ltp 之 kcmp 问题分析

1 问题概述

在测试 ltp 的 kcmp 程序时，出现错误：
kcmp not supported

2 问题分析

查看 kcmp 系统调用的返回值为-1，出错了。用 perror 打印下对应 errno 的描述信息，说是 Function not implemented。事情到这儿看起来可以结束了，errno 也的确是 ENOSYS，对应的描述信息也说的很清楚，内核没支持没实现。

可是内核不支持不实现的话，那也应该在代码里看到，我们差这块代码，仅靠 errno 我依然表示怀疑。

于是 dive into 到内核代码，第一个问题要问自己的就是，内核是怎么实现一个系统调用的呢？几番 Google 下来，其中之一的点是，说是有个 sys_call_table 的东西，实现好的系统调用比如 write，read 这些，会把它们的函数指针放到这个表中。

可以看到 506 号 sys call 被写入串 sys_kcmp，按理说，sys_kcmp 应该是实现了的。

```
sys_call_table:
...
.quad sys_kcmp
```

那为什么上层系统调用 kcmp 就是返回 ENOSYS 呢？

看起来还要看点知识。还是那个问题，内核究竟是怎么给用户空间导出/实现一个系统调用的呢？参考文献 1，2 给出了点信息，讲述了怎么添加一个系统调用。

从参考文献 1,2 中得到点的有用信息是，sys_ni_syscall 会返回 ENOSYS，未定义的系统调用都会重定向这个 sys call 来。它定义在内核文件 kernel/sys_ni.c 里。看其来我们调用 sys_kcmp 被重定向到了这个函数。可是，哪儿的代码说明了这一点？现象如此还应该对应代码啊。

还是 sys_ni.c 文件有一个宏 COND_SYSCALL，一步步向下，展开它：

```
#define cond_syscall(x)  asm(".weak\t" #x "\n" #x " =
↪  sys_ni_syscall")
```

COND_SYSCALL(kcmp) 被定义，向下会展开成上述宏。上述宏会把符号定义成 sys_ni_syscall，也就是默认的返回 ENOSYS 的系统调用。

其中的 weak 属性就有讲究了，参考文献 3 介绍了下。概要地说就是，x 会被定义成一个弱符号，如果后续出现其强定义的符号，sys_ni_syscall 会被替代。

那么，kcmp 有强定义吗？

内核在导出系统调用给用户使用时，使用 SYSCALL_DEFINEx，其中 x 是系统调用的参数个数。于是找找 SYSCALL_DEFINE5 有没有定义 kcmp 啊？在 kernel/kcmp.c 里：

```
SYSCALL_DEFINE5(kcmp, pid_t, pid1, pid_t, pid2, int, type,
                unsigned long, idx1, unsigned long, idx2)
```

的确有 kcmp 的定义。这下看清楚了，kcmp 是实现了的！并不是上层应用程序看到的那样内核没有实现不支持。

至于上层应用程序不能使用那是编译的问题而不是实现否的问题，所以“not supported”在一定程度上是有歧义的。其实 kcmp.c 编译否在 Makefile 里要依赖一个宏配置，如 kernel/Makefile 的如下行：

```
obj-$(CONFIG_CHECKPOINT_RESTORE) += kcmp.o
```

就是要配置 CHECKPOINT_RESTORE 了。

其实 man kcmp 的最后 NOTES 也有这么一句话：

> This system call is available only if the kernel was configured with CONFIG_CHECKPOINT_RESTORE.

这么看起来 man 手册还是蛮可爱的，以后要常读呀！

3 实验验证

验证就简单了，打开内核选项 CHECKPOINT_RESTORE，重新编译上新内核，kcmp 所有测试 passed ;-)。

4 解决方案

This system call is available only if the kernel was configured with CONFIG_CHECKPOINT_RESTORE.

5 参考文献

1. https://0xax.gitbooks.io/linux-insides/content/SysCall/linux-syscall-2.html
2. https://www.kernel.org/doc/html/v4.10/process/adding-syscalls.html
3. https://gcc.gnu.org/onlinedocs/gcc-3.2/gcc/Function-Attributes.html#:~:text=The%20weak%20attribute%20causes%20the,targets-%2C%20and%20also%20for%20a.