

---

Subject: [PATCH 3/7] uts namespaces: use init\_utsname when appropriate

Posted by [serue](#) on Sat, 08 Apr 2006 04:52:06 GMT

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

---

In some places, particularly drivers and \_\_init code, the init\_utsns is the appropriate one to use. This patch replaces those with a the init\_utsname helper.

Signed-off-by: Serge E. Hallyn <[serue@us.ibm.com](mailto:serue@us.ibm.com)>

---

```
arch/arm/kernel/setup.c          | 2 +-
arch/arm26/kernel/setup.c        | 2 +-
arch/cris/kernel/setup.c         | 2 +-
arch/i386/kernel/process.c        | 6 +++---
arch/i386/kernel/traps.c         | 6 +++---
arch/powerpc/kernel/process.c     | 2 +-
arch/powerpc/kernel/setup_64.c    | 2 +-
arch/powerpc/platforms/pseries/setup.c | 2 +-
arch/sh/kernel/setup.c           | 2 +-
arch/um/kernel/um_arch.c         | 6 +++---
arch/um/sys-x86_64/sysrq.c        | 2 +-
arch/x86_64/kernel/process.c      | 6 +++---
drivers/infiniband/hw/ipath/ipath_verbs.c | 2 +-
drivers/parisc/led.c             | 2 +-
drivers/scsi/lpfc/lpfc_ct.c       | 8 ++++----
drivers/usb/core/hcd.c           | 4 ++--
drivers/usb/gadget/ether.c        | 2 +-
drivers/usb/gadget/file_storage.c | 2 +-
drivers/usb/gadget/serial.c       | 2 +-
drivers/usb/gadget/zero.c         | 2 +-
include/asm-i386/bugs.h          | 2 +-
include/asm-i386/elf.h           | 2 +-
include/asm-sh/bugs.h            | 2 +-
kernel/power/snapshot.c          | 10 +++++-----
net/ipv4/ipconfig.c              | 16 ++++++++-----
net/sunrpc/clnt.c                | 4 ++--
sound/core/info_oss.c            | 10 +++++-----
27 files changed, 55 insertions(+), 55 deletions(-)
```

65af7f7767c4447ccb3fafa6580844f4a65bcfdc

diff --git a/arch/arm/kernel/setup.c b/arch/arm/kernel/setup.c

index 4375284..a4dc8de 100644

--- a/arch/arm/kernel/setup.c

+++ b/arch/arm/kernel/setup.c

@@ -319,7 +319,7 @@ static void \_\_init setup\_processor(void)
 cpu\_name, processor\_id, (int)processor\_id & 15,

```
proc_arch[cpu_architecture()]);
```

```
- sprintf(system_utsname.machine, "%s%c", list->arch_name, ENDIANNESS);  
+ sprintf(init_utsname()->machine, "%s%c", list->arch_name, ENDIANNESS);  
  sprintf(elf_platform, "%s%c", list->elf_name, ENDIANNESS);  
  elf_hwcap = list->elf_hwcap;
```

```
diff --git a/arch/arm26/kernel/setup.c b/arch/arm26/kernel/setup.c  
index 4eb329e..8e6a441 100644  
--- a/arch/arm26/kernel/setup.c  
+++ b/arch/arm26/kernel/setup.c  
@@ -144,7 +144,7 @@ static void __init setup_processor(void)
```

```
    dump_cpu_info();
```

```
- sprintf(system_utsname.machine, "%s", list->arch_name);  
+ sprintf(init_utsname()->machine, "%s", list->arch_name);  
  sprintf(elf_platform, "%s", list->elf_name);  
  elf_hwcap = list->elf_hwcap;
```

```
diff --git a/arch/cris/kernel/setup.c b/arch/cris/kernel/setup.c  
index 619a6ee..1974c01 100644  
--- a/arch/cris/kernel/setup.c  
+++ b/arch/cris/kernel/setup.c  
@@ -161,7 +161,7 @@ setup_arch(char **cmdline_p)  
    show_etrax_copyright();
```

```
    /* Setup utsname */  
- strcpy(system_utsname.machine, cris_machine_name);  
+ strcpy(init_utsname()->machine, cris_machine_name);  
}
```

```
static void *c_start(struct seq_file *m, loff_t *pos)  
diff --git a/arch/i386/kernel/process.c b/arch/i386/kernel/process.c  
index 6259afe..da2e439 100644  
--- a/arch/i386/kernel/process.c  
+++ b/arch/i386/kernel/process.c  
@@ -297,9 +297,9 @@ void show_regs(struct pt_regs * regs)  
    if (user_mode_vm(regs))  
        printk(" ESP: %04x:%08lx", 0xffff & regs->xss, regs->esp);  
        printk(" EFLAGS: %08lx  %s (%s %s)\n",  
-        regs->eflags, print_tainted(), system_utsname.release,  
-        (int)strcspn(system_utsname.version, " "),  
-        system_utsname.version);  
+        regs->eflags, print_tainted(), init_utsname()->release,  
+        (int)strcspn(init_utsname()->version, " "),  
+        init_utsname()->version);  
        printk("EAX: %08lx EBX: %08lx ECX: %08lx EDX: %08lx\n",
```

```

regs->eax,regs->ebx,regs->ecx,regs->edx);
printk("ESI: %08lx EDI: %08lx EBP: %08lx",
diff --git a/arch/i386/kernel/traps.c b/arch/i386/kernel/traps.c
index e385279..dd62423 100644
--- a/arch/i386/kernel/traps.c
+++ b/arch/i386/kernel/traps.c
@@ -260,9 +260,9 @@ void show_registers(struct pt_regs *regs
    printk(KERN_EMERG "CPU:  %d\nEIP:  %04x:[<%08lx>]  %s VLI\n"
        "EFLAGS: %08lx  (%s %.*s) \n",
        smp_processor_id(), 0xffff & regs->xcs, regs->eip,
-   print_tainted(), regs->eflags, system_utsname.release,
-   (int)strcspn(system_utsname.version, " "),
-   system_utsname.version);
+   print_tainted(), regs->eflags, init_utsname()->release,
+   (int)strcspn(init_utsname()->version, " "),
+   init_utsname()->version);
    print_symbol(KERN_EMERG "EIP is at %s\n", regs->eip);
    printk(KERN_EMERG "eax: %08lx ebx: %08lx ecx: %08lx edx: %08lx\n",
        regs->eax, regs->ebx, regs->ecx, regs->edx);
diff --git a/arch/powerpc/kernel/process.c b/arch/powerpc/kernel/process.c
index 2dd47d2..6ce9e10 100644
--- a/arch/powerpc/kernel/process.c
+++ b/arch/powerpc/kernel/process.c
@@ -425,7 +425,7 @@ void show_regs(struct pt_regs * regs)
    printk("NIP: "REG" LR: "REG" CTR: "REG"\n",
        regs->nip, regs->link, regs->ctr);
    printk("REGS: %p TRAP: %04lx  %s (%s)\n",
-   regs, regs->trap, print_tainted(), system_utsname.release);
+   regs, regs->trap, print_tainted(), init_utsname()->release);
    printk("MSR: "REG" ", regs->msr);
    printbits(regs->msr, msr_bits);
    printk(" CR: %08lx XER: %08lx\n", regs->ccr, regs->xer);
diff --git a/arch/powerpc/kernel/setup_64.c b/arch/powerpc/kernel/setup_64.c
index 13e91c4..1c6619f 100644
--- a/arch/powerpc/kernel/setup_64.c
+++ b/arch/powerpc/kernel/setup_64.c
@@ -435,7 +435,7 @@ void __init setup_system(void)
    smp_release_cpus();
#endif

-   printk("Starting Linux PPC64 %s\n", system_utsname.version);
+   printk("Starting Linux PPC64 %s\n", init_utsname()->version);

    printk("-----\n ");
    printk("ppc64_pft_size          = 0x%lx\n", ppc64_pft_size);
diff --git a/arch/powerpc/platforms/pseries/setup.c b/arch/powerpc/platforms/pseries/setup.c
index 5eb55ef..58b7a74 100644
--- a/arch/powerpc/platforms/pseries/setup.c

