
Subject: [PATCH 2/7] uts namespaces: switch to using uts namespaces

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

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

Replace references to system_utsname to the per-process uts namespace where appropriate. This includes things like uname.

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

```
arch/alpha/kernel/osf_sys.c      | 24 ++++++-----  
arch/i386/kernel/sys_i386.c     | 12 ++++++----  
arch/ia64/sn/kernel/sn2/sn_hwperf.c | 2 +-  
arch/m32r/kernel/sys_m32r.c     | 2 +-  
arch/mips/kernel/linux32.c      | 2 +-  
arch/mips/kernel/syscall.c      | 18 ++++++-----  
arch/mips/kernel/sysirix.c      | 12 ++++++----  
arch/parisc/hpux/sys_hpux.c    | 22 ++++++-----  
arch/powerpc/kernel/syscalls.c | 14 ++++++----  
arch/sh/kernel/sys_sh.c        | 2 +-  
arch/sh64/kernel/sys_sh64.c     | 2 +-  
arch/sparc/kernel/sys_sparc.c   | 4 +---  
arch/sparc/kernel/sys_sunos.c   | 10 ++++++----  
arch/sparc64/kernel/sys_sparc.c | 4 +---  
arch/sparc64/kernel/sys_sunos32.c | 10 ++++++----  
arch/sparc64/solaris/misc.c    | 6 +----  
arch/um/drivers/mconsole_kern.c | 6 +----  
arch/um/kernel/syscall_kern.c   | 12 ++++++----  
arch/um/sys-x86_64/syscalls.c  | 2 +-  
arch/x86_64/ia32/sys_ia32.c    | 10 ++++++----  
arch/x86_64/kernel/sys_x86_64.c | 2 +-  
arch/xtensa/kernel/syscalls.c  | 2 +-  
drivers/char/random.c          | 4 +---  
fs/cifs/connect.c              | 28 ++++++-----  
fs/exec.c                      | 2 +-  
fs/lockd/clntproc.c            | 4 +---  
fs/lockd/mon.c                 | 2 +-  
fs/lockd/svclock.c             | 2 +-  
fs/lockd/xdr.c                 | 2 +-  
fs/nfs/nfsroot.c               | 2 +-  
include/linux/lockd/lockd.h     | 2 +-  
kernel/sys.c                   | 14 ++++++----  
32 files changed, 121 insertions(+), 121 deletions(-)
```

e98674a005449a65814b79a1e279c8192ee70445

diff --git a/arch/alpha/kernel/osf_sys.c b/arch/alpha/kernel/osf_sys.c

index 31afe3d..b793b96 100644

```

--- a/arch/alpha/kernel/osf_sys.c
+++ b/arch/alpha/kernel/osf_sys.c
@@ -402,15 +402,15 @@ osf_utsname(char __user *name)

    down_read(&uts_sem);
    error = -EFAULT;
- if (copy_to_user(name + 0, system_utsname.sysname, 32))
+ if (copy_to_user(name + 0, utsname()->sysname, 32))
    goto out;
- if (copy_to_user(name + 32, system_utsname.nodename, 32))
+ if (copy_to_user(name + 32, utsname()->nodename, 32))
    goto out;
- if (copy_to_user(name + 64, system_utsname.release, 32))
+ if (copy_to_user(name + 64, utsname()->release, 32))
    goto out;
- if (copy_to_user(name + 96, system_utsname.version, 32))
+ if (copy_to_user(name + 96, utsname()->version, 32))
    goto out;
- if (copy_to_user(name + 128, system_utsname.machine, 32))
+ if (copy_to_user(name + 128, utsname()->machine, 32))
    goto out;

    error = 0;
@@ -449,8 +449,8 @@ osf_getdomainname(char __user *name, int

    down_read(&uts_sem);
    for (i = 0; i < len; ++i) {
- __put_user(system_utsname.domainname[i], name + i);
- if (system_utsname.domainname[i] == '\0')
+ __put_user(utsname()->domainname[i], name + i);
+ if (utsname()->domainname[i] == '\0')
        break;
    }
    up_read(&uts_sem);
@@ -608,11 +608,11 @@ asmlinkage long
osf_sysinfo(int command, char __user *buf, long count)
{
    static char * sysinfo_table[] = {
- system_utsname.sysname,
- system_utsname.nodename,
- system_utsname.release,
- system_utsname.version,
- system_utsname.machine,
+ utsname()->sysname,
+ utsname()->nodename,
+ utsname()->release,
+ utsname()->version,
+ utsname()->machine,

