in arch/arm/mm/Kconfig

```
config KUSER_HELPERS
	bool "Enable kuser helpers in vector page" if !NEED_KUSER_HELPERS
	depends on MMU
	default y
	help
	  Warning: disabling this option may break user programs.

	  Provide kuser helpers in the vector page.  The kernel provides
	  helper code to userspace in read only form at a fixed location
	  in the high vector page to allow userspace to be independent of
	  the CPU type fitted to the system.  This permits binaries to be
	  run on ARMv4 through to ARMv7 without modification.

	  See Documentation/arm/kernel_user_helpers.txt for details.

	  However, the fixed address nature of these helpers can be used
	  by ROP (return orientated programming) authors when creating
	  exploits.

	  If all of the binaries and libraries which run on your platform
	  are built specifically for your platform, and make no use of
	  these helpers, then you can turn this option off to hinder
```

such exploits. However, in that case, if a binary or library relying on those helpers is run, it will receive a SIGILL signal, which will terminate the program.

Say N here only if you are absolutely certain that you do not need these helpers; otherwise, the safe option is to say Y.

in arch/arm/kernel/entry-armv.S

```
#ifdef CONFIG_KUSER_HELPERS
        .align  5
        .globl  __kuser_helper_start
__kuser_helper_start:

/*
 * Due to the length of some sequences, __kuser_cmpxchg64 spans 2 regular
 * kuser "slots", therefore 0xffff0f80 is not used as a valid entry point.
 */

__kuser_cmpxchg64:                              @ 0xffff0f60

#if defined(CONFIG_NEEDS_SYSCALL_FOR_CMPXCHG)

        /*
         * Poor you.  No fast solution possible...
         * The kernel itself must perform the operation.
         * A special ghost syscall is used for that (see traps.c).
         */
        stmfd   sp!, {r7, lr}
        ldr     r7, 1f                  @ it's 20 bits
        swi     __ARM_NR_cmpxchg64
        ldmfd   sp!, {r7, pc}
1:      .word   __ARM_NR_cmpxchg64

#elif defined(CONFIG_CPU_32v6K)

        stmfd   sp!, {r4, r5, r6, r7}
        ldrd    r4, r5, [r0]                    @ load old val
        ldrd    r6, r7, [r1]                    @ load new val
        smp_dmb arm
1:      ldrexd  r0, r1, [r2]                    @ load current val
        eors    r3, r0, r4                      @ compare with oldval (1)
        eoreqs  r3, r1, r5                      @ compare with oldval (2)
        strexdeq r3, r6, r7, [r2]               @ store newval if eq
        teqeq   r3, #1                          @ success?
        beq     1b                              @ if no then retry
        smp_dmb arm
        rsbs    r0, r3, #0                      @ set returned val and C flag
        ldmfd   sp!, {r4, r5, r6, r7}
        usr_ret lr

#elif !defined(CONFIG_SMP)

#ifdef CONFIG_MMU

        /*
         * The only thing that can break atomicity in this cmpxchg64
         * implementation is either an IRQ or a data abort exception
         * causing another process/thread to be scheduled in the middle of
         * the critical sequence.  The same strategy as for cmpxchg is used.
         */
        stmfd   sp!, {r4, r5, r6, lr}
        ldmia   r0, {r4, r5}                    @ load old val
        ldmia   r1, {r6, lr}                    @ load new val
1:      ldmia   r2, {r0, r1}                    @ load current val
        eors    r3, r0, r4                      @ compare with oldval (1)
        eoreqs  r3, r1, r5                      @ compare with oldval (2)
2:      stmeqia r2, {r6, lr}                    @ store newval if eq
        rsbs    r0, r3, #0                      @ set return val and C flag
        ldmfd   sp!, {r4, r5, r6, pc}

        .text
kuser_cmpxchg64_fixup:
        @ Called from kuser_cmpxchg_fixup.
        @ r4 = address of interrupted insn (must be preserved).
        @ sp = saved regs. r7 and r8 are clobbered.