```

```

+++ b/arch/powerpc/platforms/pseries/setup.c
@@ -255,7 +255,7 @@ static int __init pSeries_init_panel(voi
{
    /* Manually leave the kernel version on the panel. */
    ppc_md.progress("Linux ppc64\n", 0);
- ppc_md.progress(system_utsname.version, 0);
+ ppc_md.progress(init_utsname()->version, 0);

    return 0;
}
diff --git a/arch/sh/kernel/setup.c b/arch/sh/kernel/setup.c
index bb229ef..024401e 100644
--- a/arch/sh/kernel/setup.c
+++ b/arch/sh/kernel/setup.c
@@ -481,7 +481,7 @@ static int show_cpuinfo(struct seq_file
    seq_printf(m, "machine\t\t: %s\n", get_system_type());

    seq_printf(m, "processor\t: %d\n", cpu);
- seq_printf(m, "cpu family\t: %s\n", system_utsname.machine);
+ seq_printf(m, "cpu family\t: %s\n", init_utsname()->machine);
    seq_printf(m, "cpu type\t: %s\n", get_cpu_subtype());

    show_cpuflags(m);
diff --git a/arch/um/kernel/um_arch.c b/arch/um/kernel/um_arch.c
index 7d51dd7..b49dd7d 100644
--- a/arch/um/kernel/um_arch.c
+++ b/arch/um/kernel/um_arch.c
@@ -167,7 +167,7 @@ static char *usage_string =

static int __init uml_version_setup(char *line, int *add)
{
- printf("%s\n", system_utsname.release);
+ printf("%s\n", init_utsname()->release);
    exit(0);

    return 0;
@@ -278,7 +278,7 @@ static int __init Usage(char *line, int
{
    const char **p;

- printf(usage_string, system_utsname.release);
+ printf(usage_string, init_utsname()->release);
    p = &__uml_help_start;
    while (p < &__uml_help_end) {
        printf("%s", *p);
@@ -400,7 +400,7 @@ int linux_main(int argc, char **argv)
    /* Reserve up to 4M after the current brk */
    uml_reserved = ROUND_4M(brk_start) + (1 << 22);

```

```

- setup_machinename(system_utsname.machine);
+ setup_machinename(init_utsname()->machine);

#ifdef CONFIG_CMDLINE_ON_HOST
    argv1_begin = argv[1];
diff --git a/arch/um/sys-x86_64/sysrq.c b/arch/um/sys-x86_64/sysrq.c
index d0a25af..ce3e07f 100644
--- a/arch/um/sys-x86_64/sysrq.c
+++ b/arch/um/sys-x86_64/sysrq.c
@@ -16,7 +16,7 @@ void __show_regs(struct pt_regs * regs)
    printk("\n");
    print_modules();
    printk("Pid: %d, comm: %.20s %s %s\n",
-       current->pid, current->comm, print_tainted(), system_utsname.release);
+       current->pid, current->comm, print_tainted(), init_utsname()->release);
    printk("RIP: %04lx:[<%016lx>] ", PT_REGS_CS(regs) & 0xffff,
        PT_REGS_RIP(regs));
    printk("\nRSP: %016lx EFLAGS: %08lx\n", PT_REGS_RSP(regs),
diff --git a/arch/x86_64/kernel/process.c b/arch/x86_64/kernel/process.c
index 70dd8e5..79f8174 100644
--- a/arch/x86_64/kernel/process.c
+++ b/arch/x86_64/kernel/process.c
@@ -292,9 +292,9 @@ void __show_regs(struct pt_regs * regs)
    print_modules();
    printk("Pid: %d, comm: %.20s %s %s %.s\n",
        current->pid, current->comm, print_tainted(),
-   system_utsname.release,
-   (int)strcspn(system_utsname.version, " "),
-   system_utsname.version);
+   init_utsname()->release,
+   (int)strcspn(init_utsname()->version, " "),
+   init_utsname()->version);
    printk("RIP: %04lx:[<%016lx>] ", regs->cs & 0xffff, regs->rip);
    printk_address(regs->rip);
    printk("\nRSP: %04lx:%016lx EFLAGS: %08lx\n", regs->ss, regs->rsp,
diff --git a/drivers/infiniband/hw/ipath/ipath_verbs.c b/drivers/infiniband/hw/ipath/ipath_verbs.c
index 9f27fd3..f873380 100644
--- a/drivers/infiniband/hw/ipath/ipath_verbs.c
+++ b/drivers/infiniband/hw/ipath/ipath_verbs.c
@@ -1048,7 +1048,7 @@ static void *ipath_register_ib_device(in
    dev->process_mad = ipath_process_mad;

    snprintf(dev->node_desc, sizeof(dev->node_desc),
-   IPATH_IDSTR " %s kernel_SMA", system_utsname.nodename);
+   IPATH_IDSTR " %s kernel_SMA", init_utsname()->nodename);

    ret = ib_register_device(dev);

```

```

if (ret)
diff --git a/drivers/parisc/led.c b/drivers/parisc/led.c
index 298f2dd..1d778d2 100644
--- a/drivers/parisc/led.c
+++ b/drivers/parisc/led.c
@@ -684,7 +684,7 @@ int __init led_init(void)
int ret;

    snprintf(lcd_text_default, sizeof(lcd_text_default),
- "Linux %s", system_utsname.release);
+ "Linux %s", init_utsname()->release);

/* Work around the buggy PDC of KittyHawk-machines */
switch (CPU_HVERSION) {
diff --git a/drivers/scsi/lpfc/lpfc_ct.c b/drivers/scsi/lpfc/lpfc_ct.c
index b65ee57..83f53fb 100644
--- a/drivers/scsi/lpfc/lpfc_ct.c
+++ b/drivers/scsi/lpfc/lpfc_ct.c
@@ -961,8 +961,8 @@ lpfc_fdmi_cmd(struct lpfc_hba * phba, st
    ae = (ATTRIBUTE_ENTRY *) ((uint8_t *) rh + size);
    ae->ad.bits.AttrType = be16_to_cpu(OS_NAME_VERSION);
    sprintf(ae->un.OsNameVersion, "%s %s %s",
- system_utsname.sysname, system_utsname.release,
- system_utsname.version);
+ init_utsname()->sysname, init_utsname()->release,
+ init_utsname()->version);
    len = strlen(ae->un.OsNameVersion);
    len += (len & 3) ? (4 - (len & 3)) : 4;
    ae->ad.bits.AttrLen = be16_to_cpu(FOURBYTES + len);
@@ -1080,7 +1080,7 @@ lpfc_fdmi_cmd(struct lpfc_hba * phba, st
    size);
    ae->ad.bits.AttrType = be16_to_cpu(HOST_NAME);
    sprintf(ae->un.HostName, "%s",
- system_utsname.nodename);
+ init_utsname()->nodename);
    len = strlen(ae->un.HostName);
    len += (len & 3) ? (4 - (len & 3)) : 4;
    ae->ad.bits.AttrLen =
@@ -1168,7 +1168,7 @@ lpfc_fdmi_tmo_handler(struct lpfc_hba *p

    ndlp = lpfc_findnode_did(phba, NLP_SEARCH_ALL, FDMI_DID);
    if (ndlp) {
- if (system_utsname.nodename[0] != '\0') {
+ if (init_utsname()->nodename[0] != '\0') {
    lpfc_fdmi_cmd(phba, ndlp, SLI_MGMT_DHBA);
    } else {
    mod_timer(&phba->fc_fdmitmo, jiffies + HZ * 60);
diff --git a/drivers/usb/core/hcd.c b/drivers/usb/core/hcd.c

```

```

index fbd938d..c1255ec 100644
--- a/drivers/usb/core/hcd.c
+++ b/drivers/usb/core/hcd.c
@@ -318,8 +318,8 @@ static int rh_string (

    // id 3 == vendor description
    } else if (id == 3) {
-   snprintf (buf, sizeof buf, "%s %s %s", system_utsname.sysname,
-   system_utsname.release, hcd->driver->description);
+   snprintf (buf, sizeof buf, "%s %s %s", init_utsname()->sysname,
+   init_utsname()->release, hcd->driver->description);

    // unsupported IDs --> "protocol stall"
    } else
diff --git a/drivers/usb/gadget/ether.c b/drivers/usb/gadget/ether.c
index c3d8e5c..e6fe999 100644
--- a/drivers/usb/gadget/ether.c
+++ b/drivers/usb/gadget/ether.c
@@ -2242,7 +2242,7 @@ @@ eth_bind (struct usb_gadget *gadget)
    return -ENODEV;
}
    snprintf (manufacturer, sizeof manufacturer, "%s %s/%s",
-   system_utsname.sysname, system_utsname.release,
+   init_utsname()->sysname, init_utsname()->release,
    gadget->name);

    /* If there's an RNDIS configuration, that's what Windows wants to
diff --git a/drivers/usb/gadget/file_storage.c b/drivers/usb/gadget/file_storage.c
index cf3be29..d3149e5 100644
--- a/drivers/usb/gadget/file_storage.c
+++ b/drivers/usb/gadget/file_storage.c
@@ -3965,7 +3965,7 @@ @@ static int __init fsg_bind(struct usb_ga
    usb_gadget_set_selfpowered(gadget);

    snprintf(manufacturer, sizeof manufacturer, "%s %s with %s",
-   system_utsname.sysname, system_utsname.release,
+   init_utsname()->sysname, init_utsname()->release,
    gadget->name);

    /* On a real device, serial[] would be loaded from permanent
diff --git a/drivers/usb/gadget/serial.c b/drivers/usb/gadget/serial.c
index b992546..a2f905b 100644
--- a/drivers/usb/gadget/serial.c
+++ b/drivers/usb/gadget/serial.c
@@ -1496,7 +1496,7 @@ @@ static int __init gs_bind(struct usb_gad
    return -ENOMEM;

    snprintf(manufacturer, sizeof(manufacturer), "%s %s with %s",