```

```

"alpha", /* instruction set architecture */
"dummy", /* hardware serial number */
"dummy", /* hardware manufacturer */
diff --git a/arch/i386/kernel/sys_i386.c b/arch/i386/kernel/sys_i386.c
index 8fdb1fb..4af731d 100644
--- a/arch/i386/kernel/sys_i386.c
+++ b/arch/i386/kernel/sys_i386.c
@@ -210,7 +210,7 @@ asmlinkage int sys_uname(struct old_utsn
    if (!name)
        return -EFAULT;
    down_read(&uts_sem);
- err=copy_to_user(name, &system_utsname, sizeof (*name));
+ err=copy_to_user(name, utsname(), sizeof (*name));
    up_read(&uts_sem);
    return err?-EFAULT:0;
}
@@ -226,15 +226,15 @@ asmlinkage int sys_olduname(struct oldol

    down_read(&uts_sem);

- error = __copy_to_user(&name->sysname,&system_utsname.sysname,__OLD_UTS_LEN);
+ error = __copy_to_user(&name->sysname,&utsname()->sysname,__OLD_UTS_LEN);
    error |= __put_user(0,name->sysname+__OLD_UTS_LEN);
- error |= __copy_to_user(&name->nodename,&system_utsname.nodename,__OLD_UTS_LEN);
+ error |= __copy_to_user(&name->nodename,&utsname()->nodename,__OLD_UTS_LEN);
    error |= __put_user(0,name->nodename+__OLD_UTS_LEN);
- error |= __copy_to_user(&name->release,&system_utsname.release,__OLD_UTS_LEN);
+ error |= __copy_to_user(&name->release,&utsname()->release,__OLD_UTS_LEN);
    error |= __put_user(0,name->release+__OLD_UTS_LEN);
- error |= __copy_to_user(&name->version,&system_utsname.version,__OLD_UTS_LEN);
+ error |= __copy_to_user(&name->version,&utsname()->version,__OLD_UTS_LEN);
    error |= __put_user(0,name->version+__OLD_UTS_LEN);
- error |= __copy_to_user(&name->machine,&system_utsname.machine,__OLD_UTS_LEN);
+ error |= __copy_to_user(&name->machine,&utsname()->machine,__OLD_UTS_LEN);
    error |= __put_user(0,name->machine+__OLD_UTS_LEN);

    up_read(&uts_sem);
diff --git a/arch/ia64/sn/kernel/sn2/sn_hwperf.c b/arch/ia64/sn/kernel/sn2/sn_hwperf.c
index d917afa..a0632a9 100644
--- a/arch/ia64/sn/kernel/sn2/sn_hwperf.c
+++ b/arch/ia64/sn/kernel/sn2/sn_hwperf.c
@@ -420,7 +420,7 @@ static int sn_topology_show(struct seq_f
    "coherency_domain %d,\n"
    "region_size %d\n",
    "- partid, system_utsname.nodename,
+ partid, utsname()->nodename,
    shubtype ? "shub2" : "shub1",

```

```

(u64)nasid_mask << nasid_shift, nasid_msb, nasid_shift,
    system_size, sharing_size, coher, region_size);
diff --git a/arch/m32r/kernel/sys_m32r.c b/arch/m32r/kernel/sys_m32r.c
index 670cb49..11412c0 100644
--- a/arch/m32r/kernel/sys_m32r.c
+++ b/arch/m32r/kernel/sys_m32r.c
@@ -206,7 +206,7 @@ asmlinkage int sys_uname(struct old_utsn
if (!name)
    return -EFAULT;
down_read(&uts_sem);
- err=copy_to_user(name, &system_utsname, sizeof (*name));
+ err=copy_to_user(name, utsname(), sizeof (*name));
up_read(&uts_sem);
return err?-EFAULT:0;
}
diff --git a/arch/mips/kernel/linux32.c b/arch/mips/kernel/linux32.c
index 3f40c37..b9b702f 100644
--- a/arch/mips/kernel/linux32.c
+++ b/arch/mips/kernel/linux32.c
@@ -1100,7 +1100,7 @@ asmlinkage long sys32_newuname(struct ne
int ret = 0;

down_read(&uts_sem);
- if (copy_to_user(name,&system_utsname,sizeof *name))
+ if (copy_to_user(name,utsname(),sizeof *name))
    ret = -EFAULT;
up_read(&uts_sem);

diff --git a/arch/mips/kernel/syscall.c b/arch/mips/kernel/syscall.c
index 2aeaa2f..8b13d57 100644
--- a/arch/mips/kernel/syscall.c
+++ b/arch/mips/kernel/syscall.c
@@ -232,7 +232,7 @@ out:
 */
asmlinkage int sys_uname(struct old_utsname __user * name)
{
- if (name && !copy_to_user(name, &system_utsname, sizeof (*name)))
+ if (name && !copy_to_user(name, utsname(), sizeof (*name)))
    return 0;
return -EFAULT;
}
@@ -249,15 +249,15 @@ asmlinkage int sys_olduname(struct oldol
if (!access_ok(VVERIFY_WRITE,name,sizeof(struct oldold_utsname)))
    return -EFAULT;

- error = __copy_to_user(&name->sysname,&system_utsname.sysname,__OLD_UTS_LEN);
+ error = __copy_to_user(&name->sysname,&utsname()->sysname,__OLD_UTS_LEN);
error -= __put_user(0,name->sysname+__OLD_UTS_LEN);

```

```

- error -= __copy_to_user(&name->nodename,&system_utsname.nodename,__OLD_UTS_LEN);
+ error -= __copy_to_user(&name->nodename,&utsname()->nodename,__OLD_UTS_LEN);
    error -= __put_user(0,name->nodename+__OLD_UTS_LEN);
- error -= __copy_to_user(&name->release,&system_utsname.release,__OLD_UTS_LEN);
+ error -= __copy_to_user(&name->release,&utsname()->release,__OLD_UTS_LEN);
    error -= __put_user(0,name->release+__OLD_UTS_LEN);
- error -= __copy_to_user(&name->version,&system_utsname.version,__OLD_UTS_LEN);
+ error -= __copy_to_user(&name->version,&utsname()->version,__OLD_UTS_LEN);
    error -= __put_user(0,name->version+__OLD_UTS_LEN);
- error -= __copy_to_user(&name->machine,&system_utsname.machine,__OLD_UTS_LEN);
+ error -= __copy_to_user(&name->machine,&utsname()->machine,__OLD_UTS_LEN);
    error = __put_user(0,name->machine+__OLD_UTS_LEN);
error = error ? -EFAULT : 0;

@@ -293,10 +293,10 @@ asmlinkage int _sys_sysmips(int cmd, ion
    return -EFAULT;

    down_write(&uts_sem);
- strncpy(system_utsname.nodename, nodename, len);
+ strncpy(utsname()->nodename, nodename, len);
    nodename[__NEW_UTS_LEN] = '\0';
- strlcpy(system_utsname.nodename, nodename,
-         sizeof(system_utsname.nodename));
+ strlcpy(utsname()->nodename, nodename,
+         sizeof(utsname()->nodename));
    up_write(&uts_sem);
    return 0;
}
diff --git a/arch/mips/kernel/sysirix.c b/arch/mips/kernel/sysirix.c
index 5407b78..1b4e7e7 100644
--- a/arch/mips/kernel/sysirix.c
+++ b/arch/mips/kernel/sysirix.c
@@ -884,7 +884,7 @@ asmlinkage int irix_getdomainname(char _
    down_read(&uts_sem);
    if (len > __NEW_UTS_LEN)
        len = __NEW_UTS_LEN;
- err = copy_to_user(name, system_utsname.domainname, len) ? -EFAULT : 0;
+ err = copy_to_user(name, utsname()->domainname, len) ? -EFAULT : 0;
    up_read(&uts_sem);

    return err;
@@ -1127,11 +1127,11 @@ struct iuname {
asmlinkage int irix_uname(struct iuname __user *buf)
{
    down_read(&uts_sem);
- if (copy_from_user(system_utsname.sysname, buf->sysname, 65)
-     || copy_from_user(system_utsname.nodename, buf->nodename, 65)
-     || copy_from_user(system_utsname.release, buf->release, 65)

