiShan 服务器硬件开发环境进行作品的迁移适配工作。最终作品将在第 9 届深度开发者与用户大会现场进行展示评比，并在大会现场进行颁奖。

### 入围作品简介

#### 谭乐《VNote》

VNote 是一个更懂程序员和 Markdown 的笔记软件，以 MIT 协议开源。VNote 的目标就是提供一个带有强大文件管理功能的 Markdown 编辑器或者提供舒适 Markdown 体验的笔记软件。通过 VNote，在编辑模式下，用户既能掌控所有文本，也能清晰地抓住文章的所有信息。同时，VNote 将代码编辑器上面的优秀体验带到笔记软件中，提供诸如 Vim 绑定、片段、模糊跳转等功能，达到高效的管理和编辑。

#### 闫霄龙《ImagePy》

ImagePy 是基于 Python 的弹性处理框架，可以打开多种格式的图像数据，进行像素调整，数学运算，也可以对图像进行各种滤波运算，进行目标分割，像素统计，区域测量。弹性设计思想，可以方便的利用 scipy.ndimage, scikit-image, opencv 等图像处理库进行功能扩展。

#### 李登淳《Covariant Script 编程语言》

Covariant Script 编程语言是由李登淳在 2017 年初设计的一门新型编程语言，其具有语法简单易学、



深度科技副总经理与原工信部重大专项评审组组长  
周宝信共同发布名单

标准库强大易用、与 C/C++ 交互简单等特点，现其开发与发展已得到四川大学计算机学院的支持。

#### 宋佩阳《Bamboo 多功能 PDF 阅读器》

Bamboo 多功能 PDF 阅读器用于 PDF 文档的打开和编辑，主要由文档收藏、页面交互、笔记系统三个部分组成。其中，文档收藏功能可搭配网盘工具（如坚果云）使用，实现与云端同步操作。笔记系统可兼容其它工具（如 Foxit、Xodo 等），由其它工具生成的笔记也可以使用本阅读器进行编辑。最后，概览功能用于在页面两侧显示注释文字，该功能非常适用于双栏文档（如论文）的阅读和回顾。

#### 王冰《biForm 软件开发平台 V3.1》

biForm 适合于开发需要图形化用户界面（GUI）的各类桌面应用，以“表单”为所有应用的基本构成单元。基于 biForm 的开发过程即以创建、调试、打包、发布各种“表单”为基础展开进行。这种“表单”在 biForm 中有特定的格式和功能。biForm 以 XML 文件的形式描述了表单上的图形化控件、元数据、Python 脚本等。



张磊

深度科技 技术总监



靳国杰

龙芯中科 技术总监



霍旭东

成都申威 副总经理



红薯

开源中国社区创始人，开源项目 J2Cache 作者



王耀华

深度科技 高级研发工程师



晏然

Arch linux 开发者

评委组

**邓志淮《向日葵 X》**

向日葵远程控制软件客户端支持 Windows、Linux 与 Mac 三大系统，安装于远程主机上，通过向日葵控制台 web 页面或使用向日葵控制端访问并控制安装了向日葵客户端程序的远程主机。向日葵 X——将向日葵主控端与被控端合二为一，更小更快。拥有快速访问功能，集主被控为一体，通过使用识别码即可一键直达远程电脑桌面，只需安装一个软件就能快速实现在线远程协助功能，彻底结束了两个程序的历程，让远程操作真正做到“轻装上阵”，方便用户实现快速的远程协助操作。

**李常刚《电力系统仿真工具包》**

STEPS 是项目组使用 C++ 和 Python 自主开发的一款面向电力系统仿真的开源工具包，其主要功能是电力系统的潮流计算和机电暂态仿真，可作为电力系统运行控制人员的辅助决策工具。作品

采用模块化设计，具有良好的扩展性；实现了常见电力系统仿真软件尚未支持的多频电力系统互联仿真。STEPS 支持三种并行仿真模式：核心级 openMP 并行、程序级 openMP 并行和脚本级并行，大大提升了计算效率。STEPS 提供了一系列可供外部程序调用的 API 接口，并使用 Python 封装成 pysteps 模块，用户既可以直接利用现有模块功能进行研究，也可以利用 API 接口进一步扩展高级应用。

**黄安俊《sysbro》**

sysbro 是一款专为 deepin 操作系统打造的一款系统助手，并且支持在 Ubuntu18.10 进行使用，其他发行版暂未测试和适配，开发语言使用 C++，基于 Qt 图形库之上进行开发，使用 libdtkwidjet 保持 DDE 应用软件 UI/UX 风格统一，软件界面简洁，首页资源监控，系统清理等功能，内置几款实用



的工具，目标是做实用的系统优化工具。

### 贾庆彤《虚拟触控板》

虚拟触控板将各种带触摸屏的设备变成台式机的触摸板（屏）。目前支持 1-5 点触控，并且支持（模拟）压力感应，可采用 USB 或 WIFI 进行连接。用户可设置两点触摸时为触摸屏模式，便于浏览器中的滚动与缩放。同时可在电脑端对触控板的相关参数进行调整。

### 黄颖《海天鹰画图》

海天鹰画图软件力争超越微软画图，追赶色彩风暴。

### 张浩斌《Hustoj 开源版》

HUSTOJ 是由华中科技大学毕业生王良晶等同学开发并在 2008 年底以 GPL 协议进行开源的一个在线评测系统，HUSTOJ 采用了 LAMP 平台进行开发，其早期版本在华中科技大学的服务器上稳定运行至今。早期的 HUSTOJ 在开源方面与 ZOJ、BOJ 一样，缺少应有的文档和技术支持，甚至没有提供数据库结构，使得在开源后长达一年的时间中没有任何其他的学校和组织进行部署。作者从 2009 年 9 月开始加入开源在线评测系统 HUSTOJ 的开发，并在创始人的支持下成为 HUSTOJ 的项目管理员。

### Samson Wen《harbian-audit-deepin》

harbian-audit 是 GPL 3.0 协议的 free/libre software，是基于 Debian GNU/Linux 服务器的安全审计 / 修复的工具，是由两个安全部署合规参考实现 STIG 和 CIS 的合集，目前版本 (V0.3.0) 不仅实现了安全审计功能，且可自动修复 85% 的安全部署项。

基于此项目制作了满足 harbian-audit 安全部署的 AMI 云主机镜像及 QEMU 镜像，目前 AWS 的三个区 (EU(Frankfurt), Asia Pacific(Tokyo), US

East(Ohio)) 的用户都可以免费使用，对于那些不愿意使用 "Someone else's computer" 的私有云用户我们也提供了 QEMU 镜像，至于那些无法信任 HardenedLinux 社区制作镜像的用户也没关系，HardenedLinux 社区已经公开了如何制作 AWS 和 QEMU 镜像以及如何把 harbian-audit 应用到云主机镜像的文档。

### 马艺诚《deepin-file-control》

本作品是基于 Linux 系统开发，主要用于监控和记录指定目录下文件的相关操作（打开和关闭）

1. 使用了 hook->client->server
2. Client 端使用 unix 域和 hook 进行通信
3. Server 参考 muduo 网络库

通过此次大赛，我们看到了很多优秀的作品，更值得惊喜的是有很多优秀的开发者和团队关注并且参与到国产 CPU 软件生态的建设中。我们也希望没有进入到下半程大赛的参赛者继续关注大赛，并准备参与到今后深度举办的活动中。d





## 单一 deepin os 用了 5 个月，很满意

phoenifox / 2019-5-4 22:50

虽然 N 年前我买第一台电脑时就尝试过各种装机折腾，最早接触的 Linux 就是中科红旗 red-flag linux，当时还一并尝试过 redhat linux。但是那时驱动比现在还少，QQ 在 Linux 上都用不了，如何接到互联网也是搞不明白，周边一个用 Linux 的都没有。所以那时仅止于下载安装来体验下，那时的 redflag linux 界面与 windows XP 差不多，总体来说其实不错了。后来又用过湖南 bolod X OS，对 Linux 的易用性有了更进一步了解。

·第一次安装 deepin: 2014 年 X 月

第一台机器: 本机 (hp 1702tu 升级, CPU T7200, 内存 2G)

安装 deepin, 限于自己对 Linux 的知识, 那时机器 (其实还是正在用的这台 hp 旧机器) 用起来很卡, 不久我就放弃了, 等待新的 deepin 版本出来, 以及等待我的另外一台机从主要机器上退役。

·第二次安装 deepin: 2018 年 12 月

第二次安装的机器: 还是本机 hp1702tu

但去年再次安装 deepin 15.8 后, 同样一台机器使用的感觉与 deepin 2014 已经有巨大的不同了, 明显流畅了许多, 于是, 我将硬盘全盘格式化, 安装单一 deepin 系统。实际上以我这台旧机器来说, 今天以前, 速度也并不快, 只是能用, 不卡, 稍慢。体验了一段时间后, 在我的另一台机器上也安装 deepin 单一系统了。

第二次安装的第二台机器: alienware(CPU i7 ? 内存 8G)

·这台机器也用了有几年了, 不是新机, 当初买来其实并没有用于打游戏, 但是 windows 更新后磁盘分区占满了, 所以很烦, 当初我是单独购买的正版 win8 和 ms office 2013 价格不便宜。安装 deepin 后清静了。

·这次要说的是 hp 1702tu 这台机。这台机器最旧, 所以变成试验机, 每次尝试新玩法都是先在这台机上捣鼓。但是由于它其实用起来比较慢, 基本上大多数时候我都让它躺着。最近就安装 deepin 15.10 stable, 新学了一招, 分 swap 区。所以我昨天给硬盘分了几个区, 一个 SWAP 区 4G, 一个安装系统 50G, 另外的装其他文件。这样分区后, 安装 deepin 15.10 stable, 系统速度比之前提高了一倍, 打开网页的速度是 2 秒, 打开启动器也是 2 秒, 打开文件管理器 2.5 秒, 打开软件商店完全展开 8-16 秒, 打开控制中心 1 秒, 打开 wps 开始可以编辑 6-13 秒, 打开计算器 5 秒, 打开编辑器 2 秒, 打开 VLC 3 秒。我觉得这么老的机器也能这样运行, 比较满意了。

现在 alienware 那台机器, 平时娱乐, 某些文件处理, 会议用, 我喜欢里面的录屏功能, 自带软件能记录下来是很棒的。

这台 hp 机, 现在速度快起来了, 可能我也会用多起来了。

总体感觉, 连续使用这 5 个月, 机器用多了, 就会对这个系统越来越熟悉, 即使不去编程不用命令行, 一样能用好 deepin, 熟悉后就能找到各种能够用的应用软件, 慢慢也就无师自通一些技能, 根据自己的需求, 即使软件商店里面没有的软件, 也能自己搜到, 自然就找到和建立了适合自己的软件生态。就我个人而言, 如果不考虑软件使用习惯和文件管理方便, deepin 和现有的其他应用软件能完全替代 win 系统下的一切了。mac 我有一台, 用的很少, 使用习惯与 win 差别太大, 在与 win 之间切换太累, 还是 deepin 与 win 之间切换感觉不出来。

---

deepin\_zxr 发表于 2019-5-5 08:51:17

我用了 5 年, 确实不错, 越来越好了。

---

jianguo922 发表于 2019-5-5 12:26:30

中兴事件后一直深度 deepin 系统

---



## 貌似老外发的关于 deepin 的视频介绍和评价比国内的多

phoenifox / 2019-5-4 00:44

看到 youtube 上关于 deepin 的介绍和评论的视频比国内的多得多，评论也热烈。国内的视频网站发视频介绍的还是偏少啊。在 youtube 上看到了 deepin 在触摸屏上的操作体验。



Who Does the Deepin Desktop Best?

14,825 views

322

14

SHARE

SAVE

...



Deepin OS 15.9 new touchscreen features

5,223 views

100

7

SHARE

SAVE

...



Is Deepin Spyware?

8,160 views

247

15

SHARE

SAVE

...



DeepinOS 15.8: Burn it with Fire!

5,710 views

152

32

SHARE

SAVE

...

antizo 发表于 2019-5-4 20:00:27

开箱即用这一块，deepin 算是做的不错的了，毕竟桌面环境的日常使用稳定最重要的，只是国内开源软件环境生态很差，原生软件太少了，体验一般，只能靠 wine 去驱动了，这也是 deepin 的无奈。

Guumi 发表于 2019-5-4 09:57:11

这个还是使用的人群少的问题，基数少所以发视频或文章的人也少。。。

cy844757727 发表于 2019-5-4 11:11:09

国外 linux 市场占有率比国内的高啊

yun363901949 发表于 2019-5-4 22:19:33

年轻人接受新事物不难，关键 deepin 做的是操作系统，还是新生的，在不改变用户使用习惯，符合用户使用要求下，年轻人才会去接受他，使用它，deepin 也需要主动去改变，去适应!!!