```

```

- system_utsname.sysname, system_utsname.release,
+ init_utsname()->sysname, init_utsname()->release,
  gadget->name);

  memset(dev, 0, sizeof(struct gs_dev));
diff --git a/drivers/usb/gadget/zero.c b/drivers/usb/gadget/zero.c
index 51424f6..b23e0fd 100644
--- a/drivers/usb/gadget/zero.c
+++ b/drivers/usb/gadget/zero.c
@@ -1240,7 +1240,7 @@ autoconf_fail:
     EP_OUT_NAME, EP_IN_NAME);

    snprintf (manufacturer, sizeof manufacturer, "%s %s with %s",
- system_utsname.sysname, system_utsname.release,
+ init_utsname()->sysname, init_utsname()->release,
  gadget->name);

  return 0;
diff --git a/include/asm-i386/bugs.h b/include/asm-i386/bugs.h
index 50233e0..6cb79fe 100644
--- a/include/asm-i386/bugs.h
+++ b/include/asm-i386/bugs.h
@@ -190,6 +190,6 @@ static void __init check_bugs(void)
    check_fpu();
    check_hlt();
    check_popad();
- system_utsname.machine[1] = '0' + (boot_cpu_data.x86 > 6 ? 6 : boot_cpu_data.x86);
+ init_utsname()->machine[1] = '0' + (boot_cpu_data.x86 > 6 ? 6 : boot_cpu_data.x86);
    alternative_instructions();
}
diff --git a/include/asm-i386/elf.h b/include/asm-i386/elf.h
index 4153d80..8d455e2 100644
--- a/include/asm-i386/elf.h
+++ b/include/asm-i386/elf.h
@@ -108,7 +108,7 @@ typedef struct user_fxsr_struct elf_fpxr
    For the moment, we have only optimizations for the Intel generations,
    but that could change... */

-#define ELF_PLATFORM (system_utsname.machine)
+#define ELF_PLATFORM (init_utsname()->machine)

#ifdef __KERNEL__
#define SET_PERSONALITY(ex, ibcs2) do { } while (0)
diff --git a/include/asm-sh/bugs.h b/include/asm-sh/bugs.h
index a6de3d0..d09933c 100644
--- a/include/asm-sh/bugs.h
+++ b/include/asm-sh/bugs.h
@@ -18,7 +18,7 @@ static void __init check_bugs(void)

```



```

{
extern char *get_cpu_subtype(void);
extern unsigned long loops_per_jiffy;
- char *p= &system_utsname.machine[2]; /* "sh" */
+ char *p= &init_utsname()->machine[2]; /* "sh" */

cpu_data->loops_per_jiffy = loops_per_jiffy;

diff --git a/kernel/power/snapshot.c b/kernel/power/snapshot.c
index c5863d0..f43f7db 100644
--- a/kernel/power/snapshot.c
+++ b/kernel/power/snapshot.c
@@ -523,7 +523,7 @@ static void init_header(struct swsusp_in
    memset(info, 0, sizeof(struct swsusp_info));
    info->version_code = LINUX_VERSION_CODE;
    info->num_physpages = num_physpages;
- memcpy(&info->uts, &system_utsname, sizeof(system_utsname));
+ memcpy(&info->uts, init_utsname(), sizeof(struct new_utsname));
    info->cpus = num_online_cpus();
    info->image_pages = nr_copy_pages;
    info->pages = nr_copy_pages + nr_meta_pages + 1;
@@ -662,13 +662,13 @@ static int check_header(struct swsusp_in
    reason = "kernel version";
    if (info->num_physpages != num_physpages)
        reason = "memory size";
- if (strcmp(info->uts.sysname, system_utsname.sysname))
+ if (strcmp(info->uts.sysname, init_utsname()->sysname))
        reason = "system type";
- if (strcmp(info->uts.release, system_utsname.release))
+ if (strcmp(info->uts.release, init_utsname()->release))
        reason = "kernel release";
- if (strcmp(info->uts.version, system_utsname.version))
+ if (strcmp(info->uts.version, init_utsname()->version))
        reason = "version";
- if (strcmp(info->uts.machine, system_utsname.machine))
+ if (strcmp(info->uts.machine, init_utsname()->machine))
        reason = "machine";
    if (reason) {
        printk(KERN_ERR "swsusp: Resume mismatch: %s\n", reason);
diff --git a/net/ipv4/ipconfig.c b/net/ipv4/ipconfig.c
index cb8a92f..81db372 100644
--- a/net/ipv4/ipconfig.c
+++ b/net/ipv4/ipconfig.c
@@ -367,7 +367,7 @@ static int __init ic_defaults(void)
    */

    if (!ic_host_name_set)
-    sprintf(system_utsname.nodename, "%u.%u.%u.%u", NIPQUAD(ic_myaddr));

```

```

+ sprintf(init_utsname()->nodename, "%u.%u.%u.%u", NIPQUAD(ic_myaddr));

if (root_server_addr == INADDR_NONE)
    root_server_addr = ic_servaddr;
@@ -806,7 +806,7 @@ static void __init ic_do_bootp_ext(u8 *e
}
break;
case 12: /* Host name */
- ic_bootp_string(system_utsname.nodename, ext+1, *ext, __NEW_UTS_LEN);
+ ic_bootp_string(init_utsname()->nodename, ext+1, *ext, __NEW_UTS_LEN);
    ic_host_name_set = 1;
    break;
case 15: /* Domain name (DNS) */
@@ -817,7 +817,7 @@ static void __init ic_do_bootp_ext(u8 *e
    ic_bootp_string(root_server_path, ext+1, *ext, sizeof(root_server_path));
    break;
case 40: /* NIS Domain name (_not_ DNS) */
- ic_bootp_string(system_utsname.domainname, ext+1, *ext, __NEW_UTS_LEN);
+ ic_bootp_string(init_utsname()->domainname, ext+1, *ext, __NEW_UTS_LEN);
    break;
}
}
@@ -1369,7 +1369,7 @@ static int __init ip_auto_config(void)
    printk(", mask=%u.%u.%u.%u", NIPQUAD(ic_netmask));
    printk(", gw=%u.%u.%u.%u", NIPQUAD(ic_gateway));
    printk(",\n    host=%s, domain=%s, nis-domain=%s",
-        system_utsname.nodename, ic_domain, system_utsname.domainname);
+        init_utsname()->nodename, ic_domain, init_utsname()->domainname);
    printk(",\n    bootserver=%u.%u.%u.%u", NIPQUAD(ic_servaddr));
    printk(", rootserver=%u.%u.%u.%u", NIPQUAD(root_server_addr));
    printk(", rootpath=%s", root_server_path);
@@ -1479,11 +1479,11 @@ static int __init ip_auto_config_setup(c
case 4:
    if ((dp = strchr(ip, '.')) {
        *dp++ = '\0';
-        strcpy(system_utsname.domainname, dp,
-            sizeof(system_utsname.domainname));
+        strcpy(init_utsname()->domainname, dp,
+            sizeof(init_utsname()->domainname));
    }
-    strcpy(system_utsname.nodename, ip,
-        sizeof(system_utsname.nodename));
+    strcpy(init_utsname()->nodename, ip,
+        sizeof(init_utsname()->nodename));
    ic_host_name_set = 1;
    break;
case 5:
diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c