```

```

-    || copy_from_user(system_utsname.version, buf->version, 65)
-    || copy_from_user(system_utsname.machine, buf->machine, 65)) {
+ if (copy_from_user(utsname()->sysname, buf->sysname, 65)
+    || copy_from_user(utsname()->nodename, buf->nodename, 65)
+    || copy_from_user(utsname()->release, buf->release, 65)
+    || copy_from_user(utsname()->version, buf->version, 65)
+    || copy_from_user(utsname()->machine, buf->machine, 65)) {
    return -EFAULT;
}
up_read(&uts_sem);
diff --git a/arch/parisc/hpux/sys_hpux.c b/arch/parisc/hpux/sys_hpux.c
index 05273cc..9fc2c08 100644
--- a/arch/parisc/hpux/sys_hpux.c
+++ b/arch/parisc/hpux/sys_hpux.c
@@ -266,15 +266,15 @@ static int hpx uname(struct hpx_utsnam

down_read(&uts_sem);

- error = __copy_to_user(&name->sysname,&system_utsname.sysname,HPUX_UTSLEN-1);
+ error = __copy_to_user(&name->sysname,&utsname()->sysname,HPUX_UTSLEN-1);
error |= __put_user(0,name->sysname+HPUX_UTSLEN-1);
- error |= __copy_to_user(&name->nodename,&system_utsname.nodename,HPUX_UTSLEN-1);
+ error |= __copy_to_user(&name->nodename,&utsname()->nodename,HPUX_UTSLEN-1);
error |= __put_user(0,name->nodename+HPUX_UTSLEN-1);
- error |= __copy_to_user(&name->release,&system_utsname.release,HPUX_UTSLEN-1);
+ error |= __copy_to_user(&name->release,&utsname()->release,HPUX_UTSLEN-1);
error |= __put_user(0,name->release+HPUX_UTSLEN-1);
- error |= __copy_to_user(&name->version,&system_utsname.version,HPUX_UTSLEN-1);
+ error |= __copy_to_user(&name->version,&utsname()->version,HPUX_UTSLEN-1);
error |= __put_user(0,name->version+HPUX_UTSLEN-1);
- error |= __copy_to_user(&name->machine,&system_utsname.machine,HPUX_UTSLEN-1);
+ error |= __copy_to_user(&name->machine,&utsname()->machine,HPUX_UTSLEN-1);
error |= __put_user(0,name->machine+HPUX_UTSLEN-1);

up_read(&uts_sem);
@@ -373,8 +373,8 @@ int hpx_utssys(char *ubuf, int n, int t
/* TODO: print a warning about using this? */
down_write(&uts_sem);
error = -EFAULT;
- if (!copy_from_user(system_utsname.sysname, ubuf, len)) {
- system_utsname.sysname[len] = 0;
+ if (!copy_from_user(utsname()->sysname, ubuf, len)) {
+ utsname()->sysname[len] = 0;
error = 0;
}
up_write(&uts_sem);
@@ -400,8 +400,8 @@ int hpx_utssys(char *ubuf, int n, int t
/* TODO: print a warning about this? */

```

```

down_write(&uts_sem);
error = -EFAULT;
- if (!copy_from_user(system_utsname.release, ubuf, len)) {
- system_utsname.release[len] = 0;
+ if (!copy_from_user(utsname()->release, ubuf, len)) {
+ utsname()->release[len] = 0;
error = 0;
}
up_write(&uts_sem);
@@ -422,13 +422,13 @@ int hpx_getdomainname(char *name, int l

down_read(&uts_sem);

- nlen = strlen(system_utsname.domainname) + 1;
+ nlen = strlen(utsname()->domainname) + 1;

if (nlen < len)
len = nlen;
if(len > __NEW_UTS_LEN)
goto done;
- if(copy_to_user(name, system_utsname.domainname, len))
+ if(copy_to_user(name, utsname()->domainname, len))
goto done;
err = 0;
done:
diff --git a/arch/powerpc/kernel/syscalls.c b/arch/powerpc/kernel/syscalls.c
index 9b69d99..d358866 100644
--- a/arch/powerpc/kernel/syscalls.c
+++ b/arch/powerpc/kernel/syscalls.c
@@ -260,7 +260,7 @@ long ppc_newuname(struct new_utsname __u
int err = 0;

down_read(&uts_sem);
- if (copy_to_user(name, &system_utsname, sizeof(*name)))
+ if (copy_to_user(name, utsname(), sizeof(*name)))
err = -EFAULT;
up_read(&uts_sem);
if (!err)
@@ -273,7 +273,7 @@ int sys_uname(struct old_utsname __user
int err = 0;

down_read(&uts_sem);
- if (copy_to_user(name, &system_utsname, sizeof(*name)))
+ if (copy_to_user(name, utsname(), sizeof(*name)))
err = -EFAULT;
up_read(&uts_sem);
if (!err)
@@ -289,19 +289,19 @@ int sys_olduname(struct oldold_utsname __