## 统计用 deepin 的人都是做什么工作的

c798654350 / 2018-11-14 16:52

看看用 deepin 的人都是做什么的

单选投票, 共有 1424 人参与投票

1. java 开发工程师
2. c/c++ 开发工程师
3. python 开发工程师
4. php 开发工程师
5. go 开发工程师
6. web 前端工程师
7. 安卓 /ios 开发工程师
8. AI/ 大数据分析 / 挖掘 / 工程师
9. 运维 /linux 系统工程师
10. 软件测试工程师
11. dba/ 数据库开发工程师
12. 其它 IT 类工作
13. 非 IT 行业工作
14. 学生

sored520 发表于 2018-11-14 17:52:34

产品狗

w1988 发表于 2018-11-14 17:58:08

做图, 剪视频, 偶尔帮朋友鼓捣电脑, 主业 养车人。

stain 发表于 2018-11-14 18:16:40

Deepin 适合绝大多数人使用, GNU/Linux 不是程序员的玩具。

yorun 发表于 2018-11-14 18:42:42

全栈开发

sinan 发表于 2018-11-14 18:46:43

某国企小卒子, 从大学时期用 deepin, 已经习惯了, 所以现在工作生活也在使用 deepin。现在用别人的 win10 电脑都用不顺手了。

a1551912292 发表于 2018-11-14 18:57:56

本人是一个普通的煤矿工人, 然而非常喜欢折腾电脑, 各位工程师们见笑了

---

lxj929 发表于 2018-11-14 19:29:15

电工路过

---

mianyz 发表于 2018-11-14 19:43:48

业余爱好者，完全没基础

---

ailing 发表于 2018-11-14 19:45:40

科研狗，需要大规模科学计算。

---

txdtj 发表于 2018-11-14 21:26:54 来自移动端

小规模纳税人，公司技术员一名，老本行是爬电线杆的。

---

crqscn 志愿者 发表于 2018-11-14 22:38:35

我是个苦逼的个体户

---

xy\_god 志愿者 发表于 2018-11-14 23:22:47

非 it 行业的占比这么高，还是出乎意料的

---

suzuki88 发表于 2018-11-15 09:15:46

教育工作。纯粹爱折腾。

---

wzh 发表于 2018-11-15 20:19:10

物业公司办公管理人员

---

gentoo 发表于 2018-11-16 00:25:22

血汗工厂农民工

---

oliveryty 发表于 2018-12-13 22:18:37

文科生，学英语，干银行的

---

wayima 发表于 2018-12-14 12:33:55

记者啊 有新鲜感

---

xiaolin1995 发表于 2019-1-6 22:57:34

做生信，需要 Linux 系统和 python 所以就用 deepin 了 驱动各方面都支持的不错。

---

owen19 发表于 2019-5-15 05:26:07

本人是通讯 CT 行业

---



# 申威容器云平台方案

申威容器云平台方案采用 docker，其中 docker daemon 以服务的形式存在，负责管理包括容器镜像、容器、网络以及数据卷在内的所有 docker 对象，并提供 REST API 接口供 docker client 或者其他软件调用。docker client 是用户和 docker daemon 交互的主要方式，当通过 docker client 发送命令后，会转换为相应的 REST API 请求发给指定的 daemon。

镜像仓库用于存储和分发各种申威架构下的基础镜像和应用镜像，深度科技建议使用 Harbor。Harbor 提供管理 UI，支持基于角色的访问控制，提供 AD/LDAP 集成、以及审计日志 (Audit logging) 等企业用户需要的功能。

容器编排工具则提供容器的集群化管理。目前在深度操作系统申威服务器版可使用的容器编排工具包括 Docker Swarm 和 Kubernetes。

深度科技不仅使用 Go 语言，而且也参与了 Golang 生态的建设，向 Golang 编译器项目提交了多个被上游合并的补丁。容器技术栈大量使用 Go 语言开发，深度科技作为申威平台 Golang 编译器适配和优化项目的参与单位，在容器技术支撑方面具有先天的技术优势，主要表现在如下方面：

- 深度科技移植了大量容器支撑应用，在申威平台已基本完成容器云生态的建设；
- 容器的运行时使用 docker，docker 是目前业界使用最广泛的容器引擎；
- 编排方案使用 Kubernetes，支持集群负载均衡和自动缩放；
- 部署方案使用 rancher，支持多个不同的

Kubernetes 版本部署；

- 镜像仓库的管理使用 Harbor，可支持多用户账号的授权管理以及日志审计等功能；
- 报警使用 alertmanager，可通过 webhook 或者 wechat 等方式向管理员发送报警；
- 仪表盘使用 Grafana，具有丰富的图表自定义功能。

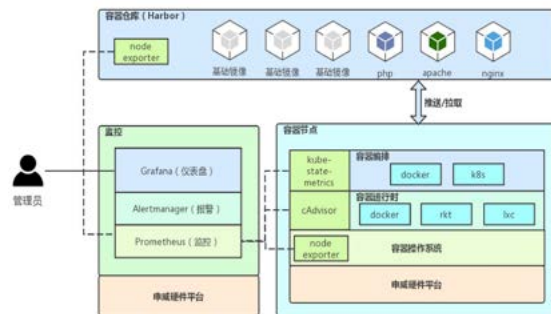


图 1 容器云方案

## 容器支持

容器是一种轻量虚拟化技术，使用容器运行时取代虚拟机中 hypervisor 的地位，负责容器的执行以及容器镜像的管理。容器和主机系统共享硬件资源，共用内核。容器运行时利用 Linux 内核提供的 namespace 隔离技术实现进程运行级别隔离，利用 cgroup 对容器资源进行限定，利用 seccomp 以及 apparmor 对容器可使用的系统调用进行限定，从而提供不同纬度的安全特性。容器技术优势体现在如下几点：

- 轻量，资源损耗少，性能上接近 Host OS

的运行效率；

- 可保证开发、测试、部署环境的一致性；
- 不依赖硬件的特性，对内核特性要求较少，容易移植到国产平台；

容器是混合架构重要组成部分之一。使用容器可以在消耗少量资源前提下，为应用提供较为理想的沙盒运行环境，是一种兼顾安全和效率的方案。而与容器相关的技术大多采用 Go 语言编写，因此是否了解申威架构、熟悉申威指令，可否提供 Go 语言库包申威平台的移植支持，是考量一个操作系统厂商能否对容器平台进行支撑的必要因素。

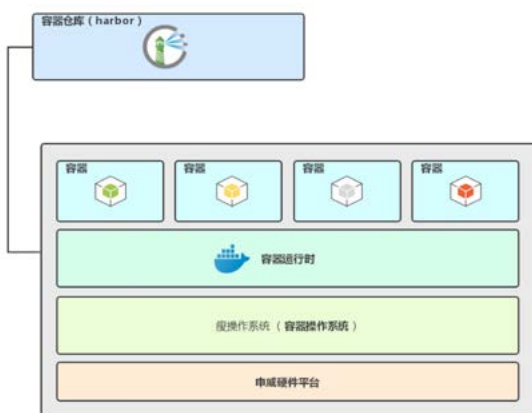


图 2 容器技术栈

深度科技在申威平台已独立完成了多个 docker 版本的移植工作。包括 docker 17.06, docker 18.06 和 docker 18.09 版本。

docker 是容器运行时，其上需支撑各种应用容器的运行。为便于用户自定义容器镜像，深度科技提供基础镜像制作工具和公开的互联网软件仓库。考虑到部分静态编译的应用无需复杂的操作系统底层库支持，深度还构建了 busybox 镜像。4M 尺寸的镜像体积可满足无复杂依赖应用的容器镜像支撑。

## 容器编排方案

容器编排是对容器进行一种有效管理与调度。可将成多个容器有序，高效地组合在一起，持续稳定地提供服务。目前深度操作系统申威版支持 Docker Compose 和 Docker Swarm。两者都是 Docker 官方推荐的编排工具。其中 Docker Compose 是在单个服务器上创建多个容器的编排工具，而 Docker Swarm 则可以在多个服务器上创建容器集群服务，对于微服务的部署，显然 Docker Swarm 会更加适合。

针对大规模容器集群，深度科技则移植了目前业界最主流的 Kubernetes。Kubernetes 基于容器技术，构建可自动包装，横向缩放，自我修复，自动部署和回滚，服务发现和负载均衡的容器云平台。

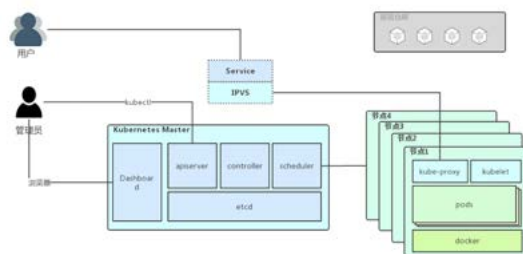


图 3 Kubernetes 架构图

Kubernetes 包含众多组件，核心组件及其功能说明如下：

组件	功能说明
etcd	分布式存储数据库，保存集群配置与状态
apiserver	提供了资源操作的唯一入口，并提供认证、授权、访问控制、API 注册和发现等机制
controller manager	负责维护集群的状态，比如故障检测、自动扩展、滚动更新等
scheduler	负责资源的调度，按照预定的调度策略对 Pod 进行调度
kubelet	负责维护容器的生命周期，同时也负责 Volume (CSI) 和网络 (CNI) 的管理
kube-proxy	提供节点内部的服务发现和负载均衡



CoreDNS	为整个集群提供 DNS 服务
Flannel	CNI 网络插件，为集群提供网络服务，动态构建集群网络
cAdvisor	资源的实时监控和性能数据采集

Kubernetes 集群包含两类不同的节点，其中 Master 节点是集群的控制节点，负责整个集群的管理和控制。针对集群执行的控制命令都是发送给 Master 节点。Master 节点上运行的主要组件如下：

- kube-apiserver: kube-apiserver 提供 REST 接口服务，是集群控制的入口
- kube-controller-manager: 是 Kubernetes 集群中所有资源对象的自动化控制中心
- kube-scheduler: 负责 Pod 的调度

Node 节点是 Kubernetes 集群中的工作负载节点，Node 上的工作负载由 Master 分配，当某个 Node 宕机时，Master 会将上面的工作负载转移到其他节点上去。Node 节点上运行的主要组件如下：

- kubelet: 负责 Pod 的创建、启动、监控、重启、销毁等工作，同时与 Master 节点协作，实现集群管理的基本功能。kubelet 把自己注册到 Master 节点，当 Node 节点纳入集群管理的范围后，kubelet 进程就定时向 Master 节点上报自身的资源情况。Master 节点掌握着每个 Node 节点的资源情况，就可以实施高效的资源调度策略。当某个 Node 超过指定时间未上报信息，Master 会将该 Node 标记为 Not Ready，同时会进行工作负载转移。

- kube-proxy: 实现 Kubernetes Service 的通信以及负载均衡。

除此之外，kube-dns 或 CoreDNS 作为集群的必备扩展来提供命名服务，通过 DNS 扩展，每一个服务都会产生一个独一无二的 FQDN (Fully Qualified Domain Name) 名称。各类 Flannel 以及

CNI 网络插件提供集群内容器之间、容器和集群中所有的节点的通信和网络互联。

## 容器云支撑情况

仓库、镜像和容器是 Docker 使用的三要素。其中镜像仓库相当于传统操作系统的软件仓库，用来存储和分发 Docker 镜像。Docker 镜像则相当于传统操作系统中的应用软件。容器是 Docker 镜像的运行实例。因此，只提供 Docker 引擎本身还无法有效支撑容器生态。只有构建一个汇聚林林总总、各式各样的容器镜像的开放容器仓库才能打造出一个丰富的容器生态。

容器编排工具则将容器的生命周期管理从部署单一节点扩展到物理集群上。一般用于部署复杂的、多容器工作负载。

深度科技在申威平台已完成 Kubernetes 的移植，此外还将通过移植 Dashboard 提供可视化 Kubernetes 管理，通过移植 Prometheus 提供对容器进行监控报警的能力。以提供一个较为完整的容器云技术栈支撑。

## Harbor

深度科技移植了 Docker Registry 组件。但 Registry 仅支持基本的镜像存储和分发，缺少诸如用户管理、授权控制、项目分类以及安全审计等功能。

相比 Docker Registry，Harbor 则提供了更好的性能和安全性。其支持在多个 Registry 节点间复制镜像资源，也提供高级的安全特性（诸如用户管理，访问控制和活动审计等）。特别适合用于搭建企业内部的私有 Registry，可确保数据和知识产权在公司内部网络中管控。Harbor 的主要特点如下：