```

index aa8965e..97c8439 100644

--- a/net/sunrpc/clnt.c

+++ b/net/sunrpc/clnt.c

```
@@ -176,10 +176,10 @@ rpc_new_client(struct rpc_xprt *xprt, ch
 }
```

```
/* save the nodename */
- clnt->cl_nodelen = strlen(system_utsname.nodename);
+ clnt->cl_nodelen = strlen(init_utsname()->nodename);
  if (clnt->cl_nodelen > UNIX_MAXNODENAME)
    clnt->cl_nodelen = UNIX_MAXNODENAME;
- memcpy(clnt->cl_nodename, system_utsname.nodename, clnt->cl_nodelen);
+ memcpy(clnt->cl_nodename, init_utsname()->nodename, clnt->cl_nodelen);
  return clnt;
```

out\_no\_auth:

diff --git a/sound/core/info\_oss.c b/sound/core/info\_oss.c

index f9ce854..35662bb 100644

--- a/sound/core/info\_oss.c

+++ b/sound/core/info\_oss.c

```
@@ -94,11 +94,11 @@ static void snd_sndstat_proc_read(struct
 {
   snd_iprintf(buffer, "Sound Driver:3.8.1a-980706 (ALSA v" CONFIG_SND_VERSION " emulation
code)\n");
   snd_iprintf(buffer, "Kernel: %s %s %s %s %s\n",
-   system_utsname.sysname,
-   system_utsname.nodename,
-   system_utsname.release,
-   system_utsname.version,
-   system_utsname.machine);
+   init_utsname()->sysname,
+   init_utsname()->nodename,
+   init_utsname()->release,
+   init_utsname()->version,
+   init_utsname()->machine);
   snd_iprintf(buffer, "Config options: 0\n");
   snd_iprintf(buffer, "\nInstalled drivers: \n");
   snd_iprintf(buffer, "Type 10: ALSA emulation\n");
--
```

1.2.4

---

Subject: Re: [PATCH 3/7] uts namespaces: use init\_utsname when appropriate  
Posted by [ebiederm](#) on Sat, 08 Apr 2006 07:09:03 GMT

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

---

"Serge E. Hallyn" <serue@us.ibm.com> writes:

```

> diff --git a/include/asm-i386/elf.h b/include/asm-i386/elf.h
> index 4153d80..8d455e2 100644
> --- a/include/asm-i386/elf.h
> +++ b/include/asm-i386/elf.h
> @@ -108,7 +108,7 @@ typedef struct user_fxr_struct elf_fpxr
>   For the moment, we have only optimizations for the Intel generations,
>   but that could change... */
>
> -#define ELF_PLATFORM (system_utsname.machine)
> +#define ELF_PLATFORM (init_utsname()->machine)
>
> #ifdef __KERNEL__
> #define SET_PERSONALITY(ex, ibcs2) do { } while (0)

```

I think this one needs to be utsname()->machine.

Currently it doesn't matter. But Herbert has expressed the desire to make a machine appear like an older one.

```

> diff --git a/net/ipv4/ipconfig.c b/net/ipv4/ipconfig.c
> index cb8a92f..81db372 100644
> --- a/net/ipv4/ipconfig.c
> +++ b/net/ipv4/ipconfig.c
> @@ -367,7 +367,7 @@ static int __init ic_defaults(void)
>   */
>
>   if (!ic_host_name_set)
> -   sprintf(system_utsname.nodename, "%u.%u.%u.%u", NIPQUAD(ic_myaddr));
> +   sprintf(init_utsname()->nodename, "%u.%u.%u.%u", NIPQUAD(ic_myaddr));
>
>   if (root_server_addr == INADDR_NONE)
>     root_server_addr = ic_servaddr;
> @@ -806,7 +806,7 @@ static void __init ic_do_bootp_ext(u8 *e
>   }
>   break;
>   case 12: /* Host name */
> -   ic_bootp_string(system_utsname.nodename, ext+1, *ext, __NEW_UTS_LEN);
> +   ic_bootp_string(init_utsname()->nodename, ext+1, *ext, __NEW_UTS_LEN);
>   ic_host_name_set = 1;
>   break;
>   case 15: /* Domain name (DNS) */
> @@ -817,7 +817,7 @@ static void __init ic_do_bootp_ext(u8 *e
>     ic_bootp_string(root_server_path, ext+1, *ext,
> sizeof(root_server_path));
>     break;
>   case 40: /* NIS Domain name (_not_ DNS) */

```

```

> - ic_bootp_string(system_utsname.domainname, ext+1, *ext, __NEW_UTS_LEN);
> + ic_bootp_string(init_utsname()->domainname, ext+1, *ext, __NEW_UTS_LEN);
>   break;
> }
> }
> @@ -1369,7 +1369,7 @@ static int __init ip_auto_config(void)
>   printk(", mask=%u.%u.%u.%u", NIPQUAD(ic_netmask));
>   printk(", gw=%u.%u.%u.%u", NIPQUAD(ic_gateway));
>   printk("\n   host=%s, domain=%s, nis-domain=%s",
> -       system_utsname.nodename, ic_domain, system_utsname.domainname);
> + init_utsname()->nodename, ic_domain, init_utsname()->domainname);
>   printk("\n   bootserver=%u.%u.%u.%u", NIPQUAD(ic_servaddr));
>   printk(", rootserver=%u.%u.%u.%u", NIPQUAD(root_server_addr));
>   printk(", rootpath=%s", root_server_path);
> @@ -1479,11 +1479,11 @@ static int __init ip_auto_config_setup(c
>   case 4:
>     if ((dp = strchr(ip, '.')) {
>       *dp++ = '\0';
> -     strcpy(system_utsname.domainname, dp,
> - sizeof(system_utsname.domainname));
> +     strcpy(init_utsname()->domainname, dp,
> + sizeof(init_utsname()->domainname));
>     }
> -     strcpy(system_utsname.nodename, ip,
> - sizeof(system_utsname.nodename));
> +     strcpy(init_utsname()->nodename, ip,
> + sizeof(init_utsname()->nodename));
>     ic_host_name_set = 1;
>     break;
>   case 5:

```

This also probably makes sense as utsname(). It doesn't really matter as this is before init is executed. But logically this is a user space or per namespace action.

```

> diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
> index aa8965e..97c8439 100644
> --- a/net/sunrpc/clnt.c
> +++ b/net/sunrpc/clnt.c
> @@ -176,10 +176,10 @@ rpc_new_client(struct rpc_xprt *xprt, ch
>   }
>
>   /* save the nodename */
> - clnt->cl_nodelen = strlen(system_utsname.nodename);
> + clnt->cl_nodelen = strlen(init_utsname()->nodename);
>   if (clnt->cl_nodelen > UNIX_MAXNODENAME)
>     clnt->cl_nodelen = UNIX_MAXNODENAME;
> - memcpy(clnt->cl_nodename, system_utsname.nodename, clnt->cl_nodelen);

```

```
> + memcpy(clnt->cl_nodename, init_utsname()->nodename, clnt->cl_nodelen);
> return clnt;
>
> out_no_auth:
```

Using nodename is practically the definition of something that should per namespace I think. Plus it would be really inconsistent to use utsname() and the init\_utsname for the nfs rpc calls.

Unless I am missing something.

Eric

---

---

Subject: Re: [PATCH 3/7] uts namespaces: use init\_utsname when appropriate  
Posted by [serue](#) on Sat, 08 Apr 2006 20:27:01 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Quoting Eric W. Biederman (ebiederm@xmission.com):

```
> "Serge E. Hallyn" <serue@us.ibm.com> writes:
>
> > diff --git a/include/asm-i386/elf.h b/include/asm-i386/elf.h
> > index 4153d80..8d455e2 100644
> > --- a/include/asm-i386/elf.h
> > +++ b/include/asm-i386/elf.h
> > @@ -108,7 +108,7 @@ typedef struct user_fxsr_struct elf_fpxr
> >    For the moment, we have only optimizations for the Intel generations,
> >    but that could change... */
> >
> > #define ELF_PLATFORM (system_utsname.machine)
> > +#define ELF_PLATFORM (init_utsname()->machine)
> >
> > #ifdef __KERNEL__
> > #define SET_PERSONALITY(ex, ibcs2) do { } while (0)
>
> I think this one needs to be utsname()->machine.
>
> Currently it doesn't matter. But Herbert has expressed
> the desire to make a machine appear like an older one.
```

Ok.

```
> > diff --git a/net/ipv4/ipconfig.c b/net/ipv4/ipconfig.c
> > index cb8a92f..81db372 100644
...
> > @@ -1479,11 +1479,11 @@ static int __init ip_auto_config_setup(c
> >    case 4:
> >    if ((dp = strchr(ip, '.'))) {
```

```

>> *dp++ = '\0';
>> - strcpy(system_utsname.domainname, dp,
>> - sizeof(system_utsname.domainname));
>> + strcpy(init_utsname()->domainname, dp,
>> + sizeof(init_utsname()->domainname));
>> }
>> - strcpy(system_utsname.nodename, ip,
>> - sizeof(system_utsname.nodename));
>> + strcpy(init_utsname()->nodename, ip,
>> + sizeof(init_utsname()->nodename));
>> ic_host_name_set = 1;
>> break;
>> case 5:
>
> This also probably makes sense as utsname(). It doesn't
> really matter as this is before init is executed. But logically
> this is a user space or per namespace action.

