# 测试报告

Issue标题：[openEuler 22.03] 增加 no\_hash\_pointers 启动参数让 printk %p 重新打印内核实际地址

Issue链接：<https://gitee.com/openeuler/kernel/issues/I4BUC2>

参与人：何金津

Gitee：[jinjin@hejinjin1](https://gitee.com/hejinjin1)

kernel version：4.19.90

提交分支：openEuler-1.0-LTS（OLK-5.10已适配过该特性）

## 补丁集

├── 0000-Add-the-no\_hash\_pointers-startup-parameter.patch

├── 0001-lib-test\_printf-Add-empty-module\_exit-function.patch

├── 0002-kselftest-Add-test-module-framework-header.patch

├── 0003-lib-Use-new-kselftest-header.patch

├── 0004-lib-use-KSTM\_MODULE\_GLOBALS-macro-in-kselftest-drivers.patch

├── 0005-kselftest-add-support-for-skipped-tests.patch

└── 0006-lib-vsprintf-no\_hash\_pointers-prints-all-addresses-as-unhashed.patch

## 补丁描述

补丁集的主要功能为向vsprintf.c中增加了新的启动参数no\_hash\_pointers，并且在通过%p打印地址时进行判断，如果no\_hash\_pointers == 0 则进入pointer\_string(buf, end, ptr, spec)函数返回真实的物理地址（这也是通过%px打印调用的函数），如果no\_hash\_pointers == 1则进入ptr\_to\_id(buf, end, ptr, spec)，这个函数功能是对物理地址进行哈希散列，然后返回散列后的地址。

原补丁作者在修改主要功能后，也对两个相关的测试文件同时进行了修改test\_printf.c与test\_bitmap.c（在0004、0005补丁中），在4.19版本的内核中缺少一些前置修改，为了使patch能够通过编译查找添加了0001、0002、0003这三个补丁，增加了两个测试文件的一些前置修改。

## 测试步骤

##### 测试样例

├── Makefile

├── no\_hash\_pointers\_test.c

└── test.sh

测试样例包含了对此时no\_hash\_pointers值的打印，并分别用%pt,%p,%pk,%px打印栈数组与堆数组的地址。如果正确则no\_hash\_pointers == 0时，%pt,%p,%pk会打印出散列值，%px会打印出物理地址，no\_hash\_pointers == 1时，%pt会打印出散列值，%p,%pk,%px会打印出物理地址

可以直接sh test.sh完成编译，内核插入，内核卸载，信息输出这几步，然后会在命名行输出结果。也可以手动输入test.sh的内容：

make

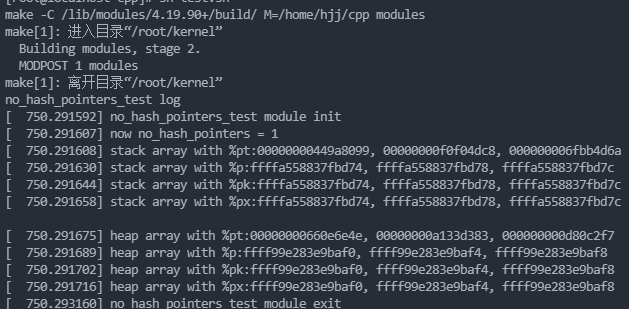
insmod no\_hash\_pointers\_test.ko

rmmod no\_hash\_pointers\_test.ko

echo "no\_hash\_pointers\_test log"

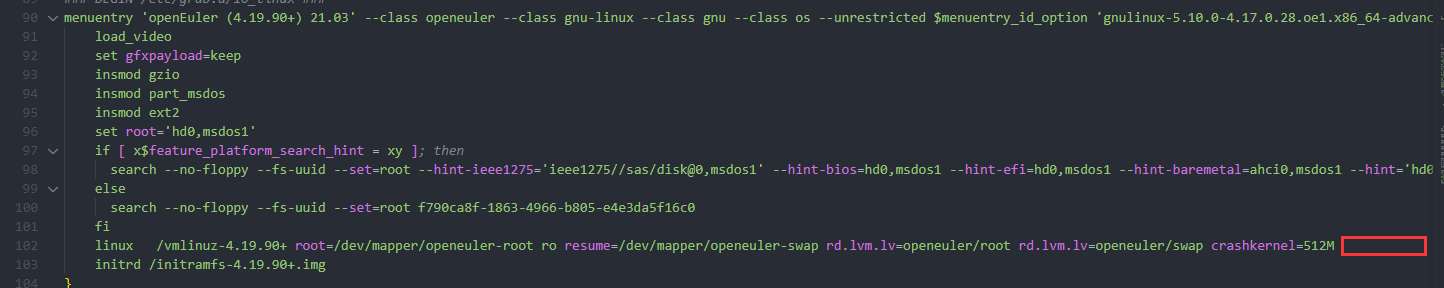
dmesg | tail -n 12

期待结果类似于下图（图为no\_hash\_pointers == 1得到的输出）:



-------------------------------------------------------------------------------

启动参数可以在/boot/grub2/grub.cfg文件中红框内指定值之后重启



如果不指定则no\_hash\_pointers为0，指定为1。