- 基于角色的访问控制；
- 镜像可以在多个 Registry 实例中复制或同步，



适合于负载均衡，高可用，混合云和多云的场景；

- 用户可以通过浏览器来浏览，检索当前镜像仓库，管理项目和命名空间；
- Harbor 可集成企业内部已有的 AD/LDAP，用于权限认证管理；
- 支持审计管理，所有针对镜像仓库的操作都可以被记录追溯；
- 支持 RESTful API 提供给管理员对于 Harbor 更多的操控，使得与其它管理软件集成变得更容易；
- 部署简单，提供在线和离线两种安装工具。

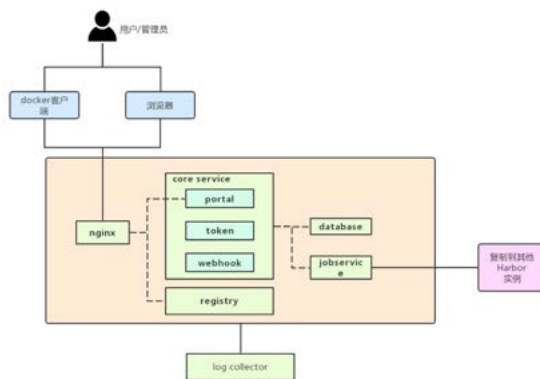


图 4 Harbor 模块结构图

Harbor 在架构上主要由五个组件构成：

- Nginx: Harbor 服务通过 Nginx 反向代理统一接收浏览器、Docker 客户端的请求，并将请求转发给后端不同的服务；
- Registry: 负责储存 Docker 镜像，并处理 Docker push/pull 命令；
- Core services: 这是 Harbor 的核心模块。Portal 提供 Harbor 的 UI 管理界面，为及时获取 Registry 上镜像状态变化的情况，在 Registry 上配置 webhook，把状态变化传递给 portal 模块。

token 服务则负责根据用户权限给每个 push/pull 命令签发 token。

- Database: 为 core services 提供数据库服务，负责储存用户权限、审计日志、Docker image 分组信息等数据。
- Log collector: 为了帮助监控 Harbor 运行，负责收集其他组件的 log，供日后进行分析。

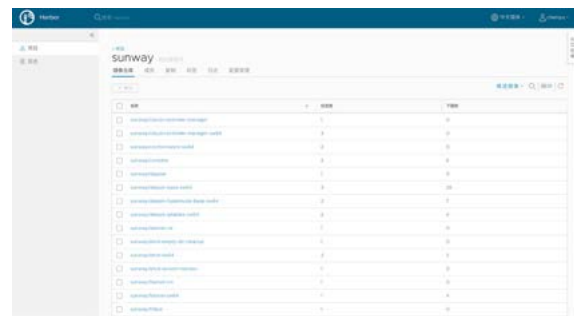


图 5 申威平台 Harbor 运行实例

有了容器仓库的支持，深度科技将持续构建更多的容器镜像，补充诸如 Busybox, MariaDB, Apache, PHP 等各类基础和应用程序镜像。同时为帮助更多的申威合作伙伴参与建设申威容器生态，深度科技将会向项目参与单位开放 Registry 以及 Harbor 的安装包和部署文档。并根据项目用户的需求，提供诸如容器镜像定制、内网容器仓库搭建等服务，协助其构建各类可持续交付的微服务。d



# 使用 Vuls 对 linux 系统进行漏洞扫描

● 深度科技 武汉公司 研发中心 / 文

## 概述

Vuls 是针对 Linux 系统的安全扫描工具，可以扫描各种系统安全漏洞并给出修订建议。

\* 漏洞 (vulnerability):

\* CVE: CVE 是一个字典，为公开披露的网络安全漏洞和风险提供定义。CVE 的目标是为漏洞管理工具更方面的使用这些共享数据。CVE 条目由标识号，描述和至少一个公共引用组成。

\* CVE list: 创建 CVE 条目的过程始于发现潜在的安全漏洞或暴露。然后由 CVE 编号机构 (CNA) 为该信息分配 CVE ID，由 CNA 添加描述和参考，然后由 CNA 在 CVE 网站上发布 CVE 条目。

\* NVD: 国际漏洞数据库，由美国国家标准与技术研究院 (NIST) 于 2005 年推出，它提供了一个增强型 CVE 内容的漏洞数据库，该数据库与 CVE 列表完全同步，因此对 CVE 列表的任何更新都会立即显示在 NVD 中。

\* Exploits 漏洞利用工具。

\* CVSS: CVSS 是一个行业的标准，主要目的是评估安全漏洞的严重性。CVSS 3.0 版提供了“捕获漏洞主要特征的方法，并生成反映其严重性的数字分数，以及该分数的文本表示。然后可以将数字分数转换为定性表示（例如低、中、高和关键），以帮助组织正确评估其漏洞管理流程并确定其优先级。”

## 扫描环境部署

依赖:

```
· go1
· automake
· gccgo-go
· go-dep
```

```
$ wget https://dl.google.com/go/go1.10.1.linux-amd64.tar.gz
$ sudo tar -C /usr/local -xzf go1.10.1.linux-amd64.tar.gz
mkdir $HOME/go
sudo apt-get install automake go-dep gccgo-go
```

在文件 /etc/profile.d/goenv.sh 添加下面的几行，把 go 的环境变量加入到系统配置中。

```
export GOROOT=/usr/local/go
export GOPATH=$HOME/go
export PATH=$PATH:$GOROOT/bin:$GOPATH/bin
```

立马应用环境设置:

```
source /etc/profile.d/goenv.sh
```

## 部署 go-cve-dictionary

```
$ sudo mkdir /var/log/vuls
$ sudo chown deepin /var/log/vuls
$ sudo chmod 700 /var/log/vuls
$ mkdir -p $GOPATH/src/github.com/kotakanbe
$ cd $GOPATH/src/github.com/kotakanbe
$ git clone https://github.com/kotakanbe/go-cve-dictionary.git
$ cd go-cve-dictionary
$ make install
```

go-cve-dictionary 会从 NVD 上 fetch 漏洞信息，然后插入 SQLite 数据库中。在 shell 中运行下面的语句，会下载 2002 年到今年的所有年份的漏洞数据。

```
for i in `seq 2002 $(date +"%Y")`; do go-cve-dictionary
fetchnvd -years $i; done
```

### 启动 CVE dictionary http 服务

```
go-cve-dictionary server
```

可以使用命令 ‘go-cve-dictionary list’ 查看最新漏洞数据的更新日期以及本地 nvd 和 jvn 漏洞信息是否需要更新。

```
deepin@deepin-PC:~$ go-cve-dictionary list
INFO[03-13]09:17:11] Fetching... https://nvd.nist.gov/feeds/
json/cve/1.0/nvdcve-1.0-modified.meta
INFO[03-13]09:17:11] Fetching... https://nvd.nist.gov/feeds/
json/cve/1.0/nvdcve-1.0-recent.meta
INFO[03-13]09:17:13] Fetched... https://nvd.nist.gov/feeds/
json/cve/1.0/nvdcve-1.0-modified.meta
INFO[03-13]09:17:13] Fetched... https://nvd.nist.gov/feeds/
json/cve/1.0/nvdcve-1.0-recent.meta
INFO[03-13]09:17:13] Fetching... https://jvndb.jvn.jp/ja/feed/
checksum.txt
INFO[03-13]09:17:15] Fetched... https://jvndb.jvn.jp/ja/feed/
checksum.txt
+-----+-----+-----+-----+-----+
| SOURCE | YEAR | STATUS | FETCHED | LATEST |
+-----+-----+-----+-----+
| NVD(JSON) | modified | Out-Dated | 2019/3/12-01:03 | 2019/3/12-20:02 |
| NVD(JSON) | recent | Out-Dated | 2019/3/12-01:00 | 2019/3/12-20:00 |
| JVN | modified | Up-to-Date | 2019/3/12-17:00 | 2019/3/12-17:00 |
| JVN | recent | Up-to-Date | 2019/3/12-17:00 | 2019/3/12-17:00 |
+-----+-----+-----+-----+-----+

```

To update feeds, execute the following commands.  
\$ go-cve-dictionary fetchnvd -latest # 更新本地 nvd 数据库信息到最新

### 部署 goval-dictionary

go-cve-dictionary 是构建 OVAL 的本地副本的工具，本地副本以 sqlite 格式存储。

部署 goval-dictionary 需要使用代理，否则会失败。

```
$ mkdir -p $GOPATH/src/github.com/kotakanbe
$ cd $GOPATH/src/github.com/kotakanbe
$ git clone https://github.com/kotakanbe/goval-dictionary.git
$ cd goval-dictionary
```

```
$ make install
```

### 获取 debian 的 OVAL 数据（漏洞字典）

```
goval-dictionary fetch-debian 7 8 9 10 # 获取 debian 的 7,8,9,10 版本的数据
```

### 部署 gost (可选)

构建安全跟踪器 (Redhat 或 Debian 或 Microsoft) 的本地副本，部署 gost 需要使用代理，否则会失败。

```
$ sudo mkdir /var/log/gost
$ sudo chown deepin /var/log/gost
$ sudo chmod 700 /var/log/gost
$ mkdir -p $GOPATH/src/github.com/knqyf263
$ cd $GOPATH/src/github.com/knqyf263
$ git clone https://github.com/knqyf263/gost.git
$ cd gost
$ make install
$ gost fetch debian 获取 debian 的漏洞信息
```

### 部署 go-exploitdb

利用 cve ID 或者漏洞利用数据库 ID 从漏洞利用数据库中获取相关 Exploits 的工具。https://www.exploit-db.com 上会有最新的漏洞信息。

```
$ sudo mkdir /var/log/go-exploitdb
$ sudo chown deepin /var/log/go-exploitdb
$ sudo chmod 700 /var/log/go-exploitdb
$
$ mkdir -p $GOPATH/src/github.com/mozqnet
$ cd $GOPATH/src/github.com/mozqnet
$ git clone https://github.com/mozqnet/go-exploitdb.git
$ cd go-exploitdb
$ make install
```

### 获取 exploit-db 信息。

```
go-exploitdb fetch
```

### 部署 Vuls

```
$ mkdir -p $GOPATH/src/github.com/future-architect
$ cd $GOPATH/src/github.com/future-architect
```



```
$ git clone https://github.com/future-architect/vuls.git
$ cd vuls
$ make install
```

### 本地扫描模式

在家目录创建 vuls, 进入该目录创建配置文件: config.toml。内容如下:

```
[server]

[servers.localhost]
host = "localhost"
port = "local"
```

检查服务器上的 config.toml 和设置

```
vuls configtest
```

修改本机的 /etc/lsb-release 文件

```
DISTRIB_ID=debian
DISTRIB_RELEASE="10"
```

下载 oval.sqlite3 和 cve.sqlite3 数据库文件到 \$HOME/vuls 目录:

- oval.sqlite3
- cve.sqlite3

### 开始扫描

开始扫描

```
vuls scan
```

### 查看扫描结果

命令行查看完整扫描结果:

```
vuls report -format-full-text |less
```

### 使用 Vuls 自带的基于终端的用户界面以显示扫描结果

```
vuls tui
```

### 使用 web UI 的方式查看结果

VulsRepo 是 Vuls 的 OSS Web UI, 通过该界面可以像 Excel 的数据透视表一样分析扫描结果。

\* 创建用户 vuls-user, 登录用户 vuls-user;

下载 vulsrepo 源码

```
$ cd $HOME
$ git clone https://github.com/usiusi360/vulsrepo.git
```

更改 vulsrepo-server 的设置。

```
$ cd $HOME/vulsrepo/server
$ cp vulsrepo-config.toml.sample vulsrepo-config.toml
$ vi vulsrepo-config.toml
[Server]
rootPath = "/home/vuls-user/vulsrepo"
resultsPath = "/home/deepin/vuls/results"
serverPort = "5111"
```

创建一个 vuls 的 json 报告

切换到 deepin 账户, 执行 vuls 扫描, 并生成 json 结果。

```
$ cd /home/deepin/vuls/
$ vuls scan
$ vuls report -format-json
```

```
# 启动 vulsrepo-server
```

```
$ cd /home/vuls-user/vulsrepo/server
$ sudo ./vulsrepo-server
```

浏览器访问 web ui:

```
http://localhost:5111
```

### 远程扫描模式

远程扫描模式参考: 远程扫描模式即可, 按照步骤来就行了。

### deepin15.9.2 扫描结果

我在 deepin15.9.2 上执行了一次扫描, 扫描结果可以通过浏览器访问: Deepin15.9.2 vuls 扫描结果。 [d](#)