```

```

return -EFAULT;

down_read(&uts_sem);
- error = __copy_to_user(&name->sysname, &system_utsname.sysname,
+ error = __copy_to_user(&name->sysname, &utsname()->sysname,
    __OLD_UTS_LEN);
error |= __put_user(0, name->sysname + __OLD_UTS_LEN);
- error |= __copy_to_user(&name->nodename, &system_utsname.nodename,
+ error |= __copy_to_user(&name->nodename, &utsname()->nodename,
    __OLD_UTS_LEN);
error |= __put_user(0, name->nodename + __OLD_UTS_LEN);
- error |= __copy_to_user(&name->release, &system_utsname.release,
+ error |= __copy_to_user(&name->release, &utsname()->release,
    __OLD_UTS_LEN);
error |= __put_user(0, name->release + __OLD_UTS_LEN);
- error |= __copy_to_user(&name->version, &system_utsname.version,
+ error |= __copy_to_user(&name->version, &utsname()->version,
    __OLD_UTS_LEN);
error |= __put_user(0, name->version + __OLD_UTS_LEN);
- error |= __copy_to_user(&name->machine, &system_utsname.machine,
+ error |= __copy_to_user(&name->machine, &utsname()->machine,
    __OLD_UTS_LEN);
error |= override_machine(name->machine);
up_read(&uts_sem);
diff --git a/arch/sh/kernel/sys_sh.c b/arch/sh/kernel/sys_sh.c
index 917b2f3..e4966b2 100644
--- a/arch/sh/kernel/sys_sh.c
+++ b/arch/sh/kernel/sys_sh.c
@@ @ -267,7 +267,7 @@ asmlinkage int sys_uname(struct old_utsn
if (!name)
    return -EFAULT;
down_read(&uts_sem);
- err=copy_to_user(name, &system_utsname, sizeof (*name));
+ err=copy_to_user(name, utsname(), sizeof (*name));
up_read(&uts_sem);
return err?-EFAULT:0;
}
diff --git a/arch/sh64/kernel/sys_sh64.c b/arch/sh64/kernel/sys_sh64.c
index 58ff7d5..a8dc88c 100644
--- a/arch/sh64/kernel/sys_sh64.c
+++ b/arch/sh64/kernel/sys_sh64.c
@@ @ -279,7 +279,7 @@ asmlinkage int sys_uname(struct old_utsn
if (!name)
    return -EFAULT;
down_read(&uts_sem);
- err=copy_to_user(name, &system_utsname, sizeof (*name));
+ err=copy_to_user(name, utsname(), sizeof (*name));
up_read(&uts_sem);

```

```

    return err?-EFAULT:0;
}
diff --git a/arch/sparc/kernel/sys_sparc.c b/arch/sparc/kernel/sys_sparc.c
index 0cdfc9d..c8ad73c 100644
--- a/arch/sparc/kernel/sys_sparc.c
+++ b/arch/sparc/kernel/sys_sparc.c
@@ -470,13 +470,13 @@ asmlinkage int sys_getdomainname(char __

    down_read(&uts_sem);

- nlen = strlen(system_utsname.domainname) + 1;
+ nlen = strlen(utsname()->domainname) + 1;

if (nlen < len)
    len = nlen;
if (len > __NEW_UTS_LEN)
    goto done;
- if (copy_to_user(name, system_utsname.domainname, len))
+ if (copy_to_user(name, utsname()->domainname, len))
    goto done;
err = 0;
done:
diff --git a/arch/sparc/kernel/sys_sunos.c b/arch/sparc/kernel/sys_sunos.c
index 288de27..9f9206f 100644
--- a/arch/sparc/kernel/sys_sunos.c
+++ b/arch/sparc/kernel/sys_sunos.c
@@ -483,13 +483,13 @@ asmlinkage int sunos_uname(struct sunos_{
{
    int ret;
    down_read(&uts_sem);
- ret = copy_to_user(&name->sname[0], &system_utsname.sysname[0], sizeof(name->sname) - 1);
+ ret = copy_to_user(&name->sname[0], &utsname()->sysname[0], sizeof(name->sname) - 1);
    if (!ret) {
- ret |= __copy_to_user(&name->nname[0], &system_utsname.nodename[0], sizeof(name->nname) - 1);
+ ret |= __copy_to_user(&name->nname[0], &utsname()->nodename[0], sizeof(name->nname) - 1);
    ret |= __put_user('\0', &name->nname[8]);
- ret |= __copy_to_user(&name->rel[0], &system_utsname.release[0], sizeof(name->rel) - 1);
- ret |= __copy_to_user(&name->ver[0], &system_utsname.version[0], sizeof(name->ver) - 1);
- ret |= __copy_to_user(&name->mach[0], &system_utsname.machine[0], sizeof(name->mach) - 1);
+ ret |= __copy_to_user(&name->rel[0], &utsname()->release[0], sizeof(name->rel) - 1);
+ ret |= __copy_to_user(&name->ver[0], &utsname()->version[0], sizeof(name->ver) - 1);
+ ret |= __copy_to_user(&name->mach[0], &utsname()->machine[0], sizeof(name->mach) - 1);
    }
    up_read(&uts_sem);

