
Subject: [PATCH 3/5] Simplify kallsyms_lookup()

Posted by [Alexey Dobriyan](#) on Mon, 02 Apr 2007 14:57:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

Several kallsyms_lookup() pass dummy arguments but only need, say, module's name. Make kallsyms_lookup() accept NULLs where possible.

Also, makes picture clearer about what interfaces are needed for all symbol resolving business.

Signed-off-by: Alexey Dobriyan <adobriyan@sw.ru>

```
arch/parisc/kernel/unwind.c | 5 +----
arch/powerpc/xmon/xmon.c    | 3 +--
arch/sh64/kernel/unwind.c   | 6 +++----
fs/proc/base.c              | 5 ++---
kernel/kallsyms.c           | 6 ++++--
kernel/kprobes.c            | 4 ++--
kernel/lockdep.c            | 5 +----
kernel/module.c             | 6 ++++--
kernel/time/timer_list.c    | 4 +---
kernel/time/timer_stats.c   | 4 +---
10 files changed, 20 insertions(+), 28 deletions(-)
```

--- a/arch/parisc/kernel/unwind.c

+++ b/arch/parisc/kernel/unwind.c

@@ -216,11 +216,8 @@ #ifdef CONFIG_KALLSYMS

/* Handle some frequent special cases.... */

```
{
    char symname[KSYM_NAME_LEN+1];
```

- char *modname;

- unsigned long symsize, offset;

- kallsyms_lookup(info->ip, &symsize, &offset,

- &modname, symname);

+ kallsyms_lookup(info->ip, NULL, NULL, NULL, symname);

dbg("info->ip = 0x%lx, name = %s\n", info->ip, symname);

--- a/arch/powerpc/xmon/xmon.c

+++ b/arch/powerpc/xmon/xmon.c

@@ -1218,7 +1218,6 @@ static void get_function_bounds(unsigned

```
{
```

unsigned long size, offset;

const char *name;

- char *modname;

```

*startp = *endp = 0;
if (pc == 0)
@@ -1226,7 +1225,7 @@ static void get_function_bounds(unsigned
if (setjmp(bus_error_jmp) == 0) {
    catch_memory_errors = 1;
    sync();
- name = kallsyms_lookup(pc, &size, &offset, &modname, tmpstr);
+ name = kallsyms_lookup(pc, &size, &offset, NULL, tmpstr);
    if (name != NULL) {
        *startp = pc - offset;
        *endp = pc - offset + size;
--- a/arch/sh64/kernel/unwind.c
+++ b/arch/sh64/kernel/unwind.c
@@ -46,15 +46,15 @@ static int lookup_prev_stack_frame(unsig
    struct pt_regs *regs)
{
    const char *sym;
- char *modname, namebuf[128];
- unsigned long offset, size;
+ char namebuf[128];
+ unsigned long offset;
    unsigned long prologue = 0;
    unsigned long fp_displacement = 0;
    unsigned long fp_prev = 0;
    unsigned long offset_r14 = 0, offset_r18 = 0;
    int i, found_prologue_end = 0;

- sym = kallsyms_lookup(pc, &size, &offset, &modname, namebuf);
+ sym = kallsyms_lookup(pc, NULL, &offset, NULL, namebuf);
    if (!sym)
        return -EINVAL;

--- a/fs/proc/base.c
+++ b/fs/proc/base.c
@@ -275,14 +275,13 @@ @@ #ifdef CONFIG_KALLSYMS
    */
    static int proc_pid_wchan(struct task_struct *task, char *buffer)
    {
- char *modname;
- const char *sym_name;
- unsigned long wchan, size, offset;
+ unsigned long wchan;
+ char namebuf[KSYM_NAME_LEN+1];

    wchan = get_wchan(task);

- sym_name = kallsyms_lookup(wchan, &size, &offset, &modname, namebuf);
+ sym_name = kallsyms_lookup(wchan, NULL, NULL, NULL, namebuf);

```

```

if (sym_name)
    return sprintf(buffer, "%s", sym_name);
return sprintf(buffer, "%lu", wchan);
--- a/kernel/kallsyms.c
+++ b/kernel/kallsyms.c
@@ -214,8 +214,10 @@ static unsigned long get_symbol_pos(unsigned long
    symbol_end = (unsigned long)_etext;
}

- *symbolsize = symbol_end - symbol_start;
- *offset = addr - symbol_start;
+ if (symbolsize)
+ *symbolsize = symbol_end - symbol_start;
+ if (offset)
+ *offset = addr - symbol_start;

return low;
}
--- a/kernel/kprobes.c
+++ b/kernel/kprobes.c
@@ -867,13 +867,13 @@ static int __kprobes show_kprobe_addr(struct
    struct kprobe *p, *kp;
    const char *sym = NULL;
    unsigned int i = *(loff_t *) v;
- unsigned long size, offset = 0;
+ unsigned long offset = 0;
    char *modname, namebuf[128];

    head = &kprobe_table[i];
    preempt_disable();
    hlist_for_each_entry_rcu(p, node, head, hlist) {
- sym = kallsyms_lookup((unsigned long)p->addr, &size,
+ sym = kallsyms_lookup((unsigned long)p->addr, NULL,
        &offset, &modname, namebuf);
        if (p->pre_handler == aggr_pre_handler) {
            list_for_each_entry_rcu(kp, &p->list, list)
--- a/kernel/lockdep.c
+++ b/kernel/lockdep.c
@@ -341,10 +341,7 @@ static const char *usage_str[] =

const char * __get_key_name(struct lockdep_subclass_key *key, char *str)
{
- unsigned long offs, size;
- char *modname;
-
- return kallsyms_lookup((unsigned long)key, &size, &offs, &modname, str);
+ return kallsyms_lookup((unsigned long)key, NULL, NULL, NULL, str);
}

```

```

void
--- a/kernel/module.c
+++ b/kernel/module.c
@@ -2094,8 +2094,10 @@ static const char *get_ksymbol(struct mo
    if (!best)
        return NULL;

- *size = nextval - mod->symtab[best].st_value;
- *offset = addr - mod->symtab[best].st_value;
+ if (size)
+ *size = nextval - mod->symtab[best].st_value;
+ if (offset)
+ *offset = addr - mod->symtab[best].st_value;
    return mod->strtab + mod->symtab[best].st_name;
}

--- a/kernel/time/timer_list.c
+++ b/kernel/time/timer_list.c
@@ -40,11 +40,9 @@ static void print_name_offset(struct seq
{
    unsigned long addr = (unsigned long)sym;
    char namebuf[KSYM_NAME_LEN+1];
- unsigned long size, offset;
- const char *sym_name;
- char *modname;

- sym_name = kallsyms_lookup(addr, &size, &offset, &modname, namebuf);
+ sym_name = kallsyms_lookup(addr, NULL, NULL, NULL, namebuf);
    if (sym_name)
        SEQ_printf(m, "%s", sym_name);
    else
--- a/kernel/time/timer_stats.c
+++ b/kernel/time/timer_stats.c
@@ -258,11 +258,9 @@ void timer_stats_update_stats(void *time
static void print_name_offset(struct seq_file *m, unsigned long addr)
{
    char namebuf[KSYM_NAME_LEN+1];
- unsigned long size, offset;
- const char *sym_name;
- char *modname;

- sym_name = kallsyms_lookup(addr, &size, &offset, &modname, namebuf);
+ sym_name = kallsyms_lookup(addr, NULL, NULL, NULL, namebuf);
    if (sym_name)
        seq_printf(m, "%s", sym_name);
    else

```