```

Right, I was kind of favoring using `init_utsname()` for anything `__init`. But `utsname()` will of course work just as well there.

```

>> diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
>> index aa8965e..97c8439 100644
>> --- a/net/sunrpc/clnt.c
>> +++ b/net/sunrpc/clnt.c
>> @@ -176,10 +176,10 @@ rpc_new_client(struct rpc_xprt *xprt, ch
>> }
>>
>> /* save the nodename */
>> - clnt->cl_nodelen = strlen(system_utsname.nodename);
>> + clnt->cl_nodelen = strlen(init_utsname()->nodename);
>> if (clnt->cl_nodelen > UNIX_MAXNODENAME)
>>     clnt->cl_nodelen = UNIX_MAXNODENAME;
>> - memcpy(clnt->cl_nodename, system_utsname.nodename, clnt->cl_nodelen);
>> + memcpy(clnt->cl_nodename, init_utsname()->nodename, clnt->cl_nodelen);
>> return clnt;
>>
>> out_no_auth:
>
> Using nodename is practically the definition of something
> that should per namespace I think. Plus it would be really inconsistent
> to use utsname() and the init_utsname for the nfs rpc calls.
>
> Unless I am missing something.

```

It seemed like this would be happening in any old context, so that `current->uts_ns` could be any process'. Tracing it back further, it seems like `nfs+lockd` should have the context available. So I'll

switch this as well.

thanks,  
-serge

---

Subject: Re: [PATCH 3/7] uts namespaces: use init\_utsname when appropriate  
Posted by [Sam Vilain](#) on Sat, 08 Apr 2006 23:44:39 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Sat, 2006-04-08 at 01:09 -0600, Eric W. Biederman wrote:

```
> > -#define ELF_PLATFORM (system_utsname.machine)
> > +#define ELF_PLATFORM (init_utsname()->machine)
> >
> > #ifdef __KERNEL__
> > #define SET_PERSONALITY(ex, ibcs2) do { } while (0)
> I think this one needs to be utsname()->machine.
> Currently it doesn't matter. But Herbert has expressed
> the desire to make a machine appear like an older one.
```

This is extremely useful for faking it as "i386" on x86\_64 systems, for instance.

Sam.

---

Subject: Re: Re: [PATCH 3/7] uts namespaces: use init\_utsname when appropriate  
Posted by [kir](#) on Sun, 09 Apr 2006 00:12:58 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Sam Vilain wrote:

```
>On Sat, 2006-04-08 at 01:09 -0600, Eric W. Biederman wrote:
>
>
>>>#define ELF_PLATFORM (system_utsname.machine)
>>>+#define ELF_PLATFORM (init_utsname()->machine)
>>>
>>> #ifdef __KERNEL__
>>> #define SET_PERSONALITY(ex, ibcs2) do { } while (0)
>>>
>>>
>>I think this one needs to be utsname()->machine.
>>Currently it doesn't matter. But Herbert has expressed
>>the desire to make a machine appear like an older one.
>>
>>
```



>  
>This is extremely useful for faking it as "i386" on x86\_64 systems, for  
>instance.

>  
>

Could 'setarch' be of any help here? Works fine for us. Or am I missing something?

---

Subject: Re: Re: [PATCH 3/7] uts namespaces: use init\_utsname when appropriate  
Posted by [ebiederm](#) on Sun, 09 Apr 2006 09:25:04 GMT

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

---

Kir Kolyshkin <[kir@openvz.org](mailto:kir@openvz.org)> writes:

> Sam Vilain wrote:

>

>>On Sat, 2006-04-08 at 01:09 -0600, Eric W. Biederman wrote:

>>

>>

>>>>-#define ELF\_PLATFORM (system\_utsname.machine)

>>>>+#define ELF\_PLATFORM (init\_utsname()->machine)

>>>> #ifdef \_\_KERNEL\_\_

>>>> #define SET\_PERSONALITY(ex, ibcs2) do { } while (0)

>>>>

>>>>

>>>I think this one needs to be utsname()->machine.

>>>Currently it doesn't matter. But Herbert has expressed

>>>the desire to make a machine appear like an older one.

>>>

>>>

>>

>>This is extremely useful for faking it as "i386" on x86\_64 systems, for

>>instance.

>>

>>

> Could 'setarch' be of any help here? Works fine for us. Or am I missing

> something?

For the specific case that is clearly the better solution,  
as it already exists, and it handles the weird 32/64bit  
logic. The ELF\_PLATFORM bit I was commenting on was 32bit  
only.

I'm not ready to implement any new functionality at the moment,  
but what I heard suggested and was it may be reasonable to allow  
machine to be modified on a per uts namespace basis. If that  
kind of thing is ever to happen ELF\_PLATFORM needs to be per

uts on x86. Actually allowing modification of machine is an entirely different conversation.

Eric

---

---

Subject: Re: [PATCH 3/7] uts namespaces: use init\_utsname when appropriate  
Posted by [ebiederm](#) on Sun, 09 Apr 2006 09:44:19 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

"Serge E. Hallyn" <serue@us.ibm.com> writes:

```
>> This also probably makes sense as utsname(). It doesn't
>> really matter as this is before init is executed. But logically
>> this is a user space or per namespace action.
>
> Right, I was kind of favoring using init_utsname() for anything
> __init. But utsname() will of course work just as well there.
```

Basically anything that should move to klibc I favor using  
utsname() for. That tends to make it clear it follows  
the usual user space rules.

With a little luck HPA might actually have this code deleted  
in -mm before we get to far.

```
>> > diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
>> > index aa8965e..97c8439 100644
>> > --- a/net/sunrpc/clnt.c
>> > +++ b/net/sunrpc/clnt.c
>> > @@ -176,10 +176,10 @@ rpc_new_client(struct rpc_xprt *xprt, ch
>> > }
>> >
>> > /* save the nodename */
>> > - clnt->cl_nodelen = strlen(system_utsname.nodename);
>> > + clnt->cl_nodelen = strlen(init_utsname()->nodename);
>> > if (clnt->cl_nodelen > UNX_MAXNODENAME)
>> >   clnt->cl_nodelen = UNX_MAXNODENAME;
>> > - memcpy(clnt->cl_nodename, system_utsname.nodename, clnt->cl_nodelen);
>> > + memcpy(clnt->cl_nodename, init_utsname()->nodename, clnt->cl_nodelen);
>> > return clnt;
>> >
>> > out_no_auth:
>> >
>> Using nodename is practically the definition of something
>> that should per namespace I think. Plus it would be really inconsistent
>> to use utsname() and the init_utsname for the nfs rpc calls.
>>
```

>> Unless I am missing something.

>

> It seemed like this would be happening in any old context, so that  
> current->uts\_ns could be any process'. Tracing it back further,  
> it seems like nfs+lockd should have the context available. So I'll  
> switch this as well.

I have not traced that path recently. So I don't remember.  
This is one of those odd cases that makes a real difference.

This reminds me of another piece of the conversation.  
kernel\_thread vs. kthread, and the oddities of daemonize.

In general user space cannot kill kernel threads, so having  
a kernel thread inside a namespace is dangerous because it  
means the namespace can never exit.

There are two ways to avoid the associated problems.

- modify daemonize to always use the instance of that  
namespace associated with init\_task.
- modify all interesting kernel threads to use the  
kthread api instead of kernel\_thread. Using kthread  
makes the kernel threads children of keventd and always  
in the initial namespace instance. As such we know  
we aren't inside of any user space namespace instance.