```

```

return ret ? -EFAULT : 0;
diff --git a/arch/sparc64/kernel/sys_sparc.c b/arch/sparc64/kernel/sys_sparc.c
index 7a86913..0453bd2 100644
--- a/arch/sparc64/kernel/sys_sparc.c
+++ b/arch/sparc64/kernel/sys_sparc.c
@@ -707,13 +707,13 @@ asmlinkage long sys_getdomainname(char __
down_read(&uts_sem);

-nlen = strlen(system_utsname.domainname) + 1;
+nlen = strlen(utsname()->domainname) + 1;

    if (nlen < len)
        len = nlen;
    if (len > __NEW_UTS_LEN)
        goto done;
-    if (copy_to_user(name, system_utsname.domainname, len))
+    if (copy_to_user(name, utsname()->domainname, len))
        goto done;
    err = 0;
done:
diff --git a/arch/sparc64/kernel/sys_sunos32.c b/arch/sparc64/kernel/sys_sunos32.c
index ae5b32f..ba98c47 100644
--- a/arch/sparc64/kernel/sys_sunos32.c
+++ b/arch/sparc64/kernel/sys_sunos32.c
@@ -439,16 +439,16 @@ asmlinkage int sunos_uname(struct sunos_
int ret;

down_read(&uts_sem);
-ret = copy_to_user(&name->sname[0], &system_utsname.sysname[0],
+ret = copy_to_user(&name->sname[0], &utsname()->sysname[0],
    sizeof(name->sname) - 1);
-ret |= copy_to_user(&name->nname[0], &system_utsname.nodename[0],
+ret |= copy_to_user(&name->nname[0], &utsname()->nodename[0],
    sizeof(name->nname) - 1);
ret |= put_user('\0', &name->nname[8]);
-ret |= copy_to_user(&name->rel[0], &system_utsname.release[0],
+ret |= copy_to_user(&name->rel[0], &utsname()->release[0],
    sizeof(name->rel) - 1);
-ret |= copy_to_user(&name->ver[0], &system_utsname.version[0],
+ret |= copy_to_user(&name->ver[0], &utsname()->version[0],
    sizeof(name->ver) - 1);
-ret |= copy_to_user(&name->mach[0], &system_utsname.machine[0],
+ret |= copy_to_user(&name->mach[0], &utsname()->machine[0],
    sizeof(name->mach) - 1);
up_read(&uts_sem);
return (ret ? -EFAULT : 0);
diff --git a/arch/sparc64/solaris/misc.c b/arch/sparc64/solaris/misc.c

```

```

index 5284996..5d0162a 100644
--- a/arch/sparc64/solaris/misc.c
+++ b/arch/sparc64/solaris/misc.c
@@ -239,7 +239,7 @@ asmlinkage int solaris_utssys(u32 buf, u
 /* Let's cheat */
 err = set_utsfield(v->sysname, "SunOS", 1, 0);
 down_read(&uts_sem);
- err |= set_utsfield(v->nodename, system_utsname.nodename,
+ err |= set_utsfield(v->nodename, utsname()->nodename,
    1, 1);
 up_read(&uts_sem);
 err |= set_utsfield(v->release, "2.6", 0, 0);
@@ -263,7 +263,7 @@ asmlinkage int solaris_utsname(u32 buf)
 /* Why should we not lie a bit? */
 down_read(&uts_sem);
 err = set_utsfield(v->sysname, "SunOS", 0, 0);
- err |= set_utsfield(v->nodename, system_utsname.nodename, 1, 1);
+ err |= set_utsfield(v->nodename, utsname()->nodename, 1, 1);
 err |= set_utsfield(v->release, "5.6", 0, 0);
 err |= set_utsfield(v->version, "Generic", 0, 0);
 err |= set_utsfield(v->machine, machine(), 0, 0);
@@ -295,7 +295,7 @@ asmlinkage int solaris_sysinfo(int cmd,
 case SI_HOSTNAME:
 r = buffer + 256;
 down_read(&uts_sem);
- for (p = system_utsname.nodename, q = buffer;
+ for (p = utsname()->nodename, q = buffer;
      q < r && *p && *p != ':'; *q++ = *p++);
 up_read(&uts_sem);
 *q = 0;
diff --git a/arch/um/drivers/mconsole_kern.c b/arch/um/drivers/mconsole_kern.c
index 28e3760..5f87323 100644
--- a/arch/um/drivers/mconsole_kern.c
+++ b/arch/um/drivers/mconsole_kern.c
@@ -106,9 +106,9 @@ void mconsole_version(struct mc_request
{
    char version[256];

- sprintf(version, "%s %s %s %s %s", system_utsname.sysname,
- system_utsname.nodename, system_utsname.release,
- system_utsname.version, system_utsname.machine);
+ sprintf(version, "%s %s %s %s %s", utsname()->sysname,
+ utsname()->nodename, utsname()->release,
+ utsname()->version, utsname()->machine);
    mconsole_reply(req, version, 0, 0);
}

diff --git a/arch/um/kernel/syscall_kern.c b/arch/um/kernel/syscall_kern.c

```

```

index 37d3978..d90e9ed 100644
--- a/arch/um/kernel/syscall_kern.c
+++ b/arch/um/kernel/syscall_kern.c
@@ -110,7 +110,7 @@ long sys_uname(struct old_utsname __user
if (!name)
    return -EFAULT;
down_read(&uts_sem);
- err=copy_to_user(name, &system_utsname, sizeof (*name));
+ err=copy_to_user(name, utsname(), sizeof (*name));
up_read(&uts_sem);
return err?-EFAULT:0;
}
@@ -126,19 +126,19 @@ long sys_oldsuname(struct oldold_utsname
down_read(&uts_sem);

- error = __copy_to_user(&name->sysname,&system_utsname.sysname,
+ error = __copy_to_user(&name->sysname,&utsname()->sysname,
    __OLD_UTS_LEN);
error |= __put_user(0,name->sysname+__OLD_UTS_LEN);
- error |= __copy_to_user(&name->nodename,&system_utsname.nodename,
+ error |= __copy_to_user(&name->nodename,&utsname()->nodename,
    __OLD_UTS_LEN);
error |= __put_user(0,name->nodename+__OLD_UTS_LEN);
- error |= __copy_to_user(&name->release,&system_utsname.release,
+ error |= __copy_to_user(&name->release,&utsname()->release,
    __OLD_UTS_LEN);
error |= __put_user(0,name->release+__OLD_UTS_LEN);
- error |= __copy_to_user(&name->version,&system_utsname.version,
+ error |= __copy_to_user(&name->version,&utsname()->version,
    __OLD_UTS_LEN);
error |= __put_user(0,name->version+__OLD_UTS_LEN);
- error |= __copy_to_user(&name->machine,&system_utsname.machine,
+ error |= __copy_to_user(&name->machine,&utsname()->machine,
    __OLD_UTS_LEN);
error |= __put_user(0,name->machine+__OLD_UTS_LEN);