# 得到系统调用的函数签名

● 深度科技 武汉公司 研发中心 / 文

由于有时候需要获取一大堆系统调用的函数签名 (glibc 的)，一个个查实在很麻烦，因此需要写一个程序来做，思路还是蛮简单的，就是从 0 开始循环，使用 `ausyscall` 获取每个系统调用的名称，然后再使用 `man 2` 来获取每个系统调用的函数签名，在这里最麻烦的就是要分析 `man 2` 的输出结果，因为一个函数可能是这样的 (函数包含多行)：

```
SYNOPSIS
#include <sys/mman.h>

void *mmap(void *addr, size_t length, int prot, int flags,
           int fd, off_t offset);
int munmap(void *addr, size_t length);

See NOTES for information on feature test macro
requirements.
```

也可能是这样的 (函数所在行包含注释)：

```
SYNOPSIS
#include <unistd.h>

int setpgid(pid_t pid, pid_t pgid);
pid_t getpgid(pid_t pid);

pid_t getpgrp(void);      /* POSIX.1 version */
pid_t getpgrp(pid_t pid); /* BSD version */

int setpgrp(void);       /* System V version */
int setpgrp(pid_t pid, pid_t pgid); /* BSD version */

Feature Test Macro Requirements for glibc (see feature_test_
macros(7)):

getpgid():
_XOPEN_SOURCE >= 500
|| /* Since glibc 2.12: */ _POSIX_C_SOURCE >= 200809L
```

还有可能是这样的：

```
SYNOPSIS
#include <signal.h>

int sigaction(int signum, const struct sigaction *act,
              struct sigaction *oldact);

Feature Test Macro Requirements for glibc (see feature_test_
macros(7)):

sigaction(): _POSIX_C_SOURCE

siginfo_t: _POSIX_C_SOURCE >= 199309L
```

甚至还有名称不一致的，比如 `man 2 exit` 的结果里并没有 `exit`，而是 `_exit`：

```
SYNOPSIS
#include <unistd.h>

void _exit(int status);

#include <stdlib.h>

void _Exit(int status);
```

所以虽然开始写了一个 shell 脚本来做这种事，`for ((i=0; i<333; i++)) do syscname=ausyscall ${i}; syscsig=man 2 ${syscname}|grep -m 1 "${syscname}(" ; echo &quot;${i} ${syscname} ${syscsig}&quot;; done`，但后来也只能用 python 程序了，如下：

```
#!/usr/bin/env python
```



```

import subprocess
import re

def strip_comment(line):
    line = line.strip()
    if line.endswith('/') and line.startswith('/'):
        return ""

    if line.endswith('/') and line.find('/') > 0:
        return line[line.find('/')].strip()

    if line.startswith('/') and line.find('/') > 0:
        return line[line.find('/')+2:].strip()

    return line

def get_synopsis(manpage):
    lines = [_strip() for _ in manpage.split('\n')]
    in_synopsis = False
    synopsis = []
    for line in lines:
        if in_synopsis:
            if line == 'DESCRIPTION':
                break
            else:
                synopsis.append(strip_comment(line))
        else:
            if line == 'SYNOPSIS':
                in_synopsis = True

    return synopsis

def get_functions(synopsis):
    in_function = False
    result = []
    for l in synopsis:
        m = re.match(r'^[a-z][a-z_0-9]+ [\*]?([a-z][a-z_0-9]+[ ])?\*(.*)', l)

        if m is None:
            m = re.match(r'^unsigned int (alarm)\*(.*)', l)

        # one line
        if not in_function and m is not None and (l.count('(') == l.count(')') and l.endswith(';')):
            result.append(m.group(1) + " " + l)
            continue

        # first line
        if not in_function and m is not None and (l.count('(') == l.count('')+1 and (l.endswith(',') or l.endswith('...'))):
            result.append(m.group(1) + " " + l)
            in_function = True
            continue

        # middle line
        if in_function:
            result[-1] = result[-1] + l

```

```

        # last line
        if in_function and (l.count('(')+1 == l.count(')') and l.endswith(';')):
            in_function = False
            continue

        return result

def get_apropos(name, functions):
    result = []
    for l in functions:
        if l.startswith(name + ' '):
            result.append([len(name)+1:])

    if name.startswith('rt_') and len(result) == 0:
        return get_apropos(name[3:], functions)

    return result

total = 0
empties = []
for i in range(350):
    try:
        syscname = subprocess.check_output(['ausyscall', str(i)],
stderr=subprocess.STDOUT).strip()
    except subprocess.CalledProcessError, err:
        break

    total += 1
    try:
        manpage = subprocess.check_output(['man', '3' if 60 == i
else '2', syscname], stderr=subprocess.STDOUT)
    except subprocess.CalledProcessError, err:
        manpage = ""

    functions = get_apropos(syscname, get_functions(get_synopsis(manpage)))
    if len(functions) == 0:
        empties.append((i, syscname))
        continue
    print i, syscname
    for l in functions:
        print '\t', l

print '='*10, 'EMPTY SYSCALLS {0}/{1}'.format(len(empties), total), '='*10
for emp in empties:
    print emp[0], emp[1]

```

下面是输出:

```

0 read
ssize_t read(int fd, void *buf, size_t count);
1 write
ssize_t write(int fd, const void *buf, size_t count);

```

```

2 open
  int open(const char *pathname, int flags);
  int open(const char *pathname, int flags, mode_t mode);
3 close
  int close(int fd);
4 stat
  int stat(const char *pathname, struct stat *buf);
5 fstat
  int fstat(int fd, struct stat *buf);
6 lstat
  int lstat(const char *pathname, struct stat *buf);
7 poll
  int poll(struct pollfd *fds, nfds_t nfd, int timeout);
8 lseek
  off_t lseek(int fd, off_t offset, int whence);
9 mmap
  void *mmap(void *addr, size_t length, int prot, int flags, int
fd, off_t offset);
10 mprotect
  int mprotect(void *addr, size_t len, int prot);
11 munmap
  int munmap(void *addr, size_t length);
12 brk
  int brk(void *addr);
... .. ( 此处省略几百行 )
===== EMPTY SYSCALLS 22/333 =====
156 _sysctl
181 getpmsg
182 putpmsg
183 afs_syscall
184 tuxcall
185 security
214 epoll_ctl_old
215 epoll_wait_old
221 fadvise64
236 vserver
262 newfstatat
270 pselect6
283 timerfd
289 signalfd4
290 eventfd2
302 prlimit64
305 clock_adjtime
323 userfaultfd
329 pkey_mprotect
330 pkey_alloc
331 pkey_free
332 statx

```



## 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”





# 质量管理体系

● 深度科技 武汉公司 研发中心 / 文

## 质量管理体系的定义

任何组织都需要管理。当管理与质量有关时，则为质量管理。质量管理是在质量方面指挥和控制组织的协调活动，通常包括制定质量方针、目标以及质量策划、质量控制、质量保证和质量改进等活动。实现质量管理的方针目标，有效地开展各项质量管理活动，必须建立相应的管理体系，这个体系就叫质量管理体系。

## 质量管理体系的内涵

### 质量管理体系应具有符合性

欲有效开展质量管理，必须设计、建立、实施和保持质量管理体系。组织的最高管理者对依据ISO9001 国际标准设计、建立、实施和保持质量管理体系的决策负责，对建立合理的组织结构和提供适宜的资源负责；管理者代表和质量职能部门对形成文件的程序的制定和实施、过程的建立和运行负直接责任。

### 质量管理体系应具有惟一性

质量管理体系的设计和建立，应结合组织的质量目标、产品类别、过程特点和实践经验。因此，不同组织的质量管理体系有不同的特点。

### 质量管理体系应具有系统性

质量管理体系是相互关联和作用的组合体，包括：①组织结构——合理的组织机构和明确的职责、权限及其协调的关系；②程序——规定到位的形成文件的程序和作业指导书，是过程运行和进行活动

的依据；③过程——质量管理体系的有效实施，是通过其所需过程的有效运行来实现的；④资源——必需、充分且适宜的资源包括人员、资金、设施、设备、料件、能源、技术和方法。

### 质量管理体系应具有全面有效性

质量管理体系的运行应是全面有效的，既能满足组织内部质量管理的要求，又能满足组织与顾客的要求，还能满足第二方认定、第三方认证和注册的要求。

### 质量管理体系应具有预防性

质量管理体系应能采用适当的预防措施，有一定的防止重要质量问题发生的能力。

### 质量管理体系应具有动态性

最高管理者定期批准进行内部质量管理体系审核，定期进行管理评审，以改进质量管理体系；还要支持质量职能部门（含车间）采用纠正措施和预防措施改进过程，从而完善体系。

### 质量管理体系应持续受控

质量管理体系所需过程及其活动应持续受控。

### 质量管理体系应最佳化

组织应综合考虑利益、成本和风险，通过质量管理体系持续有效运行使其最佳化。

## 质量管理体系的特点



(一) 它代表现代企业或政府机构思考如何真正发挥质量的作用和如何最优地作出质量决策的一种观点。

(二) 它是深入细致的质量文件的基础。

(三) 质量体系是使公司内更为广泛的质量活动能够得以切实管理的基础。

(四) 质量体系是有计划、有步骤地把整个公司主要质量活动按重要性顺序进行改善的基础。

任何组织都需要管理。当管理与质量有关时，则为质量管理。质量管理是在质量方面指挥和控制组织的协调活动，通常包括制定质量方针、目标以及质量策划、质量控制、质量保证和质量改进等活动。实现质量管理的方针目标，有效地开展各项质量管理活动，必须建立相应的管理体系，这个体系就叫质量管理体系。它可以有效达到质量改进。ISO 9000 是国际上通用的质量管理体系。

## 质量管理体系的实施

### 采购标准

你在准备实施前，需要一本标准。你需要读懂，读通。

### 参考相关文献和软件

有许多有关质量出版物，软件工具帮助你理解，实施并注册质量管理体系。

### 组建队伍制订策略

你通过与最高管理层制订策略，组织策划全面实施体系。质量管理体系的职责在于高级管理层，所以在开始实施体系时需要高级经理参与。

### 考虑培训

无论是质量经理还是高级经理负责实施体系，

都需要提高 ISO 9001:2000, 总体意识。小组活动，研讨会和培训课可以起到帮助作用。

### 选择顾问

你可以得到保持中立的顾问建议如何更好地实施质量管理体系。他们有丰富的经验实施 QMS 并保证你少走弯路。

### 选择认证公司

认证公司是第三方机构，如 BSI。可以前往并有效地审核贵公司质量管理体系，如果符合标准，BSI 将颁发证书。由于种种市场原因，选择认证公司可能是一个复杂的过程。考虑的因素包括：工厂经验，地理范围，价格和服务水准。关键是找到最适合你需要的认证机构。找 BSI 你可能站的更高些。

### 撰写质量手册

质量手册是高级别文件，它要列出你对质量管理的要点。WHAT, WHY 和 HOW 在业务中实施质量管理体系。

### 建立支持性文件

建立程序文件以支持质量手册。清晰简练，列出为完成一项工作的要点，WHO, WHAT 和 HOW。

### 实施你的质量管理体系

实施的关键是沟通和培训。在实施阶段，所有执行程序的人都要收集记录已证明；规定的做到了，做到的符合规定。

### 预审核服务

预审核服务通常是在体系实施后 6 周进行。目的是找出哪些区域没能达到标准。这将使你在初审之前考虑改进的方向。



### 获得认证

你与你的认证机构安排初审。在此阶段认证机构将审核你的质量管理体系，并建议是否发证。

### 后续审核

一旦你获得认证并拿到证书，你就可以对外宣传你的企业已成功获得认证。为保证认证资格你需要继续实施所有质量体系。认证机构定期对标准执行情况要进行检查。

### 质量管理体系的运行机制

质量管理体系的运行机制可分为驱动层（包括中国合格评定国家认可委员会 CNAS、中国认证人员与培训机构认可委员会 CCAA、认证机构、审核员）、目标层（企业的质量管理体系）和自我改进层（包括最高管理者、管理者代表、内审员）三层。

(1) 驱动层。驱动层及第三方的质量审核。质量审核的监督机制体现在审核的独立性、公正性、系统性、权威性和持续性上。

(2) 目标层。组织通过建立、实施和运行质量管理体系，来实现质量方针和质量目标，进而满足顾客、相关方的需求和期望以及法律法规的要求。

(3) 自我改进层。2000 版标准的基本结构是以 PDCA 循环为指导的不断改进的过程。为了持续地保持认证资格，组织在体系建立运行中要做到“四落实”，即组织落实、活动和资源落实、实施程序落实、目标落实。组织要通过管理评审、内审、顾客满意度、不合格控制以及测量分析与改进、纠正、预防措施等过程，形成持续改进的运行机制。