        @ 1b = first critical insn, 2b = last critical insn.
        @ If r4 >= 1b and r4 <= 2b then saved pc_usr is set to 1b.
        mov     r7, #0xffff0fff
        sub     r7, r7, #(0xffff0fff - (0xffff0f60 + (1b - __kuser_cmpxchg64)))
        subs    r8, r4, r7
        rsbcss  r8, r8, #(2b - 1b)
        strcs   r7, [sp, #S_PC]
#if __LINUX_ARM_ARCH__ < 6
        bcc     kuser_cmpxchg32_fixup
#endif
        ret     lr
        .previous

#else
#warning "NPTL on non MMU needs fixing"
        mov     r0, #-1
        adds    r0, r0, #0
        usr_ret lr
#endif

#else
#error "incoherent kernel configuration"
#endif

        kuser_pad __kuser_cmpxchg64, 64

__kuser_memory_barrier:                         @ 0xffff0fa0
        smp_dmb arm
        usr_ret lr

        kuser_pad __kuser_memory_barrier, 32

__kuser_cmpxchg:                                @ 0xffff0fc0

#if defined(CONFIG_NEEDS_SYSCALL_FOR_CMPXCHG)

        /*
         * Poor you.  No fast solution possible...
         * The kernel itself must perform the operation.
         * A special ghost syscall is used for that (see traps.c).
         */
        stmfd   sp!, {r7, lr}
        ldr     r7, 1f                  @ it's 20 bits
        swi     __ARM_NR_cmpxchg
        ldmfd   sp!, {r7, pc}
1:      .word   __ARM_NR_cmpxchg

#elif __LINUX_ARM_ARCH__ < 6

#ifdef CONFIG_MMU

        /*
         * The only thing that can break atomicity in this cmpxchg
         * implementation is either an IRQ or a data abort exception
         * causing another process/thread to be scheduled in the middle
         * of the critical sequence.  To prevent this, code is added to
         * the IRQ and data abort exception handlers to set the pc back
         * to the beginning of the critical section if it is found to be
         * within that critical section (see kuser_cmpxchg_fixup).
         */
1:      ldr     r3, [r2]                        @ load current val
        subs    r3, r3, r0                      @ compare with oldval
2:      streq   r1, [r2]                        @ store newval if eq
        rsbs    r0, r3, #0                      @ set return val and C flag
        usr_ret lr

        .text
kuser_cmpxchg32_fixup:
        @ Called from kuser_cmpxchg_check macro.
        @ r4 = address of interrupted insn (must be preserved).
        @ sp = saved regs. r7 and r8 are clobbered.
        @ 1b = first critical insn, 2b = last critical insn.
        @ If r4 >= 1b and r4 <= 2b then saved pc_usr is set to 1b.
        mov     r7, #0xffff0fff
        sub     r7, r7, #(0xffff0fff - (0xffff0fc0 + (1b - __kuser_cmpxchg)))
        subs    r8, r4, r7
        rsbcss  r8, r8, #(2b - 1b)
        strcs   r7, [sp, #S_PC]
        ret     lr
        .previous

#else
#warning "NPTL on non MMU needs fixing"
        mov     r0, #-1
        adds    r0, r0, #0
        usr_ret lr
#endif

#else

        smp_dmb arm
1:      ldrex   r3, [r2]
        subs    r3, r3, r0
        strexeq r3, r1, [r2]
        teqeq   r3, #1
        beq     1b
        rsbs    r0, r3, #0
        /* beware -- each __kuser slot must be 8 instructions max */
        ALT_SMP(b       __kuser_memory_barrier)
        ALT_UP(usr_ret  lr)

#endif

        kuser_pad __kuser_cmpxchg, 32

__kuser_get_tls:                                @ 0xffff0fe0
        ldr     r0, [pc, #(16 - 8)]     @ read TLS, set in kuser_get_tls_init
        usr_ret lr
        mrc     p15, 0, r0, c13, c0, 3  @ 0xffff0fe8 hardware TLS code
        kuser_pad __kuser_get_tls, 16
        .rep    3
        .word   0                       @ 0xffff0ff0 software TLS value, then
        .endr                           @ pad up to __kuser_helper_version

__kuser_helper_version:                         @ 0xffff0ffc
        .word   ((__kuser_helper_end - __kuser_helper_start) >> 5)

        .globl  __kuser_helper_end
__kuser_helper_end:

#endif
```