```

```

diff --git a/arch/um/sys-x86_64/syscalls.c b/arch/um/sys-x86_64/syscalls.c
index 6acee5c..3ad014e 100644
--- a/arch/um/sys-x86_64/syscalls.c
+++ b/arch/um/sys-x86_64/syscalls.c
@@ -21,7 +21,7 @@ asmlinkage long sys_uname64(struct new_u
{
int err;
down_read(&uts_sem);
- err = copy_to_user(name, &system_utsname, sizeof (*name));
+ err = copy_to_user(name, utsname(), sizeof (*name));
up_read(&uts_sem);

```

```

if (personality(current->personality) == PER_LINUX32)
    err |= copy_to_user(&name->machine, "i686", 5);
diff --git a/arch/x86_64/ia32/sys_ia32.c b/arch/x86_64/ia32/sys_ia32.c
index f182b20..6e0a19d 100644
--- a/arch/x86_64/ia32/sys_ia32.c
+++ b/arch/x86_64/ia32/sys_ia32.c
@@ -801,13 +801,13 @@ asmlinkage long sys32_olduname(struct ol

    down_read(&uts_sem);

- error = __copy_to_user(&name->sysname,&system_utsname.sysname,__OLD_UTS_LEN);
+ error = __copy_to_user(&name->sysname,&utsname()->sysname,__OLD_UTS_LEN);
    __put_user(0,name->sysname+__OLD_UTS_LEN);
- __copy_to_user(&name->nodename,&system_utsname.nodename,__OLD_UTS_LEN);
+ __copy_to_user(&name->nodename,&utsname()->nodename,__OLD_UTS_LEN);
    __put_user(0,name->nodename+__OLD_UTS_LEN);
- __copy_to_user(&name->release,&system_utsname.release,__OLD_UTS_LEN);
+ __copy_to_user(&name->release,&utsname()->release,__OLD_UTS_LEN);
    __put_user(0,name->release+__OLD_UTS_LEN);
- __copy_to_user(&name->version,&system_utsname.version,__OLD_UTS_LEN);
+ __copy_to_user(&name->version,&utsname()->version,__OLD_UTS_LEN);
    __put_user(0,name->version+__OLD_UTS_LEN);
{
    char *arch = "x86_64";
@@ -830,7 +830,7 @@ long sys32_uname(struct old_utsname __us
    if (!name)
        return -EFAULT;
    down_read(&uts_sem);
- err=copy_to_user(name, &system_utsname, sizeof (*name));
+ err=copy_to_user(name, utsname(), sizeof (*name));
    up_read(&uts_sem);
    if (personality(current->personality) == PER_LINUX32)
        err |= copy_to_user(&name->machine, "i686", 5);
diff --git a/arch/x86_64/kernel/sys_x86_64.c b/arch/x86_64/kernel/sys_x86_64.c
index 6449ea8..76bf7c2 100644
--- a/arch/x86_64/kernel/sys_x86_64.c
+++ b/arch/x86_64/kernel/sys_x86_64.c
@@ -148,7 +148,7 @@ asmlinkage long sys_uname(struct new_uts
{
    int err;
    down_read(&uts_sem);
- err = copy_to_user(name, &system_utsname, sizeof (*name));
+ err = copy_to_user(name, utsname(), sizeof (*name));
    up_read(&uts_sem);
    if (personality(current->personality) == PER_LINUX32)
        err |= copy_to_user(&name->machine, "i686", 5);
diff --git a/arch/xtensa/kernel/syscalls.c b/arch/xtensa/kernel/syscalls.c
index f20c649..30060c1 100644

```

```

--- a/arch/xtensa/kernel/syscalls.c
+++ b/arch/xtensa/kernel/syscalls.c
@@ -129,7 +129,7 @@ out:

int sys_utsname(struct old_utsname * name)
{
- if (name && !copy_to_user(name, &system_utsname, sizeof (*name)))
+ if (name && !copy_to_user(name, utsname(), sizeof (*name)))
    return 0;
 return -EFAULT;
}

diff --git a/drivers/char/random.c b/drivers/char/random.c
index 86be04b..ec4c11d 100644
--- a/drivers/char/random.c
+++ b/drivers/char/random.c
@@ -888,8 +888,8 @@ static void init_std_data(struct entropy

    do_gettimeofday(&tv);
    add_entropy_words(r, (__u32 *)&tv, sizeof(tv)/4);
- add_entropy_words(r, (__u32 *)&system_utsname,
-     sizeof(system_utsname)/4);
+ add_entropy_words(r, (__u32 *)utsname(),
+     sizeof(*utsname())/4);
}

static int __init rand_initialize(void)
diff --git a/fs/cifs/connect.c b/fs/cifs/connect.c
index 0b86d5c..852ff41 100644
--- a/fs/cifs/connect.c
+++ b/fs/cifs/connect.c
@@ -765,12 +765,12 @@ cifs_parse_mount_options(char *options,
    separator[1] = 0;

    memset(vol->source_rfc1001_name,0x20,15);
- for(i=0;i < strnlen(system_utsname.nodename,15);i++) {
+ for(i=0;i < strnlen(utsname()->nodename,15);i++) {
        /* does not have to be a perfect mapping since the field is
         informational, only used for servers that do not support
         port 445 and it can be overridden at mount time */
        vol->source_rfc1001_name[i] =
- toupper(system_utsname.nodename[i]);
+ toupper(utsname()->nodename[i]);
    }
    vol->source_rfc1001_name[15] = 0;
    /* null target name indicates to use *SMBSSERVER default called name
@@ -2077,7 +2077,7 @@ CIFSSessSetup(unsigned int xid, struct c
    32, nls_codepage);
    bcc_ptr += 2 * bytes_returned;