### 质量管理体系建立的步骤

建立、完善质量体系一般要经历质量体系的策

划与设计，质量体系文件的编制、质量体系的试运行，质量体系审核和评审四个阶段，每个阶段又可分为若干具体步骤。

### 质量体系的策划与设计

该阶段主要是做好各种准备工作，包括教育培训，统一认识，组织落实，拟定计划；确定质量方针，制订质量目标；现状调查和分析；调整组织结构，配备资源等方面。

#### 一、教育培训，统一认识

质量体系建立和完善的过程，是始于教育，终于教育的过程，也是提高认识和统一认识的过程，教育培训要分层次，循序渐进地进行。

第一层次为决策层，包括党、政、技（术）领导。主要培训：

1. 通过介绍质量管理和质量保证的发展和本站的经验教训，说明建立、完善质量体系的迫切性



和重要性；

2. 通过 ISO9000 族标准的总体介绍，提高按国家（国际）标准建立质量体系的意识。

3. 通过质量体系要素讲解（重点应讲解“管理职责”等总体要素），明确决策层领导在质量体系建设中的关键地位和主导作用。

第二层次为管理层，重点是管理、技术和生产部门的负责人，以及与建立质量体系有关的工作人员。

这二层次的人员是建设、完善质量体系的骨干力量，起着承上启下的作用，要使他们全面接受 ISO9000 族标准有关内容的培训，在方法上可采取讲解与研讨结合。第三层次为执行层，即与产品质量形成全过程有关的作业人员。对这一层次人员主要培训与本岗位质量活动有关的内容，包括在质量活动中应承担的任务，完成任务应赋予的权限，以及造成质量过失应承担的责任等。

## 二、组织落实，拟定计划

尽管质量体系建设涉及到一个组织的所有部门和全体职工，但对多数单位来说，成立一个精干的工作班子可能是需要的，根据一些单位的做法，这个班子也可分三个层次。

第一层次：成立以最高管理者（总经理等）为组长，质量主管领导为副组长的质量本体系建设领导小组（或委员会）。其主要任务包括：

1. 体系建设的总体规划；
2. 制订质量方针和目标；
3. 按职能部门进行质量职能的分解。

第二层次，成立由各职能部门领导（或代表）参加的工作班子。这个工作班子一般由质量部门和计划部门的领导共同牵头，其主要任务是按照体系建设的总体规划具体组织实施。

第三层次：成立要素工作小组。根据各职能部门的分工明确质量体系要素的责任单位，例如，“设计控制”一般应由设计部门负责，“采购”要素由物资采购部门负责。组织和责任落实后，按不同层次分别制定工作计划，在制定工作计划时应注意：

1. 目标要明确。要完成什么任务，要解决哪些主要问题，要达到什么目的？

2. 要控制进程。建立质量体系的主要阶段要规定完成任务的时间表、主要负责人和参与人员、以及他们的职责分工及相互协作关系。

3. 要突出重点。重点主要是体系中的薄弱环节及关键的少数。这少数可能是某个或某几个要素，也可能是要素中的一些活动。

## 三、确定质量方针，制定质量目标

质量方针体现了一个组织对质量的追求，对顾客的承诺，是职工质量行为的准则和质量工作的方向。制定质量方针的要求是：

1. 与总方针相协调；
2. 应包含质量目标；
3. 结合组织的特点；
4. 确保各级人员都能理解和坚持执行。

## 四、现状调查和分析

现状调查和分析的目的是为了合理地选择体系要素，内容包括：



1. 体系情况分析。即分析本组织的质量体系情况，以便根据所处的质量体系情况选择质量体系要素的要求。

2. 产品特点分析。即分析产品的技术密集程度、使用对象、产品安全特性等，以确定要素的采用程度。

3. 组织结构分析。组织的管理机构设置是否适应质量体系的需要。应建立与质量体系相适应的组织结构并确立各机构间隶属关系、联系方法。

4. 生产设备和检测设备能否适应质量体系的有关要求。

5. 技术、管理和操作人员的组成、结构及水平状况的分析。

6. 管理基础工作情况分析。即标准化、计量、质量责任制、质量教育和质量信息等工作的分析。

对以上内容可采取与标准中规定的质量体系要素要求进行对比性分析。

## 五、调整组织结构，配备资源

因为在一个组织中除质量管理外，还有其他各种管理。组织机构设置由于历史沿革多数并不是按质量形成客观规律来设置相应的职能部门的，所以在完成落实质量体系要素并展开成对应的质量活动以后，必须将活动中相应的工作职责和权限分配到各职能部门。

一方面是客观展开的质量活动，一方面是人为的现有的职能部门，两者之间的关系处理，一般地讲，一个质量职能部门可以负责或参与多个质量活动，但不要让一项质量活动由多个职能部门来负责。目前我国企业现有职能部门对质量管理活动所承担的职责、所起的作用普遍不够理想总的来说应该加强。在活动展开的过程中，必须涉及相应的硬件、软件和人员配备，根据需要应进行适当的调配和充实。

## 质量体系文件的编制

质量体系文件的编制内容和要求，从质量体系的建设角度讲，应强调几个问题：

1. 体系文件一般应在第一阶段工作完成后才正式制订，必要时也可交叉进行。如果前期工作不做，直接编制体系文件就容易产生系统性、整体性不强，以及脱离实际等弊病。

2. 除质量手册需统一组织制订外，其它体系文件应按分工由归口职能部门分别制订，先提出草案，再组织审核，这样做有利于今后文件的执行。

3. 质量体系文件的编制应结合本单位的质量职能分配进行。按所选择的质量体系要求，逐个展开为各项质量活动（包括直接质量活动和间接质量活动），将质量职能分配落实到各职能部门。质量活动项目和分配可采用矩阵图的形式表述，质量职能矩阵图也可作为附件附于质量手册之后。

4. 为了使所编制的质量体系文件做到协调、统一，在编制前应制订“质量体系文件明细表”，将现行的质量手册（如果已编制）、企业标准、规章制度、管理办法以及记录表式收集在一起，与质量体系要素进行比较，从而确定新编、增编或修订质量体系文件项目。

5. 为了提高质量体系文件的编制效率，减少返工，在文件编制过程中要加强文件的层次间、文件与文件间的协调。尽管如此，一套质量好的质量体系文件也要经过自上而下和自下而上的多次反复。

6. 编制质量体系文件的关键是讲求实效，不走形式。既要从总体上和原则上满足 ISO9000 族标准，又要在方法上和具体做法上符合本单位的实际。

## 质量体系的试运行

质量体系文件编制完成后，质量体系将进入试运行阶段。其目的，是通过试运行，考验质量体系

文件的有效性和协调性，并对暴露出的问题，采取改进措施和纠正措施，以达到进一步完善质量体系文件的目的。在质量体系试运行过程中，要重点抓好以下工作：

1. 有针对性地宣贯质量体系文件。使全体职工认识到新建立或完善的质量体系是对过去质量体系的变革，是为了向国际标准接轨，要适应这种变革就必须认真学习、贯彻质量体系文件。

2. 实践是检验真理的唯一标准。体系文件通过试运行必然会出现一些问题，全体职工立将从实践中出现的问题和改进意见如实反映给有关部门，以便采取纠正措施。

3. 将体系试运行中暴露出的问题，如体系设计不周、项目不全等进行协调、改进。

4. 加强信息管理，不仅是体系试运行本身的需要，也是保证试运行成功的关键。所有与质量活动有关的人员都应按体系文件要求，做好质量信息的收集、分析、传递、反馈、处理和归档等工作。

### 质量体系的审核与评审

质量体系审核在体系建立的初始阶段往往更加重要。在这一阶段，质量体系审核的重点，主要是验证和确认体系文件的适用性和有效性。

1. 审核与评审的主要内容一般包括：
  - (1) 规定的质量方针和质量目标是否可行；
  - (2) 体系文件是否覆盖了所有主要质量活动，各文件之间的接口是否清楚；
  - (3) 组织结构能否满足质量体系运行的需要，各部门、各岗位的质量职责是否明确；
  - (4) 质量体系要素的选择是否合理；
  - (5) 规定的质量记录是否能起到见证作用
  - (6) 所有职工是否养成了按体系文件操作或工

作的习惯，执行情况如何。

2. 该阶段体系审核的特点是：

(1) 体系正常运行时的体系审核，重点在符合性，在试运行阶段，通常是将符合性与适用性结合起来进行；

(2) 为使问题尽可能地在试运行阶段暴露无遗，除组织审核组进行正式审核外，还应有广大职工的参与，鼓励他们通过试运行的实践，发现和提出问题；

(3) 在试运行的每一阶段结束后，一般应正式安排一次审核，以便及时对发现的问题进行纠正，对一些重大问题也可根据需要，适时地组织审核；

(4) 在试运行中要对所有要素审核覆盖一遍；

(5) 充分考虑对产品的保证作用；

(6) 在内部审核的基础上，由最高管理者组织一次体系评审。

应当强调，质量体系是在不断改进中行以完善的，质量体系进入正常运行后，仍然要采取内部审核，管理评审等各种手段以使质量体系能够保持和不断完善。

### 参考文献

韩福荣. 现代质量管理学. 机械工业出版社, 2007.08. [d](#)



# 开发 DDE 的屏幕保护程序

● 深度科技 武汉公司 研发中心 / 文

## 前言

还记得刚推出屏幕保护功能那会儿，小编偶逛论坛，围观大家对这个功能的评价。其中让我印象最深的一句话就是：“一股 Windows98 风”，总之，评价总结出来就是一个字：“吃藕（丑）”。大家追求美好事物的诚挚之心深深地打动了我，而且，小编个人做事情稍有些未雨绸缪，在屏幕保护程序开发之初，就已经定好了易于扩展的架构，所以我当时就下定了决心，为大家开发一个非 Windows98 风格的屏保。为了达到绝对“非 Windows98”的目的，小编特意选择了 Windows10 中的默认屏保作为参考，在无数个周末的战斗下，最终成功将名为“泡泡”的屏保应用发布到了商店（项目地址：<https://github.com/zccrs/screensaver-pp>）。本着“授人以鱼不如授人以渔”的理念，小编特地整理了这篇文章协助大家开发一款属于自己的时尚屏保应用。

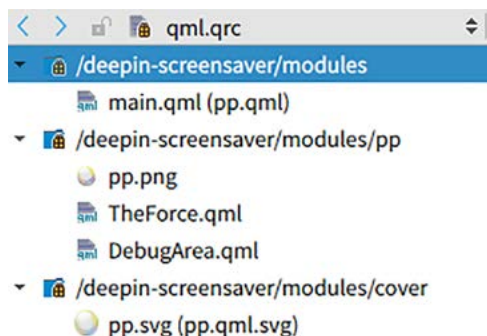
## 正文

在 Linux+X11 生态环境中，xscreensaver 是最“流行”的屏幕保护程序，有着非常多的屏保资源，所以 deepin-screensaver 必然要兼容它的资源。

但是，xscreensaver 对屏保资源的扩展方式并不符合 deepin 的开发理念，因此，deepin-screensaver 实现了一套全新的屏保扩展方式。

支持使用 Qt qml 模块编写屏保应用，一个标准的屏保应用只需要包含一个“xx.rcc”文件，将文件安装到 /usr/lib/deepin-screensaver/resources 目录。

rcc 格式是一个编译之后的 Qt 资源文件，在这个资源文件中至少要包含两个文件：qml 代码文件、屏保封面图。



图中文件名括号内为其别名，也就是屏保主应用加载文件时能读取到的文件名。

qml 代码文件：屏保应用的代码入口，会被屏保主程序加载显示屏保封面图；设置屏保入口显示的预览图，支持 svg png jpeg bmp 等格式所有的文件必须以特定的目录结构组织到一个 Qt 资源文件（qrc 文件），以“泡泡”屏保为例：qml.qrc 为其资源文件，包含三个前缀路径。

·/deepin-screensaver/modules：放置屏保应用的主 qml 文件，此路径下的所有 qml 文件都会被当做一个独立的屏保应用，因此，项目中的其它文件需要额外建立新的前缀放置

·/deepin-screensaver/modules/cover：放置屏保应用封面图文件，文件名称必须和 modules 目录中的 qml 文件一致，且包含它的“.qml”后缀。如

图上, qml 文件全名为: ” pp.qml” , 封面图全名为: ” pp.qml.svg” 。

·/deepin-screensaver/modules/pp: 此前缀不是必须的, 用于放置项目中的其它文件。为了不与其它项目产生冲突, 建议使用项目名作为目录名称