Eric

---

Subject: Re: [PATCH 3/7] uts namespaces: use init\_utsname when appropriate  
Posted by [Christoph Hellwig](#) on Sun, 09 Apr 2006 10:14:36 GMT

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

---

On Sun, Apr 09, 2006 at 03:44:19AM -0600, Eric W. Biederman wrote:

- > There are two ways to avoid the associated problems.
- > - modify daemonize to always use the instance of that  
> namespace associated with init\_task.
- > - modify all interesting kernel threads to use the  
> kthread api instead of kernel\_thread. Using kthread  
> makes the kernel threads children of keventd and always  
> in the initial namespace instance. As such we know  
> we aren't inside of any user space namespace instance.

I've added a deprecation entry for the kernel\_thread export and plan  
to convert all users to the kthread API. Any help on that is of course  
greatly appreciated.

---

---

Subject: Re: [PATCH 3/7] uts namespaces: use init\_utsname when appropriate  
Posted by [serue](#) on Mon, 10 Apr 2006 20:39:03 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Quoting Eric W. Biederman (ebiederm@xmission.com):

> "Serge E. Hallyn" <serue@us.ibm.com> writes:  
>  
> >> This also probably makes sense as utsname(). It doesn't  
> >> really matter as this is before init is executed. But logically  
> >> this is a user space or per namespace action.  
> >  
> > Right, I was kind of favoring using init\_utsname() for anything  
> > \_\_init. But utsname() will of course work just as well there.  
>  
> Basically anything that should move to klibc I favor using  
> utsname() for. That tends to make it clear it follows  
> the usual user space rules.  
>  
> With a little luck HPA might actually have this code deleted  
> in -mm before we get to far.

Here is a new version of the init\_utsname patch. I'm sending a new version of the utsname() patch in reply to the original [2/7] patch.

From: Serge Hallyn <serue@us.ibm.com>  
Subject: [PATCH 3/7] uts namespaces: use init\_utsname when appropriate

In some places, particularly drivers and \_\_init code, the init utsns is the appropriate one to use. This patch replaces those with a the init\_utsname helper.

Changes: Removed several uses of init\_utsname(). Hope I picked all the right ones in net/ipv4/ipconfig.c. These are now changed to utsname() (the per-process namespace utsname) in the previous patch (2/7)

Signed-off-by: Serge E. Hallyn <serue@us.ibm.com>

---

|                         |                                   |                                  |                                    |                                      |                                           |                                        |                                                |          |
|-------------------------|-----------------------------------|----------------------------------|------------------------------------|--------------------------------------|-------------------------------------------|----------------------------------------|------------------------------------------------|----------|
| arch/arm/kernel/setup.c | 2 +-<br>arch/arm26/kernel/setup.c | 2 +-<br>arch/cris/kernel/setup.c | 2 +-<br>arch/i386/kernel/process.c | 6 +++---<br>arch/i386/kernel/traps.c | 6 +++---<br>arch/powerpc/kernel/process.c | 2 +-<br>arch/powerpc/kernel/setup_64.c | 2 +-<br>arch/powerpc/platforms/pseries/setup.c | 2 +-<br> |
|-------------------------|-----------------------------------|----------------------------------|------------------------------------|--------------------------------------|-------------------------------------------|----------------------------------------|------------------------------------------------|----------|

```

arch/sh/kernel/setup.c          | 2 +-
arch/um/kernel/um_arch.c       | 6 +++---
arch/um/sys-x86_64/sysrq.c      | 2 +-
arch/x86_64/kernel/process.c    | 6 +++---
drivers/infiniband/hw/ipath/ipath_verbs.c | 2 +-
drivers/parisc/led.c           | 2 +-
drivers/scsi/lpfc/lpfc_ct.c     | 8 ++++----
drivers/usb/core/hcd.c         | 4 ++--
drivers/usb/gadget/ether.c      | 2 +-
drivers/usb/gadget/file_storage.c | 2 +-
drivers/usb/gadget/serial.c    | 2 +-
drivers/usb/gadget/zero.c      | 2 +-
include/asm-i386/bugs.h        | 2 +-
include/asm-sh/bugs.h          | 2 +-
kernel/power/snapshot.c        | 10 +++++-----
net/ipv4/ipconfig.c            | 2 +-
sound/core/info_oss.c          | 10 +++++-----
25 files changed, 45 insertions(+), 45 deletions(-)

```

3941501899f74ae1fee21a074cdc2e420a4b3f27

diff --git a/arch/arm/kernel/setup.c b/arch/arm/kernel/setup.c

index 4375284..a4dc8de 100644

--- a/arch/arm/kernel/setup.c

+++ b/arch/arm/kernel/setup.c

```

@@ -319,7 +319,7 @@ static void __init setup_processor(void)
     cpu_name, processor_id, (int)processor_id & 15,
     proc_arch[cpu_architecture()]);

```

```

- sprintf(system_utsname.machine, "%s%c", list->arch_name, ENDIANNESS);
+ sprintf(init_utsname()->machine, "%s%c", list->arch_name, ENDIANNESS);
  sprintf(elf_platform, "%s%c", list->elf_name, ENDIANNESS);
  elf_hwcap = list->elf_hwcap;

```

diff --git a/arch/arm26/kernel/setup.c b/arch/arm26/kernel/setup.c

index 4eb329e..8e6a441 100644

--- a/arch/arm26/kernel/setup.c

+++ b/arch/arm26/kernel/setup.c

```

@@ -144,7 +144,7 @@ static void __init setup_processor(void)

```

```

    dump_cpu_info();

```

```

- sprintf(system_utsname.machine, "%s", list->arch_name);
+ sprintf(init_utsname()->machine, "%s", list->arch_name);
  sprintf(elf_platform, "%s", list->elf_name);
  elf_hwcap = list->elf_hwcap;

```

diff --git a/arch/cris/kernel/setup.c b/arch/cris/kernel/setup.c

index 619a6ee..1974c01 100644

```

--- a/arch/cris/kernel/setup.c
+++ b/arch/cris/kernel/setup.c
@@ -161,7 +161,7 @@ setup_arch(char **cmdline_p)
    show_etrax_copyright();

    /* Setup utsname */
- strcpy(system_utsname.machine, cris_machine_name);
+ strcpy(init_utsname()->machine, cris_machine_name);
}

static void *c_start(struct seq_file *m, loff_t *pos)
diff --git a/arch/i386/kernel/process.c b/arch/i386/kernel/process.c
index 6259afe..da2e439 100644
--- a/arch/i386/kernel/process.c
+++ b/arch/i386/kernel/process.c
@@ -297,9 +297,9 @@ void show_regs(struct pt_regs * regs)
    if (user_mode_vm(regs))
        printk(" ESP: %04x:%08lx", 0xffff & regs->xss, regs->esp);
    printk(" EFLAGS: %08lx  %s (%s %.*s)\n",
-       regs->eflags, print_tainted(), system_utsname.release,
-       (int)strcspn(system_utsname.version, " "),
-       system_utsname.version);
+       regs->eflags, print_tainted(), init_utsname()->release,
+       (int)strcspn(init_utsname()->version, " "),
+       init_utsname()->version);
    printk("EAX: %08lx EBX: %08lx ECX: %08lx EDX: %08lx\n",
        regs->eax, regs->ebx, regs->ecx, regs->edx);
    printk("ESI: %08lx EDI: %08lx EBP: %08lx",
diff --git a/arch/i386/kernel/traps.c b/arch/i386/kernel/traps.c
index e385279..dd62423 100644
--- a/arch/i386/kernel/traps.c
+++ b/arch/i386/kernel/traps.c
@@ -260,9 +260,9 @@ void show_registers(struct pt_regs *regs)
    printk(KERN_EMERG "CPU:  %d\nEIP:  %04x:[<%08lx>]  %s VLI\n"
        "EFLAGS: %08lx  (%s %.*s) \n",
        smp_processor_id(), 0xffff & regs->xcs, regs->eip,
-   print_tainted(), regs->eflags, system_utsname.release,
-   (int)strcspn(system_utsname.version, " "),
-   system_utsname.version);
+   print_tainted(), regs->eflags, init_utsname()->release,
+   (int)strcspn(init_utsname()->version, " "),
+   init_utsname()->version);
    print_symbol(KERN_EMERG "EIP is at %s\n", regs->eip);
    printk(KERN_EMERG "eax: %08lx ebx: %08lx ecx: %08lx edx: %08lx\n",
        regs->eax, regs->ebx, regs->ecx, regs->edx);
diff --git a/arch/powerpc/kernel/process.c b/arch/powerpc/kernel/process.c
index 2dd47d2..6ce9e10 100644
--- a/arch/powerpc/kernel/process.c

