Linux初学者Patch使用指南 PDF转换可能丢失图片或格式，建议阅读原文
https://www.100test.com/kao_ti2020/252/2021_2022_Linux_E5_88_9D_E5_AD_c103_252960.htm 本文的目的是向Linux新手介绍一种无价的资源，Larry Wall的patch程序。patch是用来查找文件之间差异的GNU diff命令的一个接口；diff有很多选项，但是该命令最常用的用途是用来生成一个文件，该文件中列出了内容发生改变的行，显示两个原始文件、修改过的行以及由于内容没有变化而忽略掉的行。patch典型地用于把一个目录下的源代码文件更新到新的版本，从而就避免了下载整个新的源代码档案的必要。下载一个有效的 patch仅仅需要下载发生变化的那些代码行就可以了。 patch最初源自十年前，那时网络带宽的限制促进了patch的发展，然而和当时的很多Unix工具一样，直到现在，patch还在广泛应用。在Dr. Dobb之旅的2月份的程序员杂志中，Larry Wall对早期的patch做了一些很有趣的说明： DDJ:顺便问一下，patch和diff哪个出现的早？ LW:从很长一段时间来说，diff出现地比较早。我想diff大约比patch早10年出现，一回想起来，我就纳闷为什么没有人早些想到使用patch呢？ 但是我想我知道这中间的原因。这很大程度上是心理因素使然。当开发出diff时，程序员增加了一个e选项，我想就是这个选项的原因，该选项后来滋生为一个ed脚本，因此大家都会对自己说，"嗯，如果我想自动使用diff，那么我就使用这个选项。"因此从来都没有人编写一个计算机程序来获取其它格式的输出并使用这些结果。或者是那些设计diff的人员，或者是那些使用diff格式而受益的人员太沉迷其中了，因为你可以对那些已经修改过的内容使用diff

操作并让这些内容正常工作都是很容易的。 现在回想起来，这个问题是显而易见的。但是平心而论，与其说这是一个天才的灵感的闪现，还不如说这是自信心的体现。我开发出rn的第一个版本，然后继续为它编写补丁，这整个事情就是一团乱麻。你不可能强制用户使用补丁，因为他们可以手工完成这些工作。因此，他们就会省略一些自己认为不必要的工作，他们把新的修改加诸于原来的程序之上，因此而使得程序混乱。我编写补丁，这样就没有人找借口说这很难了。 我不清楚是否事实就是如此，但是多年以来，我一直对别人讲patch对于计算机文化的影响比rn和Perl的影响都要大。现在Internet的速度比原来有大幅度的提高，把整个发行版本分散到世界各地也变得更加简单，似乎只有在开发者之间才需要传送补丁。我已经很多年没有传送 Perl的patch工具包了。我认为虽然patch整体上的重要性在逐渐降低，但是仍然是开发者交流思想的一种方法。但是就那一段时间而言，patch 真正在相当大的程度上都影响了软件的开发方式。 Larry Wall针对patch对于计算机业界总体上重要性正在降低的评价可能是正确的，但是在自由软件世界中，patch仍然是一种必不可少的工具。无处不在的patch使得新手和非程序员能够简单地参与软件的alpha测试和 beta测试，这对于整个计算机业界是十分有益的。 在我留意到在Linux内核邮件列表中会周期性的出现这样一件事情时就产生了写这篇文章的念头。大约每三个月就会有人张贴要求把Linux内核源代码的发行版本独立出来的文章，据说这是因为有些人只对i386的代码和IDE的磁盘驱动感兴趣，他们并不想为每个内核发行版本都下载Alpha、Sparc等等的文件和众多的SCSI驱动程序。这篇文章后面紧

跟的是一些耐心的回复文章（有些文章则并没有耐心），大部分文章都是在讨论原来的使用有关patch来更新内核源代码。接着Linus Torvalds就会再次声明自己没有兴趣投身于这种把内核源程序切割成小块的繁杂的劳动，但是如果有人愿意，他们可以自由地开展这项独立的工程。到现在为止都没有志愿者出现。我并不想谴责那些内核黑客不能耐心等待，却把生活变得复杂；我猜想直接使用内核工作可能比检查整个内核的发行版本方案要更加有趣、更富有挑战性。下载11M的内核源程序包可是件非常耗费时间的事情（对于那些按照时间上网的人来说，这是很昂贵的），但是内核patch才只有几十K大小，很少会超过1M。我的硬盘上的2.1.99开发内核源程序经过patch的升级，已经升级到了2.1.119版本，我怀疑如果我紧随内核的发展而不断升级，那么也许我就要完整地下载每一个发行版本了。 使用patch patch附带有一个很好的帮助，其中罗列了很多选项，但是99%的时间只要两个选项就能满足我们的需要： patch -p1 patch -R -p1选项代表patchfile中文件名左边目录的层数，顶层目录在不同的机器上有所不同。要使用这个选项，就要把你的patch放在要被打补丁的目录下，然后在这个目录中运行path -p1 diff -u recursive new-file v2.1.118/linux/mm/swapfile.c linux/mm/swapfile. c v2.1.118/linux/mm/swapfile.c Wed Aug 26 11:37:45 1998 linux/mm/swapfile.c Wed Aug 26 16:01:57 1998 @@ -489,7 489,7 @@ int swap_header_version. int lock_map_size = PAGE_SIZE. int nr_good_pages = 0. - char tmp_lock_map = 0. unsigned long tmp_lock_map = 0. 应用来自本段中使用-p1开关拷贝的patch可以有效地减短patch定位的路径；patch会查找当前目录下一个

名为/mm的子目录，接着应该会在这儿发现swapfile.c文件，然后等待打补丁。在这个过程中，以破折号（“ - ”号，译者注）开始的行会被一个以加号（“ ”号，译者注）开始的行代替。一个典型的patch会包含对多个文件的更新，每个部分中都由对两个版本的文件运行diff -u命令的输出结果组成。patch在操作时把自己的输出结果显示在屏幕上，但是这种输出通常都滚屏太快，来不及观看。原来准备patch的文件名为*.orig，新的patch文件会覆盖这个初始文件名。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com