合理配置Fedora下的Firefox使其加速Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/644/2021_2022__E5_90_88_E7_90_86_E9_85_8D_E7_c103_644792.htm Firefox的加速方式有很多种，这里只列出常用的简单方法。在地址栏里输入：about：config，然后更改如下几个键值(如果找不到可以新建)： network.http.pipelining 双击赋值为 true，默认为 false。如果没有找到这个键值，可以右键新建一个 Boolean，把她赋值为 true 就可以了。激活这个键值之后，Pipelining 同时发出成倍数的连接请求，可以减少 Firefox 载入网页的时间。不过并不是所有网页所在的服务器都支持这种操作。 network.http.pipelining.maxrequests 双击并赋值为 8或者更多，默认键值为4。增加最大连接数，从而加快速度。 network.http.proxy.pipelining 双击并赋值为 true。理由同上。 nglayout.initialpaint.delay 右键新建 Integer 值，键名为 nglayout.initialpaint.delay，赋值 0。 这里实际上延迟了整个网页的显示速度，但是因为用户更喜欢在整个网页完全截入之前就开始阅读网页 ，所以在这里可以把值调为零，加速用户阅读网页的速度，有时候阅读速度和载入速度并不是成正比的。 以上四项的修改已经很明显的能改善Firefox的浏览速度了，如果还不满意可以再尝试继续修改以下键值： network.dns.disableIPv6 双击并赋值为 true。IPv6 把 IP 地址由 32 位增加到 128 位，从而能够支持更大的地址空间，当用户在终端向一个 IPv6-capable DNS 服务器发送连接请求时，也许服务器端会错误的返回给用户一个 IPv4 地址。而 Firefox 可以察觉这一切，不过在 Firefox 纠错的同时也必然会导致信号

的延迟，所以这里我们把她赋值为 true，禁用掉。 content.interrupt.parsing 右键新建 Boolean 值，键名为 content.interrupt.parsing，赋值 true。默认这个键值并不存在。我们激活这个键值之后，当目标网页载入时，Firefox 会根据一定频率打断解析的过程，不断的向用户反馈她所收集到的网页信息。 content.max.tokenizing.time 右键新建 Integer 值，键名为 content.max.tokenizing.time，赋值 2250000。 这个键值的作用其实就是指定一个循环事件的处理周期，这里的单位是微秒。理论上当我们将这个值取的越小，网页就会从视觉上载入的越流畅，因为 Firefox 会在很短的单位时间里反馈回解析到的网页信息。可是这样无疑延迟了网页整体载入的时间，所以在这里我们不妨将这个周期取的大一些，理论上可以加速网页的载入。 content.notify.interval 右键新建 Integer 值，键名为 content.notify.interval，赋值 750000。 第一次向远端主机发出连接请求到在终端收到这个预载入页面花费的时间，就是这里要定义的键值。理论上当这个时间设置的很低时，肯定会更快的拿到所谓的预载入页面，可这是一种杀鸡取卵的做法，这样无形中反而增加了整体页面的载入时间。按照官方的说法，低于 100,000 将会降低 Firefox 的性能，那好吧，那就把它彪到 750000 吧。 content.notify.ontimer 右键新建 Boolean 值，键名为 content.notify.ontimer，赋值 true。 为了使上面设置的 750000 微秒生效，还需要把这个键值激活。只有这两个键值配合，才会起作用。 content.notify.backoffcount 右键新建 Integer 值，键名为 content.notify.backoffcount，赋值 5。 这个键值控制 Firefox 的内置计数器在归零之前载入页面返回的次数。我们将目标网页分成好多个部分进行下载，每下

载完一个部分，计数器归零一次。-1 就是没有限制，值为 0 时这项功能被禁用。这里将它设置成 5， 当返回的次数达到五次而这部分网页还没有完全下载完时，那么剩下的没有下载完的网页内容将不会再按照预设置的周期，像之前的五次那样一点一点的搬运回来，而是会一次性的下载完。也就是说在这个部分的网页下载过程中，Firefox 一共反馈了 6 次信息，前 5 次的时间间隔是上面的键值中设置的周期 2250000 微秒，而第 6 次也就是最后一次则没有时间限制，什么时候把剩下的下完了，什么时候反馈回来。 当然，只有上面提到的 content.notify.ontimer 键值为 true 的时候，这里的设置才会生效。 content.switch.threshold 右键新建 Integer 值，键名为 content.switch.threshold ，赋值 750000，也就是四分之三秒。在前面提到了一个键值 content.interrupt.parsing，通过激活它实际上可以在载入页面的过程中跟 Firefox 产生互动。把 content.interrupt.parsing 激活后当页面载入时 Firefox 会有两种操作模式：高频和低频中断模式。使用高频模式时，网页回馈的频率也很高，网页载入过程也会更加的平滑。低频时网页回馈的频率相对比较低，可是这时反而加快了网页载入的时间。当移动鼠标或者触击键盘时，高频模式被激活。在经过某一段时间没有碰鼠标和键盘，程序没有接到鼠标和键盘发出的任何指令时，Firefox 就会自动进入低频模式工作，而这所谓的某一段时间，就是这里要指定的值。 编辑特别推荐: Linux系统通过手机GPRS上网设置简介 提高Apache服务器性能的四个建议 Linux认证能帮助你找到一份好工作吗 linux面试题参考答案 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