由于汇编源码有很多条件编译，有点乱，还不如直接看反汇编码。(在Granite2 / Gemstone2 LSP的配置下)

```
CONFIG_NEEDS_SYSCALL_FOR_CMPXCHG is not set

CONFIG_CPU_32v6K=y
```

```
================================================================

c05acb80 <__kuser_helper_start>:

c05acb80:       e92d00f0        push    {r4, r5, r6, r7}

c05acb84:       e1c040d0        ldrd  r4, [r0]

c05acb88:       e1c160d0        ldrd  r6, [r1]
```

```
c05acb8c:    f57ff05b   dmb ish
c05acb90:    e1b20f9f   ldrexd     r0, [r2]
c05acb94:    e0303004        eors r3, r0, r4
c05acb98:    00313005        eorseq     r3, r1, r5
c05acb9c:    01a23f96        strexdeq  r3, r6, [r2]
c05acba0:    03330001        teqeq      r3, #1
c05acba4:    0afffff9   beq  c05acb90 <__kuser_helper_start+0x10>
c05acba8:    f57ff05b   dmb ish
c05acbac:    e2730000        rsbs r0, r3, #0
c05acbb0:    e8bd00f0        pop  {r4, r5, r6, r7}
c05acbb4:    e12fff1e   bx    lr
c05acbb8:    e7fddef1   .word      0xe7fddef1
c05acbbc:    e7fddef1   .word      0xe7fddef1


c05acbc0 <__kuser_memory_barrier>:                    ① 内存屏障函数
c05acbc0:    f57ff05b   dmb ish
c05acbc4:    e12fff1e   bx    lr
c05acbc8:    e7fddef1   .word      0xe7fddef1
c05acbcc:    e7fddef1   .word      0xe7fddef1
c05acbd0:    e7fddef1   .word      0xe7fddef1
c05acbd4:    e7fddef1   .word      0xe7fddef1
c05acbd8:    e7fddef1   .word      0xe7fddef1
c05acbdc:    e7fddef1   .word      0xe7fddef1


c05acbe0 <__kuser_cmpxchg>:                                 ② 原子的比较交换函数
```

```
c05acbe0:    f57ff05b   dmb ish
c05acbe4:    e1923f9f   ldrexr3, [r2]
c05acbe8:    e0533000       subs      r3, r3, r0
c05acbec:    01823f91       strexeq   r3, r1, [r2]
c05acbf0:    03330001       teqeq     r3, #1
c05acbf4:    0afffffa   beq  c05acbe4 <__kuser_cmpxchg+0x4>
c05acbf8:    e2730000       rsbs r0, r3, #0
c05acbfc:    eaffffef   b      c05acbc0 <__kuser_memory_barrier>


c05acc00 <__kuser_get_tls>:                                  ③get_tls() get Thread Local Storage
c05acc00:    e59f0008       ldr    r0, [pc, #8]      ; c05acc10 <__kuser_get_tls+0x10>
c05acc04:    e12fff1e   bx    lr
c05acc08:    ee1d0f70       mrc  15, 0, r0, cr13, cr0, {3}
c05acc0c:    e7fddef1  .word      0xe7fddef1
      ...


c05acc1c <__kuser_helper_version>:
c05acc1c:    00000005       .word      0x00000005
=================================================================
```

从code上看当CONFIG_KUSER_HELPERS enable之时，定义了如下汇编函数：

① __kuser_memory_barrier　使得可能乱序（out of order）执行的code在内存屏障这点上得到同步

② __kuser_cmpxchg　　提供比较与交换某个variable是原子(atomic)不可分的（在多ARM core的情况下也如此），在Intel CPU上可以通过加指令前缀lock实现

③ __kuser_get_tls　　取得每个thread的local storage

这些函数而且都是固定地址的：

__kuser_memory_barrier: @ 0xffff0fa0

__kuser_cmpxchg: @ 0xffff0fc0

__kuser_get_tls: @ 0xffff0fe0

原因见《about exception entries of ARM》笔记。

kuser_help code将被复制到high vector page所在的0xffff,0000 virtual page的尾部。

in arch/arm/kernel/traps.c

```
#ifdef CONFIG_KUSER_HELPERS

static void __init kuser_init(void *vectors)

{

        extern char __kuser_helper_start[], __kuser_helper_end[];

        int kuser_sz = __kuser_helper_end - __kuser_helper_start;


        memcpy(vectors + 0x1000 - kuser_sz, __kuser_helper_start, kuser_sz);


        /*

         * vectors + 0xfe0 = __kuser_get_tls

         * vectors + 0xfe8 = hardware TLS instruction at 0xffff0fe8

         */

        if (tls_emu || has_tls_reg)
        memcpy(vectors + 0xfe0, vectors + 0xfe8, 4);

}

#else

static inline void __init kuser_init(void *vectors)

{

}

#endif
```

Documentation/arm/kernel_user_helpers.txt有相关kuser helper为什么要提供这些函数的description.

