

内核中的likely和unlikely宏定义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\\_ti2020/645/2021\\_2022\\_\\_E5\\_86\\_85\\_E6\\_A0\\_B8\\_E4\\_B8\\_AD\\_E7\\_c103\\_645065.htm](https://www.100test.com/kao_ti2020/645/2021_2022__E5_86_85_E6_A0_B8_E4_B8_AD_E7_c103_645065.htm) 在内核代码中经常

会看到unlikely和likely的踪影。他们实际上是定义在linux/compiler.h 中的两个宏。 #define likely(x)

\_\_builtin\_expect(!!(x), 1) #define unlikely(x)

\_\_builtin\_expect(!!(x), 0) 这里的\_\_builtin\_expect()函数是gcc的内置函数。至于为什么要在内核代码中使用这两个宏，主要的目的是为了进行代码的优化，提高系统执行速度。比如：if (likely(alt.b)) { fun2(). } 这里就是程序员可以确定 alt.b 的可能性很小所以这里在编译时，将fun2()的二进制代码尽量不要和前边的编译在一块。我们不用对likely和unlikely感到迷惑，需要知道的就是 if(likely(alt.b))在功能上是等价的，同样 if(unlikely(alt.b)) 的功能也是一样的。不一样的只是他们声称的二进制代码有所不同，这一点我们也可以从他们的汇编代码中看到。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](http://www.100test.com)