```

```

+++ b/arch/powerpc/kernel/process.c
@@ -425,7 +425,7 @@ void show_regs(struct pt_regs * regs)
    printk("NIP: \"REG\" LR: \"REG\" CTR: \"REG\"\\n",
           regs->nip, regs->link, regs->ctr);
    printk("REGS: %p TRAP: %04lx  %s  (%s)\\n",
-       regs, regs->trap, print_tainted(), system_utsname.release);
+       regs, regs->trap, print_tainted(), init_utsname()->release);
    printk("MSR: \"REG\" ", regs->msr);
    printbits(regs->msr, msr_bits);
    printk(" CR: %08IX XER: %08IX\\n", regs->ccr, regs->xer);
diff --git a/arch/powerpc/kernel/setup_64.c b/arch/powerpc/kernel/setup_64.c
index 13e91c4..1c6619f 100644
--- a/arch/powerpc/kernel/setup_64.c
+++ b/arch/powerpc/kernel/setup_64.c
@@ -435,7 +435,7 @@ void __init setup_system(void)
    smp_release_cpus();
#endif

-    printk("Starting Linux PPC64 %s\\n", system_utsname.version);
+    printk("Starting Linux PPC64 %s\\n", init_utsname()->version);

    printk("-----\\n ");
    printk("ppc64_pft_size          = 0x%lx\\n", ppc64_pft_size);
diff --git a/arch/powerpc/platforms/pseries/setup.c b/arch/powerpc/platforms/pseries/setup.c
index 5eb55ef..58b7a74 100644
--- a/arch/powerpc/platforms/pseries/setup.c
+++ b/arch/powerpc/platforms/pseries/setup.c
@@ -255,7 +255,7 @@ static int __init pSeries_init_panel(voi
{
    /* Manually leave the kernel version on the panel. */
    ppc_md.progress("Linux ppc64\\n", 0);
-    ppc_md.progress(system_utsname.version, 0);
+    ppc_md.progress(init_utsname()->version, 0);

    return 0;
}
diff --git a/arch/sh/kernel/setup.c b/arch/sh/kernel/setup.c
index bb229ef..024401e 100644
--- a/arch/sh/kernel/setup.c
+++ b/arch/sh/kernel/setup.c
@@ -481,7 +481,7 @@ static int show_cpuinfo(struct seq_file
    seq_printf(m, "machine\\t: %s\\n", get_system_type());

    seq_printf(m, "processor\\t: %d\\n", cpu);
-    seq_printf(m, "cpu family\\t: %s\\n", system_utsname.machine);
+    seq_printf(m, "cpu family\\t: %s\\n", init_utsname()->machine);
    seq_printf(m, "cpu type\\t: %s\\n", get_cpu_subtype());

```

```

    show_cpuflags(m);
diff --git a/arch/um/kernel/um_arch.c b/arch/um/kernel/um_arch.c
index 7d51dd7..b49dd7d 100644
--- a/arch/um/kernel/um_arch.c
+++ b/arch/um/kernel/um_arch.c
@@ -167,7 +167,7 @@ static char *usage_string =

static int __init uml_version_setup(char *line, int *add)
{
- printf("%s\n", system_utsname.release);
+ printf("%s\n", init_utsname()->release);
    exit(0);

    return 0;
@@ -278,7 +278,7 @@ static int __init Usage(char *line, int
{
    const char **p;

- printf(usage_string, system_utsname.release);
+ printf(usage_string, init_utsname()->release);
    p = &__uml_help_start;
    while (p < &__uml_help_end) {
        printf("%s", *p);
@@ -400,7 +400,7 @@ int linux_main(int argc, char **argv)
/* Reserve up to 4M after the current brk */
    uml_reserved = ROUND_4M(brk_start) + (1 << 22);

- setup_machinename(system_utsname.machine);
+ setup_machinename(init_utsname()->machine);

#ifdef CONFIG_CMDLINE_ON_HOST
    argv1_begin = argv[1];
diff --git a/arch/um/sys-x86_64/sysrq.c b/arch/um/sys-x86_64/sysrq.c
index d0a25af..ce3e07f 100644
--- a/arch/um/sys-x86_64/sysrq.c
+++ b/arch/um/sys-x86_64/sysrq.c
@@ -16,7 +16,7 @@ void __show_regs(struct pt_regs * regs)
    printk("\n");
    print_modules();
    printk("Pid: %d, comm: %.20s %s %s\n",
-        current->pid, current->comm, print_tainted(), system_utsname.release);
+        current->pid, current->comm, print_tainted(), init_utsname()->release);
    printk("RIP: %04lx:[<%016lx>] ", PT_REGS_CS(regs) & 0xffff,
        PT_REGS_RIP(regs));
    printk("\nRSP: %016lx EFLAGS: %08lx\n", PT_REGS_RSP(regs),
diff --git a/arch/x86_64/kernel/process.c b/arch/x86_64/kernel/process.c
index 70dd8e5..79f8174 100644
--- a/arch/x86_64/kernel/process.c

```



```

+++ b/arch/x86_64/kernel/process.c
@@ -292,9 +292,9 @@ void __show_regs(struct pt_regs * regs)
    print_modules();
    printk("Pid: %d, comm: %.20s %s %s %s\n",
        current->pid, current->comm, print_tainted(),
-   system_utsname.release,
-   (int)strncpy(system_utsname.version, " "),
-   system_utsname.version);
+   init_utsname()->release,
+   (int)strncpy(init_utsname()->version, " "),
+   init_utsname()->version);
    printk("RIP: %04lx:[<%016lx>] ", regs->cs & 0xffff, regs->rip);
    printk_address(regs->rip);
    printk("\nRSP: %04lx:%016lx  EFLAGS: %08lx\n", regs->ss, regs->rsp,
diff --git a/drivers/infiniband/hw/ipath/ipath_verbs.c b/drivers/infiniband/hw/ipath/ipath_verbs.c
index 9f27fd3..f873380 100644
--- a/drivers/infiniband/hw/ipath/ipath_verbs.c
+++ b/drivers/infiniband/hw/ipath/ipath_verbs.c
@@ -1048,7 +1048,7 @@ static void *ipath_register_ib_device(in
    dev->process_mad = ipath_process_mad;

    snprintf(dev->node_desc, sizeof(dev->node_desc),
-   IPATH_IDSTR " %s kernel_SMA", system_utsname.nodename);
+   IPATH_IDSTR " %s kernel_SMA", init_utsname()->nodename);

    ret = ib_register_device(dev);
    if (ret)
diff --git a/drivers/parisc/led.c b/drivers/parisc/led.c
index 298f2dd..1d778d2 100644
--- a/drivers/parisc/led.c
+++ b/drivers/parisc/led.c
@@ -684,7 +684,7 @@ int __init led_init(void)
    int ret;

    snprintf(lcd_text_default, sizeof(lcd_text_default),
-   "Linux %s", system_utsname.release);
+   "Linux %s", init_utsname()->release);

    /* Work around the buggy PDC of KittyHawk-machines */
    switch (CPU_HVERSION) {
diff --git a/drivers/scsi/lpfc/lpfc_ct.c b/drivers/scsi/lpfc/lpfc_ct.c
index b65ee57..83f53fb 100644
--- a/drivers/scsi/lpfc/lpfc_ct.c
+++ b/drivers/scsi/lpfc/lpfc_ct.c
@@ -961,8 +961,8 @@ lpfc_fdm_i_cmd(struct lpfc_hba * phba, st
    ae = (ATTRIBUTE_ENTRY *) ((uint8_t *) rh + size);
    ae->ad.bits.AttrType = be16_to_cpu(OS_NAME_VERSION);
    sprintf(ae->un.OsNameVersion, "%s %s %s",