这些有点怪怪的函数是专门提供给user mode application使用的。它们提供了在user mode无法做到，但又希望尽量lightweight的实现（如果用类似system call方式实现，

未免太“重”了）。这也是为什么这些函数和变量（比如__kuser_helper_version）都要是fixed address。

在kerkel_user_helpers.txt有如下描述：

These are segment of kernel provided user code reachable from user space

at a fixed address in kernel memory. This is used to provide user space

with some operations which require kernel help because of unimplemented

native feature and/or instructions in many ARM CPUs. The idea is for this

code to be executed directly in user mode for best efficiency but which is

too intimate with the kernel counter part to be left to user libraries.

在user mode利用kernel提供的kuser helper来实现普通的user library(应该是NPTL library，一个pthread implementation in Linux).

```
walterzh$ cat verify_kuser_helper.c
#include <stdio.h>

typedef void * (__kuser_get_tls_t)(void);
#define __kuser_get_tls (*(__kuser_get_tls_t *)0xffff0fe0)

void foo()
{
        void *tls = __kuser_get_tls();
        printf("TLS = %p\n", tls);
}

typedef int (__kuser_cmpxchg_t)(int oldval, int newval, volatile int *ptr);
#define __kuser_cmpxchg (*(__kuser_cmpxchg_t *)0xffff0fc0)

int atomic_add(volatile int *ptr, int val)
{
int old, new;

        do {
                old = *ptr;
                new = old + val;
        } while(__kuser_cmpxchg(old, new, ptr));

        return new;
}

typedef void (__kuser_dmb_t)(void);
#define __kuser_dmb (*(__kuser_dmb_t *)0xffff0fa0)

void memory_barry()
{
        __kuser_dmb();
}

int main(int argc, char *argv[])
{
        int count = 0;

        foo();

        atomic_add(&count, 1);

        printf("count = %d\n", count);

        memory_barry();

        return 0;
}
```

exception vector table所在的0xffff,0000 page要能被user mode application read / execute，同时如

此总要的kernel data又不能被usermode application “write”,否则一个application就可以轻易crash Linux。

```
root@granite2:~# ./verify_kuser_helper

TLS = 0xb6f4a4c0

count = 1
```

==

```
walterzh$ cat dump_vector_page.c

#include <stdio.h>


int main(int argc, char *argv[])

{

        unsigned int *p = (unsigned int *)0xffff0000;


        int i = 0;


        for(; i < 0x2000 / 4; i++)        ①

        {

                if(i % 4 == 0)

                {

                        printf("\n");

                        printf("%p: ", p + i);

                }
                printf("%08x ", *(p + i));
        }
        return 0;
}
```

①

read 0xffff0000, and 0xffff1000 2 pages.

只应该读取0xffff0000，而不能读取0xffff1000 page (vector stub code)