```

```

bytes_returned =
-    cifs_strtoUCS((__le16 *) bcc_ptr, system_utsname.release,
+    cifs_strtoUCS((__le16 *) bcc_ptr, utsname()->release,
     32, nls_codepage);
bcc_ptr += 2 * bytes_returned;
bcc_ptr += 2;
@@ -2104,8 +2104,8 @@ CIFSSessSetup(unsigned int xid, struct c
}
strcpy(bcc_ptr, "Linux version ");
bcc_ptr += strlen("Linux version ");
- strcpy(bcc_ptr, system_utsname.release);
- bcc_ptr += strlen(system_utsname.release) + 1;
+ strcpy(bcc_ptr, utsname()->release);
+ bcc_ptr += strlen(utsname()->release) + 1;
strcpy(bcc_ptr, CIFS_NETWORK_OPSYS);
bcc_ptr += strlen(CIFS_NETWORK_OPSYS) + 1;
}
@@ -2346,7 +2346,7 @@ CIFSSnegoSessSetup(unsigned int xid, st
32, nls_codepage);
bcc_ptr += 2 * bytes_returned;
bytes_returned =
-    cifs_strtoUCS((__le16 *) bcc_ptr, system_utsname.release, 32,
+    cifs_strtoUCS((__le16 *) bcc_ptr, utsname()->release, 32,
     nls_codepage);
bcc_ptr += 2 * bytes_returned;
bcc_ptr += 2;
@@ -2371,8 +2371,8 @@ CIFSSnegoSessSetup(unsigned int xid, st
}
strcpy(bcc_ptr, "Linux version ");
bcc_ptr += strlen("Linux version ");
- strcpy(bcc_ptr, system_utsname.release);
- bcc_ptr += strlen(system_utsname.release) + 1;
+ strcpy(bcc_ptr, utsname()->release);
+ bcc_ptr += strlen(utsname()->release) + 1;
strcpy(bcc_ptr, CIFS_NETWORK_OPSYS);
bcc_ptr += strlen(CIFS_NETWORK_OPSYS) + 1;
}
@@ -2622,7 +2622,7 @@ CIFSNTLMSSPNegotiateSessSetup(unsigned i
32, nls_codepage);
bcc_ptr += 2 * bytes_returned;
bytes_returned =
-    cifs_strtoUCS((__le16 *) bcc_ptr, system_utsname.release, 32,
+    cifs_strtoUCS((__le16 *) bcc_ptr, utsname()->release, 32,
     nls_codepage);
bcc_ptr += 2 * bytes_returned;
bcc_ptr += 2; /* null terminate Linux version */
@@ -2639,8 +2639,8 @@ CIFSNTLMSSPNegotiateSessSetup(unsigned i
} else { /* ASCII */

```

```

strcpy(bcc_ptr, "Linux version ");
bcc_ptr += strlen("Linux version ");
- strcpy(bcc_ptr, system_utsname.release);
- bcc_ptr += strlen(system_utsname.release) + 1;
+ strcpy(bcc_ptr, utsname()->release);
+ bcc_ptr += strlen(utsname()->release) + 1;
strcpy(bcc_ptr, CIFS_NETWORK_OPSYS);
bcc_ptr += strlen(CIFS_NETWORK_OPSYS) + 1;
bcc_ptr++; /* empty domain field */
@@ -3001,7 +3001,7 @@ @@ CIFSNTLMSSPAuthSessSetup(unsigned int xi
    32, nls_codepage);
bcc_ptr += 2 * bytes_returned;
bytes_returned =
-   cifs_strtoUCS((__le16 *) bcc_ptr, system_utsname.release, 32,
+   cifs_strtoUCS((__le16 *) bcc_ptr, utsname()->release, 32,
      nls_codepage);
bcc_ptr += 2 * bytes_returned;
bcc_ptr += 2; /* null term version string */
@@ -3053,8 +3053,8 @@ @@ CIFSNTLMSSPAuthSessSetup(unsigned int xi

strcpy(bcc_ptr, "Linux version ");
bcc_ptr += strlen("Linux version ");
- strcpy(bcc_ptr, system_utsname.release);
- bcc_ptr += strlen(system_utsname.release) + 1;
+ strcpy(bcc_ptr, utsname()->release);
+ bcc_ptr += strlen(utsname()->release) + 1;
strcpy(bcc_ptr, CIFS_NETWORK_OPSYS);
bcc_ptr += strlen(CIFS_NETWORK_OPSYS) + 1;
bcc_ptr++; /* null domain */
diff --git a/fs/exec.c b/fs/exec.c
index 0291a68..d881479 100644
--- a/fs/exec.c
+++ b/fs/exec.c
@@ -1347,7 +1347,7 @@ static void format_corename(char *corename
    case 'h':
        down_read(&uts_sem);
        rc = snprintf(out_ptr, out_end - out_ptr,
-          "%s", system_utsname.nodename);
+          "%s", utsname()->nodename);
        up_read(&uts_sem);
        if (rc > out_end - out_ptr)
            goto out;
diff --git a/fs/lockd/clntproc.c b/fs/lockd/clntproc.c
index f96e381..915e596 100644
--- a/fs/lockd/clntproc.c
+++ b/fs/lockd/clntproc.c
@@ -130,11 +130,11 @@ static void nlmclnt_setlockargs(struct n
    nlmclnt_next_cookie(&argp->cookie);