```

```

- system_utsname.sysname, system_utsname.release,
- system_utsname.version);
+ init_utsname()->sysname, init_utsname()->release,
+ init_utsname()->version);
    len = strlen(ae->un.OsNameVersion);
    len += (len & 3) ? (4 - (len & 3)) : 4;
    ae->ad.bits.AttrLen = be16_to_cpu(FOURBYTES + len);
@@ -1080,7 +1080,7 @@ lpfc_fdmi_cmd(struct lpfc_hba * phba, st
    size);
    ae->ad.bits.AttrType = be16_to_cpu(HOST_NAME);
    sprintf(ae->un.HostName, "%s",
- system_utsname.nodename);
+ init_utsname()->nodename);
    len = strlen(ae->un.HostName);
    len += (len & 3) ? (4 - (len & 3)) : 4;
    ae->ad.bits.AttrLen =
@@ -1168,7 +1168,7 @@ lpfc_fdmi_tmo_handler(struct lpfc_hba *p

    ndlp = lpfc_findnode_did(phba, NLP_SEARCH_ALL, FDMI_DID);
    if (ndlp) {
- if (system_utsname.nodename[0] != '\0') {
+ if (init_utsname()->nodename[0] != '\0') {
    lpfc_fdmi_cmd(phba, ndlp, SLI_MGMT_DHBA);
    } else {
        mod_timer(&phba->fc_fdmittmo, jiffies + HZ * 60);
diff --git a/drivers/usb/core/hcd.c b/drivers/usb/core/hcd.c
index fbd938d..c1255ec 100644
--- a/drivers/usb/core/hcd.c
+++ b/drivers/usb/core/hcd.c
@@ -318,8 +318,8 @@ static int rh_string (

    // id 3 == vendor description
    } else if (id == 3) {
- snprintf (buf, sizeof buf, "%s %s %s", system_utsname.sysname,
- system_utsname.release, hcd->driver->description);
+ snprintf (buf, sizeof buf, "%s %s %s", init_utsname()->sysname,
+ init_utsname()->release, hcd->driver->description);

    // unsupported IDs --> "protocol stall"
    } else
diff --git a/drivers/usb/gadget/ether.c b/drivers/usb/gadget/ether.c
index c3d8e5c..e6fe999 100644
--- a/drivers/usb/gadget/ether.c
+++ b/drivers/usb/gadget/ether.c
@@ -2242,7 +2242,7 @@ eth_bind (struct usb_gadget *gadget)
    return -ENODEV;
}
    snprintf (manufacturer, sizeof manufacturer, "%s %s/%s",

```

```
- system_utsname.sysname, system_utsname.release,  
+ init_utsname()->sysname, init_utsname()->release,  
  gadget->name);
```

```
/* If there's an RNDIS configuration, that's what Windows wants to  
diff --git a/drivers/usb/gadget/file_storage.c b/drivers/usb/gadget/file_storage.c  
index cf3be29..d3149e5 100644
```

```
--- a/drivers/usb/gadget/file_storage.c  
+++ b/drivers/usb/gadget/file_storage.c  
@@ -3965,7 +3965,7 @@ static int __init fsg_bind(struct usb_ga  
  usb_gadget_set_selfpowered(gadget);
```

```
  snprintf(manufacturer, sizeof manufacturer, "%s %s with %s",  
- system_utsname.sysname, system_utsname.release,  
+ init_utsname()->sysname, init_utsname()->release,  
  gadget->name);
```

```
/* On a real device, serial[] would be loaded from permanent  
diff --git a/drivers/usb/gadget/serial.c b/drivers/usb/gadget/serial.c  
index b992546..a2f905b 100644  
--- a/drivers/usb/gadget/serial.c  
+++ b/drivers/usb/gadget/serial.c  
@@ -1496,7 +1496,7 @@ static int __init gs_bind(struct usb_gad  
  return -ENOMEM;
```

```
  snprintf(manufacturer, sizeof(manufacturer), "%s %s with %s",  
- system_utsname.sysname, system_utsname.release,  
+ init_utsname()->sysname, init_utsname()->release,  
  gadget->name);
```

```
  memset(dev, 0, sizeof(struct gs_dev));  
diff --git a/drivers/usb/gadget/zero.c b/drivers/usb/gadget/zero.c  
index 51424f6..b23e0fd 100644  
--- a/drivers/usb/gadget/zero.c  
+++ b/drivers/usb/gadget/zero.c  
@@ -1240,7 +1240,7 @@ autoconf_fail:  
  EP_OUT_NAME, EP_IN_NAME);
```

```
  snprintf (manufacturer, sizeof manufacturer, "%s %s with %s",  
- system_utsname.sysname, system_utsname.release,  
+ init_utsname()->sysname, init_utsname()->release,  
  gadget->name);
```

```
  return 0;  
diff --git a/include/asm-i386/bugs.h b/include/asm-i386/bugs.h  
index 50233e0..6cb79fe 100644  
--- a/include/asm-i386/bugs.h  
+++ b/include/asm-i386/bugs.h
```

```

@@ -190,6 +190,6 @@ static void __init check_bugs(void)
    check_fpu();
    check_hlt();
    check_popad();
- system_utsname.machine[1] = '0' + (boot_cpu_data.x86 > 6 ? 6 : boot_cpu_data.x86);
+ init_utsname()->machine[1] = '0' + (boot_cpu_data.x86 > 6 ? 6 : boot_cpu_data.x86);
    alternative_instructions();
}
diff --git a/include/asm-sh/bugs.h b/include/asm-sh/bugs.h
index a6de3d0..d09933c 100644
--- a/include/asm-sh/bugs.h
+++ b/include/asm-sh/bugs.h
@@ -18,7 +18,7 @@ static void __init check_bugs(void)
{
    extern char *get_cpu_subtype(void);
    extern unsigned long loops_per_jiffy;
- char *p= &system_utsname.machine[2]; /* "sh" */
+ char *p= &init_utsname()->machine[2]; /* "sh" */

    cpu_data->loops_per_jiffy = loops_per_jiffy;

diff --git a/kernel/power/snapshot.c b/kernel/power/snapshot.c
index c5863d0..f43f7db 100644
--- a/kernel/power/snapshot.c
+++ b/kernel/power/snapshot.c
@@ -523,7 +523,7 @@ static void init_header(struct swsusp_in
    memset(info, 0, sizeof(struct swsusp_info));
    info->version_code = LINUX_VERSION_CODE;
    info->num_physpages = num_physpages;
- memcpy(&info->uts, &system_utsname, sizeof(system_utsname));
+ memcpy(&info->uts, init_utsname(), sizeof(struct new_utsname));
    info->cpus = num_online_cpus();
    info->image_pages = nr_copy_pages;
    info->pages = nr_copy_pages + nr_meta_pages + 1;
@@ -662,13 +662,13 @@ static int check_header(struct swsusp_in
    reason = "kernel version";
    if (info->num_physpages != num_physpages)
        reason = "memory size";
- if (strcmp(info->uts.sysname, system_utsname.sysname))
+ if (strcmp(info->uts.sysname, init_utsname()->sysname))
        reason = "system type";
- if (strcmp(info->uts.release, system_utsname.release))
+ if (strcmp(info->uts.release, init_utsname()->release))
        reason = "kernel release";
- if (strcmp(info->uts.version, system_utsname.version))
+ if (strcmp(info->uts.version, init_utsname()->version))
        reason = "version";
- if (strcmp(info->uts.machine, system_utsname.machine))

```

```

+ if (strcmp(info->uts.machine,init_utsname()->machine))
    reason = "machine";
    if (reason) {
        printk(KERN_ERR "swsusp: Resume mismatch: %s\n", reason);
diff --git a/net/ipv4/ipconfig.c b/net/ipv4/ipconfig.c
index b9bdf0f..4c13acb 100644
--- a/net/ipv4/ipconfig.c
+++ b/net/ipv4/ipconfig.c
@@ -367,7 +367,7 @@ static int __init ic_defaults(void)
    */

    if (!lic_host_name_set)
-   sprintf(system_utsname.nodename, "%u.%u.%u.%u", NIPQUAD(ic_myaddr));
+   sprintf(init_utsname()->nodename, "%u.%u.%u.%u", NIPQUAD(ic_myaddr));

    if (root_server_addr == INADDR_NONE)
        root_server_addr = ic_servaddr;
diff --git a/sound/core/info_oss.c b/sound/core/info_oss.c
index f9ce854..35662bb 100644
--- a/sound/core/info_oss.c
+++ b/sound/core/info_oss.c
@@ -94,11 +94,11 @@ static void snd_sndstat_proc_read(struct
{
    snd_iprintf(buffer, "Sound Driver:3.8.1a-980706 (ALSA v" CONFIG_SND_VERSION " emulation
code)\n");
    snd_iprintf(buffer, "Kernel: %s %s %s %s %s\n",
-   system_utsname.sysname,
-   system_utsname.nodename,
-   system_utsname.release,
-   system_utsname.version,
-   system_utsname.machine);
+   init_utsname()->sysname,
+   init_utsname()->nodename,
+   init_utsname()->release,
+   init_utsname()->version,
+   init_utsname()->machine);
    snd_iprintf(buffer, "Config options: 0\n");
    snd_iprintf(buffer, "\nInstalled drivers: \n");
    snd_iprintf(buffer, "Type 10: ALSA emulation\n");
--
1.1.6

```

---