因为为了使得user mode application能读取/执行 kuser helper data和code，kernel需要把该page的权限开放出来（当然写是没有权限的），但0xffff1000 page则完全没有这个需要，所以application不能access。

```
root@granite2:~# ./cum   dump_vector_page

0xffff0000: ea0003ff ea000465 e59ffff0 ea000443    ②
0xffff0010: ea000422 ea000481 ea000400 ea000487
0xffff0020: e7fddef1 e7fddef1 e7fddef1 e7fddef1    ③
0xffff0030: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0040: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0050: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0060: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0070: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0080: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0090: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff00a0: e7fddef1 e7fddef1 e7fddef1 e7fddef1
```

```
0xffff00b0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff00c0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff00d0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff00e0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff00f0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0100: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0110: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0120: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0130: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0140: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0150: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0160: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0170: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0180: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0190: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff01a0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff01b0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff01c0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff01d0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff01e0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff01f0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0200: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0210: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0220: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0230: e7fddef1 e7fddef1 e7fddef1 e7fddef1
```

```
0xffff0240: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0250: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0260: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0270: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0280: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0290: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff02a0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff02b0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff02c0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff02d0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff02e0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff02f0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0300: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0310: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0320: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0330: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0340: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0350: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0360: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0370: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0380: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0390: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff03a0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff03b0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff03c0: e7fddef1 e7fddef1 e7fddef1 e7fddef1
```

```
0xffff03d0: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff03e0: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff03f0: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0400: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0410: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0420: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0f60: e92d00f0 e1c040d0 e1c160d0 f57ff05b
0xffff0f70: e1b20f9f e0303004 00313005 01a23f96
0xffff0f80: 03330001 0afffff9 f57ff05b e2730000
0xffff0f90: e8bd00f0 e12fff1e e7fddef1 e7fddef1
0xffff0fa0: f57ff05b e12fff1e e7fddef1 e7fddef1
0xffff0fb0: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0fc0: f57ff05b e1923f9f e0533000 01823f91
0xffff0fd0: 03330001 0afffffa e2730000 eaffffef
0xffff0fe0: ee1d0f70 e12fff1e ee1d0f70 e7fddef1
0xffff0ff0: 00000000 00000000 00000000 00000005
0xffff0430: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0440: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0450: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0460: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0470: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0480: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff0490: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff04a0: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff04b0: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff04c0: e7fddef1 e7fddef1 e7fddef1 e7fddef1
0xffff04d0: e7fddef1 e7fddef1 e7fddef1 e7fddef1
```

```
0xffff04e0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff04f0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0500: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0510: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0520: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0530: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0540: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0550: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0560: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0570: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0580: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0590: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff05a0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff05b0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff05c0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff05d0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff05e0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff05f0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0600: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0610: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0620: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0630: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0640: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0650: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0660: e7fddef1 e7fddef1 e7fddef1 e7fddef1
```

```
0xffff0670: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0680: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0690: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff06a0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff06b0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff06c0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff06d0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff06e0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff06f0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0700: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0710: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0720: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0730: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0740: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0750: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0760: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0770: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0780: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0790: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff07a0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff07b0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff07c0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff07d0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff07e0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff07f0: e7fddef1 e7fddef1 e7fddef1 e7fddef1
```

```
0xffff0800: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0810: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0820: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0830: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0840: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0850: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0860: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0870: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0880: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0890: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff08a0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff08b0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff08c0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff08d0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff08e0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff08f0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0900: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0910: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0920: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0930: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0940: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0950: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0960: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0970: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0980: e7fddef1 e7fddef1 e7fddef1 e7fddef1
```

```
0xffff0990: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff09a0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff09b0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff09c0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff09d0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff09e0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff09f0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0a00: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0a10: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0a20: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0a30: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0a40: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0a50: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0a60: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0a70: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0a80: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0a90: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0aa0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0ab0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0ac0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0ad0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0ae0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0af0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0b00: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0b10: e7fddef1 e7fddef1 e7fddef1 e7fddef1
```

```
0xffff0b20: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0b30: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0b40: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0b50: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0b60: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0b70: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0b80: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0b90: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0ba0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0bb0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0bc0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0bd0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0be0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0bf0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0c00: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0c10: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0c20: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0c30: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0c40: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0c50: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0c60: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0c70: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0c80: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0c90: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0ca0: e7fddef1 e7fddef1 e7fddef1 e7fddef1
```

```
0xffff0cb0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0cc0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0cd0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0ce0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0cf0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0d00: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0d10: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0d20: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0d30: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0d40: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0d50: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0d60: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0d70: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0d80: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0d90: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0da0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0db0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0dc0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0dd0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0de0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0df0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0e00: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0e10: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0e20: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0e30: e7fddef1 e7fddef1 e7fddef1 e7fddef1
```

```
0xffff0e40: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0e50: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0e60: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0e70: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0e80: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0e90: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0ea0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0eb0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0ec0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0ed0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0ee0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0ef0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0f00: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0f10: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0f20: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0f30: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0f40: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0f50: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0f60: e92d00f0 e1c040d0 e1c160d0 f57ff05b     ④

0xffff0f70: e1b20f9f e0303004 00313005 01a23f96

0xffff0f80: 03330001 0afffff9 f57ff05b e2730000

0xffff0f90: e8bd00f0 e12fff1e e7fddef1 e7fddef1

0xffff0fa0: f57ff05b e12fff1e e7fddef1 e7fddef1

0xffff0fb0: e7fddef1 e7fddef1 e7fddef1 e7fddef1

0xffff0fc0: f57ff05b e1923f9f e0533000 01823f91
```

```
0xffff0fd0: 03330001 0afffffa e2730000 eaffffef

0xffff0fe0: ee1d0f70 e12fff1e ee1d0f70 e7fddef1

0xffff0ff0: 00000000 00000000 00000000 00000005

Segmentation fault                    ⑤
```

②

vector entry

③

invalid instruction

④

kuser helper code

⑤

当要access 0xffff1000 page时，因为没有权限read,所以...