```

```

argp->state = nsm_local_state;
memcpy(&lock->fh, NFS_FH(fl->fl_file->f_dentry->d_inode), sizeof(struct nfs_fh));
- lock->caller = system_utsname.nodename;
+ lock->caller = utsname()->nodename;
lock->oh.data = req->a_owner;
lock->oh.len = snprintf(req->a_owner, sizeof(req->a_owner), "%u@%s",
    (unsigned int)fl->fl_u.nfs_fl.owner->pid,
- system_utsname.nodename);
+ utsname()->nodename);
lock->svid = fl->fl_u.nfs_fl.owner->pid;
lock->fl.fl_start = fl->fl_start;
lock->fl.fl_end = fl->fl_end;
diff --git a/fs/lockd/mon.c b/fs/lockd/mon.c
index 3fc683f..547aaa3 100644
--- a/fs/lockd/mon.c
+++ b/fs/lockd/mon.c
@@ -152,7 +152,7 @@ @ @ xdr_encode_common(struct rpc_rqst *rqstp
 */
sprintf(buffer, "%u.%u.%u.%u", NIPQUAD(argp->addr));
if (!(p = xdr_encode_string(p, buffer)))
- || !(p = xdr_encode_string(p, system_utsname.nodename)))
+ || !(p = xdr_encode_string(p, utsname()->nodename)))
    return ERR_PTR(-EIO);
*p++ = htonl(argp->prog);
*p++ = htonl(argp->vers);
diff --git a/fs/lockd/svclock.c b/fs/lockd/svclock.c
index d2b66ba..61b4791 100644
--- a/fs/lockd/svclock.c
+++ b/fs/lockd/svclock.c
@@ -326,7 +326,7 @@ @ @ static int nlmsvc_setgrantargs(struct nl
{
locks_copy_lock(&call->a_args.lock.fl, &lock->fl);
memcpy(&call->a_args.lock(fh, &lock->fh, sizeof(call->a_args.lock(fh)));
- call->a_args.lock.caller = system_utsname.nodename;
+ call->a_args.lock.caller = utsname()->nodename;
call->a_args.lock.oh.len = lock->oh.len;

/* set default data area */
diff --git a/fs/lockd/xdr.c b/fs/lockd/xdr.c
index f22a376..4eec051 100644
--- a/fs/lockd/xdr.c
+++ b/fs/lockd/xdr.c
@@ -516,7 +516,7 @@ @ @ nlmclt_decode_res(struct rpc_rqst *req,
*/
#define NLM_void_sz 0
#define NLM_cookie_sz 1+XDR_QUADLEN(NLM_MAXCOOKIELEN)
-#define NLM_caller_sz 1+XDR_QUADLEN(sizeof(system_utsname.nodename))
+#define NLM_caller_sz 1+XDR_QUADLEN(sizeof(utsname()->nodename))

```

```

#define NLM_netobj_sz 1+XDR_QUADLEN(XDR_MAX_NETOBJ)
/* #define NLM_owner_sz 1+XDR_QUADLEN(NLM_MAXOWNER) */
#define NLM_fhandle_sz 1+XDR_QUADLEN(NFS2_FHSIZE)
diff --git a/fs/nfs/nfsroot.c b/fs/nfs/nfsroot.c
index c0a754e..1d656a6 100644
--- a/fs/nfs/nfsroot.c
+++ b/fs/nfs/nfsroot.c
@@ -312,7 +312,7 @@ static int __init root_nfs_name(char *na
 /* Override them by options set on kernel command-line */
 root_nfs_parse(name, buf);

- cp = system_utsname.nodename;
+ cp = utsname()->nodename;
 if (strlen(buf) + strlen(cp) > NFS_MAXPATHLEN) {
 printk(KERN_ERR "Root-NFS: Pathname for remote directory too long.\n");
 return -1;
diff --git a/include/linux/lockd/lockd.h b/include/linux/lockd/lockd.h
index 995f89d..ac15b87 100644
--- a/include/linux/lockd/lockd.h
+++ b/include/linux/lockd/lockd.h
@@ -80,7 +80,7 @@ struct nlm_wait;
/*
 * Memory chunk for NLM client RPC request.
 */
#define NLMCLNT_OHSIZE (sizeof(system_utsname.nodename)+10)
+#define NLMCLNT_OHSIZE (sizeof(utsname()->nodename)+10)
struct nlm_rqst {
 unsigned int a_flags; /* initial RPC task flags */
 struct nlm_host * a_host; /* host handle */
diff --git a/kernel/sys.c b/kernel/sys.c
index 0b6ec0e..bcaa48e 100644
--- a/kernel/sys.c
+++ b/kernel/sys.c
@@ -1671,7 +1671,7 @@ asmlinkage long sys_newuname(struct new_
 int errno = 0;

 down_read(&uts_sem);
- if (copy_to_user(name,&system_utsname,sizeof *name))
+ if (copy_to_user(name,utsname(),sizeof *name))
 errno = -EFAULT;
 up_read(&uts_sem);
 return errno;
@@ -1689,8 +1689,8 @@ asmlinkage long sys_sethostname(char __u
 down_write(&uts_sem);
 errno = -EFAULT;
 if (!copy_from_user(tmp, name, len)) {
- memcpy(system_utsname.nodename, tmp, len);
- system_utsname.nodename[len] = 0;

```

```

+ memcpy(utsname()->nodename, tmp, len);
+ utsname()->nodename[len] = 0;
  errno = 0;
}
up_write(&uts_sem);
@@ -1706,11 +1706,11 @@ asmlinkage long sys_gethostname(char __u
if (len < 0)
  return -EINVAL;
down_read(&uts_sem);
- i = 1 + strlen(system_utsname.nodename);
+ i = 1 + strlen(utsname()->nodename);
if (i > len)
  i = len;
errno = 0;
- if (copy_to_user(name, system_utsname.nodename, i))
+ if (copy_to_user(name, utsname()->nodename, i))
  errno = -EFAULT;
up_read(&uts_sem);
return errno;
@@ -1735,8 +1735,8 @@ asmlinkage long sys_setdomainname(char __
down_write(&uts_sem);
errno = -EFAULT;
if (!copy_from_user(tmp, name, len)) {
- memcpy(system_utsname.domainname, tmp, len);
- system_utsname.domainname[len] = 0;
+ memcpy(utsname()->domainname, tmp, len);
+ utsname()->domainname[len] = 0;
  errno = 0;
}
up_write(&uts_sem);
--
```

1.2.4