资源文件最好以项目名称命名, 避免和其它屏保应用冲突。另外, 大家可能已经发现了, 这三个前缀都有一个共同点, 那就是以 “/deepin-screensaver/modules” 开头, 的确, 这是一个格式要求, 不能随意更改路径。

主 qml 文件作为屏保应用的入口, 它的根元素一定要设置:

```
anchors.fill: parent
```

这样才能确保屏保应用充满整个屏幕。在多屏的情况下, 会创建多个窗口示例, 可根据屏幕绘制不同的屏保内容。

项目编译其实很简单, 只需要使用 Qt 提供的 rcc 命令将 qrc 文件编译为 rcc 文件即可, 使用 qmake 构建系统时, 可以在 pro 文件中调用以下命令:

```
system(rcc --binary $$_PRO_FILE_PWD_/xx.qrc -o $$_PRO_FILE_PWD_/xx.rcc)
```

当然, 最后不要忘记将 xx.rcc 文件安装到 deepin-screensaver 所要求的目录。做完这所有的步骤后, 回到桌面, 在右键菜单中选择 “壁纸与屏保”, 切换到屏保设置后即可看到新添加的屏保应用。

另外, deepin-screensaver 为 qml 提供了获取当前屏幕截图的接口, 只需要为 Image 项指定特定

的路径即可:

```
Image {
    anchors.fill: parent
    source: "image://deepin-screensaver/screen/" + Screen.name
}
```

由于要获取屏幕名称, 上述代码需要 “import QtQuick.Window 2.2” 使用。

## 后记

屏保封面图最佳比例为: 8:5, 推荐使用 svg 格式, 以更好的适应高分屏缩放。

推荐大家使用 Qt Creator 作为项目的开发工具, 可以方便的编辑 qrc 文件。

泡泡屏保是一个完整的 demo, 有任何疑问的地方都可以以其作为参考。

## 参考

“泡泡” 屏保项目: <https://github.com/zccrs/screensaver-pp>

Qt 资源文件: <https://doc.qt.io/qt-5/resources.html>



# electron 初探

● 深度科技 武汉公司 研发中心 / 文

## 前言

额，首先感谢一位集各种灵巧走位和花式甩锅技巧为一身的萨克斯大神，不然本人绝对接不到如此简单明了（丧心病狂）的任务；由于是第一次接到这样任务，第一件事肯定是去查找之前大神们留下的文档以防撞车，绝对不是抱着如果有模板就去抄一份的龌龊思想，然后…感谢各位大神，如果不是在细细品读（伺机抄袭）的过程中被虐的死去活来的话，我现在也不会有思路写下这篇文章，讲真好多文章都是开头带着读者：来，这是 1，这是 2，1+2 等于几啊？ 3！哎！真棒！然后不知道哪段，突然：来，我们看一下微积分，这边先来求个导…您们是魔鬼吗！？这还不算那些开头就开始傅里叶变幻的，小白膝盖骨都碎成渣了好嘛…此处应配图：

大神们好，我是新来的渣渣



## 正篇，简单介绍

Electron 基于 Chromium 和 Node.js，让用户可以使用 HTML, CSS 和 JavaScript 构建应用，并且 Electron 兼容 Mac, Windows 和 Linux，它构建的应用可在这三个操作系统上面运行，当然了它是开源的。

## 环境搭建

这里只介绍 deepin 系统环境的搭建方法，

windows 或 mac 百度上太多了，而且绝对比我的全。

## 安装 nodejs 和 npm

```
sudo apt-get install nodejs npm
```

检查方法：

```
node -v 和 npm -v
```

看见了版本号基本上就 OK。

## 安装 cnpm (国内的 npm)

用淘宝的源下载 cnpm。

```
sudo npm install -g cnpm --registry=https://registry.npm.taobao.org
```

注：-g 表示全盘安装，sudo 最好加上不然会报权限不足的错误 (root 用户当我没说)

## 安装 electron

```
sudo cnpm install electron electron-forge -g
```

## 编辑器

首推使用 vscode (宇宙无敌超级至尊)，记得安装一些 html 相关的插件，当然不装也没关系，之后写代码他会在右下角弹出推荐你安装的插件，点一下就好。

## 手动创建一个项目

创建一个目录，地点随意，我是在家目录



```
mkdir ~/work
```

在 vscode 里新建一个 index.html 文件  
内容随便，这里我填了一个 HTML5 的模板：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>electron</title>
</head>
<body>
  hello electron
</body>
</html>
```

接下来创建 main.js 文件，内容如下：

```
//electron 对象的引用
var electron = require('electron');
var app=electron.app;
//BrowserWindow 类的引用
var BrowserWindow=electron.BrowserWindow;

var mainWindow=null;
// 监听应用准备完成的事件
app.on('ready',function(){
  // 创建窗口
  mainWindow=new BrowserWindow({width:400,height:400});
  mainWindow.loadFile('index.html');// 选择要打开的主页文件
  mainWindow.on('closed',()=>{// 监听了一个关闭事件
    mainWindow=null;
  })
})
```

然后在命令行里输入 `npm init --yes`  
之后可以发现多了一个 package.json 文件  
内容如下：

```
{
  "name": "你的文件夹名",
  "version": "1.0.0",
  "description": "",
  "main": "main.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "babel-preset-env": "^1.7.0"
  }
}
```

```
}
}
```

尤其要注意，看看你的 main 是否指向 main.js。

之后就可以运行试试看了，执行 `electron`。

### 这里注意，大概率会报错：

（因为一直是手动构建项目，所以还有很多 `electron` 的依赖没有添加到项目中）

遇到：

```
App threw an error during load
Error: Couldn't find preset "env" relative to directory...
```

需要在当前文件夹内执行：

```
npm install babel-preset-env --save-dev
```

遇到：

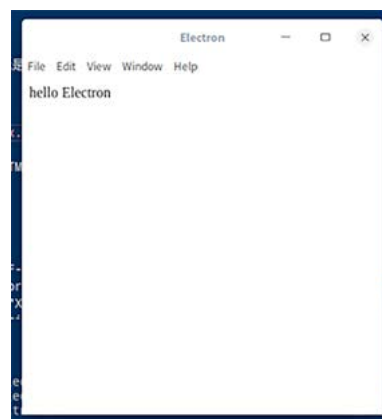
```
App threw an error during load
Error: Couldn't find preset "react" relative to directory ...
```

执行：

```
npm install react react-dom
npm install babel-core babel-loader babel-preset-react
npm install webpack
```

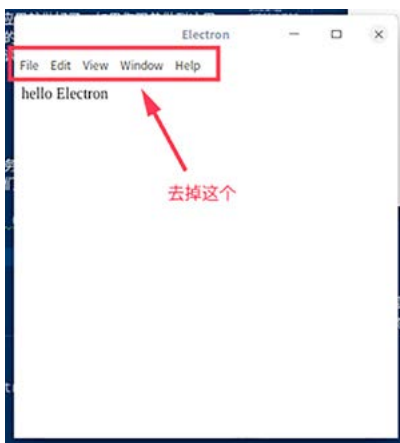
最后执行 `electron`。

应该看到下图的界面：





想去掉上面的那行菜单栏（见图片）



在此之前先把 vscode 中对 electron 的代码提示弄出来，在 work 目录下执行：

```
npm install electron --save
```

这样表示把 electron 下载到当前目录下的 node\_modules 文件夹中了，之后就会有代码提示，在 work 目录下新建一个 main 文件夹，在 main 文件夹里新建一个 menu.js，内容为：

```
//electron 对象的引用
var electron=require('electron');
//electron.Menu 对象的引用
var Menu=electron.Menu;
// 创建一个空菜单
let template=[];
// 实例化
var m=Menu.buildFromTemplate(template);
// 传值
Menu.setApplicationMenu(m);
```

然后在 main.js 的 app.on('ready',function){ 这行下面添加：

```
// 执行创建菜单的操作
require('./main/menu.js');
```

像这样：

```
1 //electron对象的引用
2 var electron = require('electron');
3 var app=electron.app;
4 //BrowserWindow类的引用
5 var BrowserWindow=electron.BrowserWindow;
6
7 var mainWindow=null;
8 //监听应用准备完成的事件
9 app.on('ready',function(){
10 //执行创建菜单的操作
11   require('./main/menu.js');
12 //创建窗口
13   mainWindow=new BrowserWindow({width:400,height:400});
14   mainWindow.loadFile('index.html');//选择要打开的
15   mainWindow.on('closed',()=>{//监听了一个关闭事件
16     mainWindow=null;
17   })
18 }
19 });
```

之后再执行 electron . 应该会看到下面的样子：



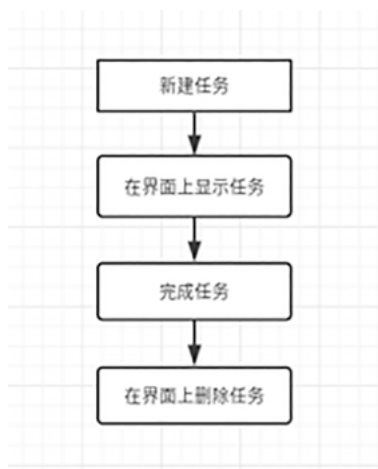
这样就成功了。

### 休息

至此一个最简单的 electron 应用就做好了，如果您跟着做到这里了，建议休息一下，喝杯茶啥的活动活动，放心本文不含任何的防不胜防，我会尽量做到代码粘到读者那就能用，请放心往下看。

### 任务管理软件需求

这里要做的是一个最基本的任务管理，那么最基本的需求就是添加任务和删除任务了。流程大概是这种感觉：



## 正式开始

之前介绍里讲了 electron 可以算是用了浏览器的技术来将网站封装成桌面应用，所以网页的制作占了很大的部分，接下来会用到的其他前端知识我都会写注释，目标是小白可以复刻出这个程序，用到的知识可以之后再补，但兴趣这东西是说没就没啊...

## 选择所需的技术

这里选择的是 vue 框架以及 element-ui 这个 ui 组件。

## 准备工作

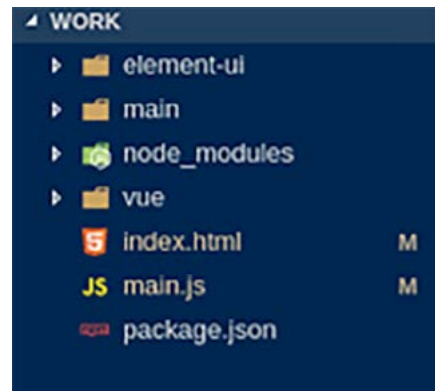
首先下载 vue 和 element-ui，在当前目录下执行：

```

cnpm install vue --save
cnpm install element-ui --save
cp -r node_modules/_vue* ./vue
cp -r node_modules/_element-ui* ./element-ui
  
```

这样在当前目录里就可以得到 element 和 vue

这两个文件夹，当前的工作区应该是这样：



## 实现

根据需求来的话需要一个输入框和一个显示区域，其他就是逻辑层面的对每一步操作进行数据绑定来实现想要的效果。

废话不多说直接上代码，将 index.html 的内容修改为如下：

```

<!DOCTYPE html>
<html lang="en" >

<head>
  <meta charset="UTF-8">
  <title> 个人任务管理 </title>
  <!-- 这里引用 element-ui 的样式 -->
  <link rel="stylesheet" href="./element-ui/lib/theme-chalk/index.css">
  <!-- 定义一个样式用于 窗口的背景色 -->
  <style>
    .box {
      background:linear-gradient(to right, #7f57ec 10%,#00b3ffb9 90%);
    }
  </style>
</head>
<!-- 这里调用了背景的背景的样式 -->
<body class="box" >
  <!-- 这里引用的是 vue 和 element-ui 组件的 js 文件，基本用于实现逻辑和 element-ui 的一些动态效果 -->
  <script src="./vue/dist/vue.js"></script>
  <script src="./element-ui/lib/index.js"></script>
  <!-- 这里调用了 vue 的一个类似于函数块一样的东西，我姑且是这样理解的，块名称为 add-->
  <div id="add" >
    <!-- 这里就是一个输入框了，v-model 是 vue 用来绑定数据的，tag 是在 add 块里面定义的一个变量，placeholder=" 在这添加，回车结束 " 表示的是给用户的默认提示，
  
```



```

v-on 后面绑定的是事件，这里绑定了一个回车事件，当用户在这个输入框内输入回车时会触发 add 块里定义好的 add_tag 函数，函数会执行相应的数据操作 -->
<el-input v-model="tag" placeholder="在这添加，回车结束" v-on:keyup.enter.native="add_tag" ></el-input>
</div>
<!-- 硬核空一行 -->
<br>
<!-- 于上面类似调用一个名为 app 的函数块 -->
<div id="app">
<!-- 这里定义了一个表格，数据绑定的是 app 块里面的 tabledata 的这个数组对象，宽度设置为 100%，label 表示列的标题 -->
<el-table
  :data="tableData"
  style="width: 100%"
  >
  <el-table-column
    label="任务"
    prop="renwu"
  >

  </el-table-column>
  <el-table-column
    fixed="right"
    label="操作"
    width="100">
  <!-- 这里是完成按钮的实现，会帮助用户删除完成的那行数据 -->
  <template slot-scope="scope">
    <el-button
      @click.native.prevent="deleteRow(scope.$index, tableData)"
      type="primary"
      icon="el-icon-check"
      size="small" plain> 完成
    </el-button>
  </template>
  </el-table-column>
</el-table>
</div>

</body>
<script>
// 这里就是定义函数块的地方，el 后表示的是这个块的名字，也就是能被 id 调用的东西
var app=new Vue({
  el:'#app',
  methods:{
    // 这是完成任务的函数
    deleteRow(index, rows) {
      rows.splice(index, 1);
      // 调用了完成的提示
      this.tishi_finsh();
    },
  },

```

```

// 定义了一个完成的提示信息，showclose 表示可以被关闭，message 表示信息，type 其实是 element-ui 的样式，duration 表示提示出现的时间以毫秒为单位，这里表示出现 800 毫秒后关闭
tishi_finsh() {
  this.$message({
    showClose: true,
    message: '恭喜完成任务',
    type: 'success',
    duration:800
  });
},
// 这里定义是任务的内容，每一次添加其实都是给这个数组里面添加对象
data:{
  tableData: [{
    renwu: '活着',
  }]
}
// 添加任务块
var add=new Vue({
  el:'#add',
  data:{
    // 这个变量是和输入框的内容进行绑定的
    tag:''
  },
  methods: {
    // 当这个函数触发时就把输入框的内容加到任务数据的数组里面去
    add_tag(){
      app.tableData.push({renwu:this.tag});
      this.tag="";
      this.tishi_add();
    },
    // 添加任务成功的提示框
    tishi_add() {
      this.$message({
        showClose: true,
        message: '添加成功',
        type: 'success',
        duration:500
      });
    }
  }
});
</script>
</html>

```

这时执行 electron . 的效果应该是这样的：



### 后记:

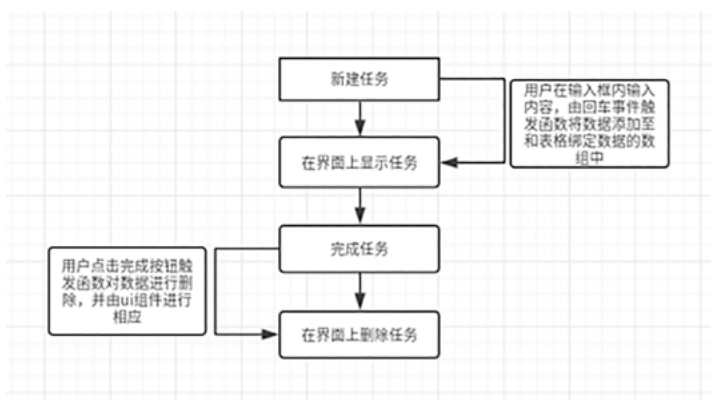
参考文档:

VUE 官方文档

element-ui 官方文档

electron 官方文档 [d](#)

实现思路是差不多是这样的:





# 玩转 Remmina 远程办公 So easy !

◎ 深度用户 myccloves / 文

由于我每天需要接收上级文件通知，我们有专门的公文管理系统（windows7 系统），在 windows 平台工作时，每天需要远程登录这台计算机查看是否有通知，对于深度系统是否也有这样的远程功能呢？当然有，通过 Remmina 软件即可实现。

## 一、安装软件

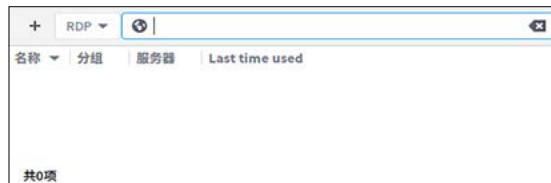
Remmina 是一款远程桌面客户端软件，提供了多种远程连接协议。打开深度商店，在搜索栏中输入：Remmina，搜索结果如下：



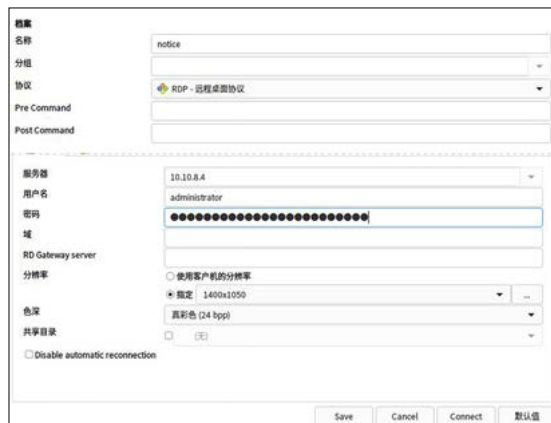
单击安装即可（深度商店一键安装，方便好用）。

## 二、配置

在用的公文管理系统的计算机有远程桌面服务，端口号：3389（默认就是这个），IP 地址为：10.10.8.4，打开 Remmina，之后单击“+”。



之后输入如下信息：



名称：连接的名称（自己随意起）

协议：由于是 windows 远程桌面，所以用“RDP-远程桌面协议”

服务器：远程 windows 的 IP 地址：10.10.8.4

用户名：远程 windows 的用户名，我用的是管

理员账号：administrator

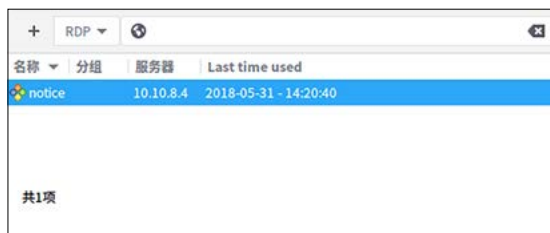
密码：远程 windows 的密码

分辨率：可以根据实际情况指定

最后保存就可以了。

### 三、登录

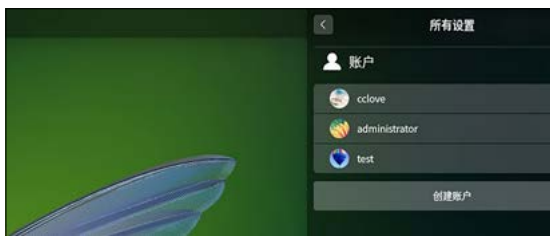
当需要查看通知时，可以在启动器中找到 Remmina，启动之后发现在下面的列表中有我们添加的配置，双击即可。



### 四、windows 登录深度桌面系统

当看到不一样的 deepin 系统会被围观，可以给同事分配一个账号，让他也登录深度系统尝尝鲜。

好吧,说干就干! 首先给同事创建用户, 打开“启动器”>“系统管理”>“控制中心”, 单击“账户”, “创建账户”, 分配用户名和密码。



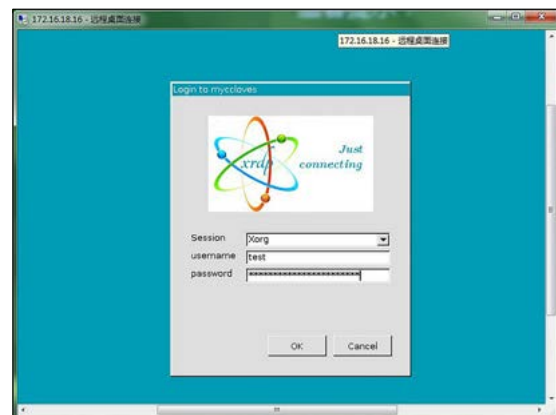
其次安装 xrdp 软件, 我在商店里没有找到, 所以需要从终端安装, ctrl + alt + t 打开终端, 输入:

```
sudo apt install xrdp
```

安装完成, 可以在 windows 主机上登录深度系统了。



在 windows 上打开远程桌面, 输入主机 IP 地址: 172.16.18.16, 之后会出现登录界面, 输入用户名和密码。



登录成功!! 现在是 ShowTime…… ^\_^

你学会了吗? 赶快上手体验一下吧! [d](#)



# 在 linux Deepin 深度系统 安装多媒体电子教室 Veyon

● 深度用户 qiqunet / 文

linux 操作系统的多媒体教学神器 Veyon (原 iTalc) 是一个强大的课堂教学环境。借助该工具, 教师可以查看和控制学生桌面。veyon 允许教师查看学生桌面上发生了什么, 控制、锁定学生桌面, 向学生演示、发送文件、文字消息, 打开或关闭学生电脑等等。

程序功能和界面截图:

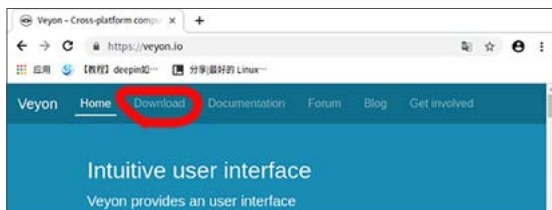


但安装及第一次使用的过程实在有点儿太难了, 至少对初次接触 Linux 的人来说真的不容易, 尤其是英文水平也太一般的情况下。

为了给其他如我一样的菜鸟节约那么一点点时间, 特意写了下面的这个教程:

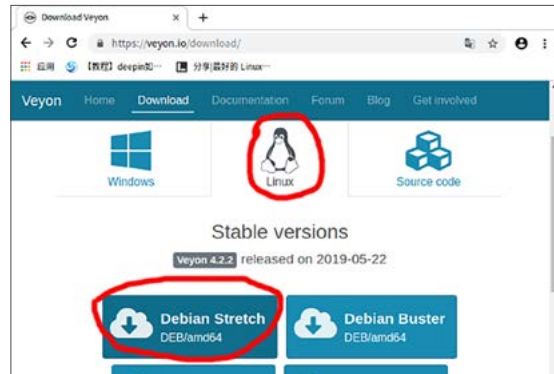
## 一、程序的下载

软件可从 <https://veyon.io> 网站上下载, 下载编译好的企鹅程序就可以了。



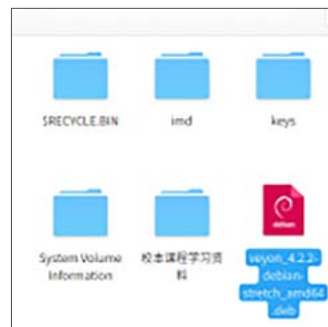
进入网站 <https://veyon.io>, 网页显示完毕后, 进入 Download 页面。

实践证明 Debian Stretch 是适合深度系统的:



## 二、veyon 系统的安装

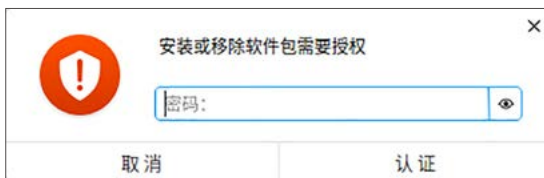
教师机和学生机的安装方法是一样的——直接双击程序文件即进入安装步骤:







单击“安装”按钮后出现授权窗口：



输入深度系统当前用户的登陆密码即开始安装：



安装完毕后点击“完成”按钮。

### 三、教师机 / 学生机共有的软件应用前设置

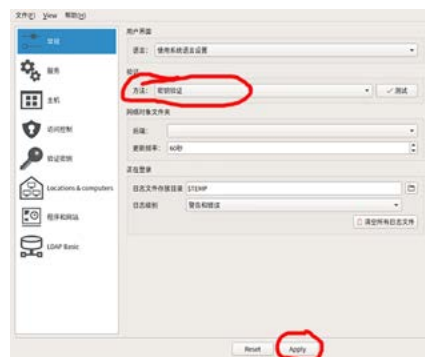
程序安装完成后，即可进入软件设置界面对软件进行设置，设置时先设置教师机。

点击深度系统开始菜单，找到“Veyon Configurator”图标，双击打开设置界面，



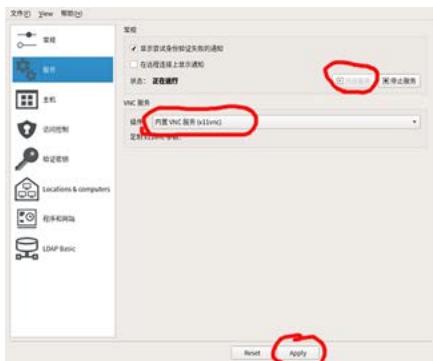
输入当前用户的登陆密码，即进入到设置界面：

“常规”页设置使用“密钥验证”的方法，设置完后点击“应用”按钮：





“服务”页设置使用“内置 VNC 服务”插件，并启动服务，设置完后点击应用按钮：



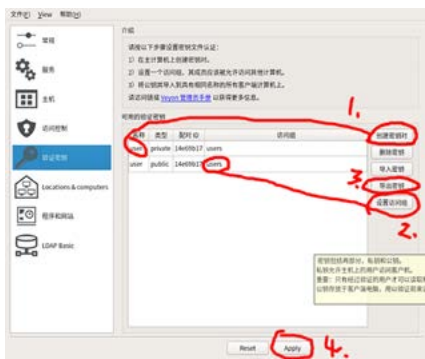
#### 四、教师机的设置：

进入“验证密钥”页面，创建一对密钥对，名称似乎自定即可，我这里设置成与 PC 的登陆用户名一致了，因为我也是初用者，不确定命名规定。

设置访问组为“users”，具体选哪个访问组请看设置界面的第 2 点说明。

设置完成后，选中第二行的 Public（公钥），并点击“导出密钥”按钮。

将导出的密钥文件用 U 盘拷贝到学生机，并导入到学生机上去。

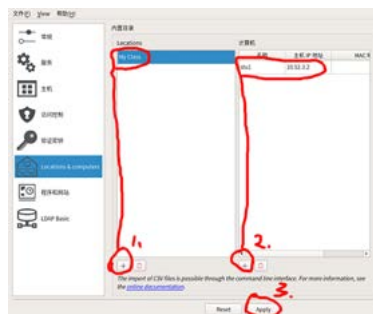


进入“本地计算机列表 (Locations & Computers)”页面。

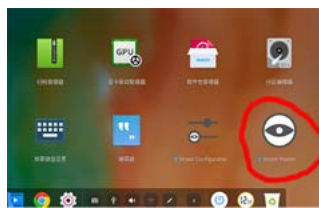
点击左边的“+”号添加一组计算机列表，并输入自定义的名称。

点击右边的“+”号添加一部学生机，并输入自定义的名称和该学生机的 IP 地址（好像没有找到批量导入的接口，软件好像也不能自动发现学生机，只能这样一行行地去加学生机，因为我也只是初次接

触，可能有不对的地方）；

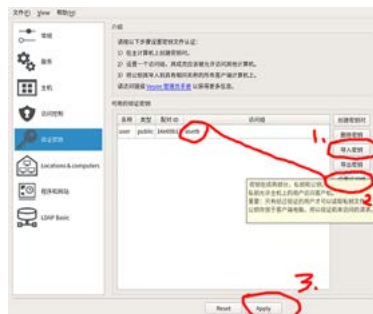


上述设置完毕后，可以关闭教师机的设置界面了，打开深度的开始菜单，击开 veyon master 图标，教师机进入应用界面：



#### 五、学生机的设置

先在文件管理器中将存放在 U 盘中的来自教师机的公钥文件转存到本地磁盘（例如我的文档文件夹），在学生机的设置界面中进入“验证密钥”页面，首先点击“导入密钥”按钮，并导入公钥；然后选中已导入的密钥，并点击“设置访问组”按钮，进而选择学生机的登陆用户名用户即可。本文到此结束，希望对大家有用。



Veyon 软件的在线英文使用说明：

<https://veyon.readthedocs.io/en/latest/admin/index.html>

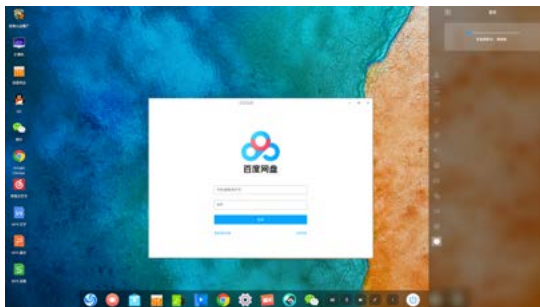
# 忘掉 Windows 吧！ Deepin 操作系统深度体验

● 中关村在线 李健 / 文

21 世纪，人类在办公、学习、娱乐、移动通信等各个领域都离不开操作系统。无论是桌面版的 Windows、Mac 还是移动端的 iOS、Android 都是我们日常所能接触到的最常用的操作系统，他们无一例外全部来自于美国。

如果说原来谈论中国团队研发的 Linux 操作系统还停留在纸面上的话，随着最近中美贸易摩擦产生的一系列事件，让国人又一次对芯片、操作系统等高度重视起来。今天我们就来讨论一下，没了 Windows，我们有没有一套完整的系统可以使用呢？

我们都知道，操作系统的打造其实并不是特别难，最难的是各种软硬件支持，比方说显卡驱动适配、办公应用软件开发、游戏程序等等。其实 Linux 操作系统确实不少，但大多数都是只有系统这幅“壳子”而没有应用的“灵魂”。而真要说可堪一用，生态建立良好的系统，呼声最高的应该就是 Deepin 深度操作系统了。



深度操作系统桌面

深度 Deepin 操作系统是基于 Linux 内核，以桌面应用为主的开源 GNU/Linux 操作系统，支持笔记本、台式机和一体机。此款操作系统无论是在交互逻辑，应用方式还是使用习惯上都和我们目前常用的系统没有太大差别，下面我就带大家体验一下，并说说我的使用感受。

## 下载与安装

下载与安装方式和 Windows 差不多，首先我们需要去深度科技的官网下载系统镜像文件 (<https://www.deepin.org/download/>)，我这里下载的是最新版本 V15.10，然后用官方提供的 U 盘制作程序制作出一个系统盘。仅仅需要选择系统——格式化 U 盘——制作这三步即可，非常方便。



深度科技官网 可以在这里下载所有文件

哦对了，忘了说，所有系统和软件都是免费的哦。

制作好 U 盘后，我们只需要和 Windows 一样插



入 U 盘，重启电脑后就可实现安装，只要你会安装系统这个对你肯定非常简单。之后我们就可以进入桌面了。



桌面样式可自由切换

进入系统后我们可以看到其非常的简洁，清爽，交互逻辑几乎和 Windows 没有区别，我花了 5 分钟就上手了。这之后，我们就要迎来第一步考验：驱动。在这里，我测试了一下显卡驱动的安装。



深度显卡驱动管理器（标红框处）

首先要说明的是，深度操作系统是有自己的官方商店的，我们日常的办公软件、浏览器、游戏等软件都可以在这里面免费下载，类似于 Mac 系统。我们在系统管理处找到显卡驱动管理器进行安装。

在这之后驱动程序会自己识别显卡种类（NV、

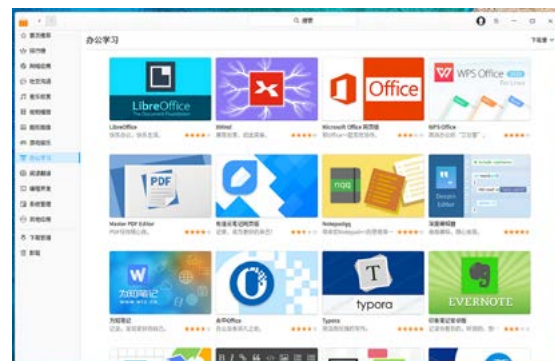
AMD、Intel）进行安装，你可以选择开源（性能较差）或闭源（性能好但兼容性差）的驱动程序。这里我实测了一下，较新的显卡例如 GTX 1650 还是没有办法识别的，不过类似于 GTX 1080 等稍早一些的显卡可以完美兼容。



GTX 1080 可完美支持

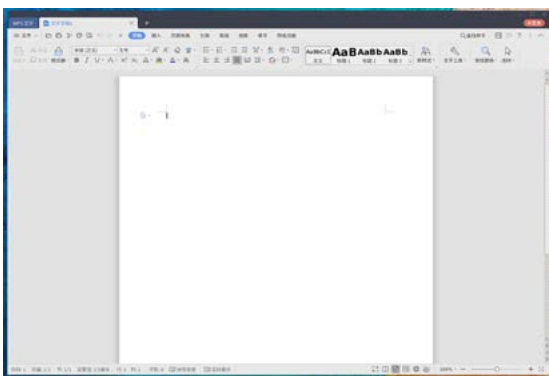
### 日常应用体验：办公

前面说过，进入系统只是第一步，一个操作系统是否能大范围传播靠的是应用。首先我们来看一下办公软件。



部分办公学习应用

在应用商店“办公学习”一栏中，我们能发现很多熟悉的面孔，比如：LibreOffice、WPS Office、Master PDF Editor、印象笔记、OneNote 等各种应用，配合图像处理软件可以轻松的处理文档，写稿、发稿什么的毫无问题。

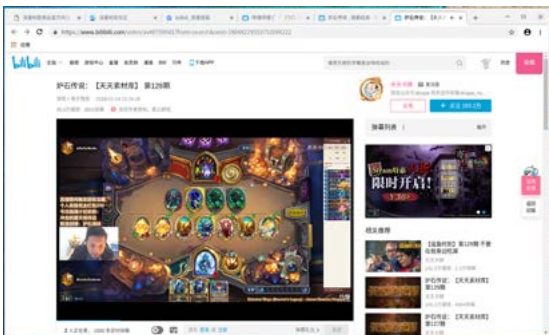


WPS Office 界面与我们现在使用的几乎一样

当然，如果是重度办公例如剪辑视频、3D 建模、CAD 画图官方也有应用可选，不过由于本人并不了解相关软件所以效率问题还有待商榷，但是大部分都有软件一一对应。

### 日常应用体验：影音娱乐

办公是没什么问题了，不过在闲暇之余放松一下能做到吗？首先是本地影音部分，系统内置了图片和影音播放器，看个电影是没问题的。



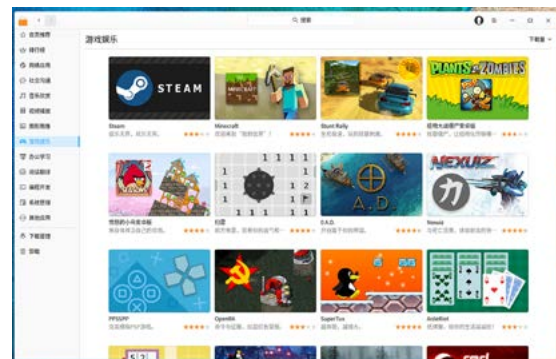
BILIBILI

在线影音方面，系统内置了谷歌浏览器（并支持诸如火狐、360 等浏览器的下载），在观看全国最大的“同性交友”网站时和我们在 Windows 环境下是一样的，弹幕也非常流畅，支持 HTML5 和 FLASH 双模式播放。

哦对了，迅雷下载和百度网盘也是有的，所以... 你懂得。

### 日常应用体验：游戏

最后我们来说一下游戏，这也是目前深度 Deepin 系统的短板之一。但是这不怪深度，毕竟那么多的游戏不可能要求厂商来做底层系统上的适配，移动端 Windows Phone 就是个例子。



游戏娱乐

首先 STEAM 软件本身是做了适配的，不过里面的游戏除了官方支持系统的以外就只能靠流式传输



本地运行 STEAM 版《以撒》



了，另外的游戏就基本上是一些棋牌类和小游戏了。

像是国内比较火的《英雄联盟》、《DNF》等腾讯系游戏和其它热门游戏并没有做适配，所以目前我们并不能玩到。不过也幸好深度是基于 Linux，已经有不少大神在网上发布了适配版本，相信如果今后大面积使用后，厂商也会相应的做适配。

### 总结：我们可能低估了国产操作系统的实力

俗话说，不去经历，你永远不知道外面的世界是什么样子的。笔者在之前只接触过 Windows 和 Mac 操作系统。当第一次要写这个国产操作系统文章时，我甚至都不确定它是否能够稳定运行。

而当我使用了几天深度 Deepin 系统后，我对国产操作系统的信心是逐渐增加的。虽然现在系统有着诸如游戏种类少、硬件兼容性稍差、偶尔卡顿的情况，但是近千款应用的支持，和华为进行合作进军服务器平台，入选中央政府采购网信息类产品名录，这些都在说明 Linux 操作系统正在快速进步。

我在这里也呼吁各大厂商——至少是国内的厂商，能够对好的中国团队研发的 Linux 操作系统进行适配，至少能够让这些系统在国内的大环境下能够进一步加强实用性，并良好的发展下去，这样才会有越来越多的用户来了解并帮助改善问题，打造一个良性发展生态圈。d



《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》已有定期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，我们开设了新栏目——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：qindi@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：

qindi@deepin.com

deepin

we do we change...