

[PATCH 2.4] add SMBIOS information to /proc/smbios -- UPDATE 2

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-04/6632.html>

From: Michael Brown (*mebrown_at_michaels-house.net*)

Date: 04/30/04

To: linux-kernel@vger.kernel.org, marcelo.tosatti@cyclades.com, michael_e_brown@dell.com
Date: Fri, 30 Apr 2004 00:08:05 -0500

Marcelo, Al,

Below is the updated patch for /proc/smbios/, with all changes integrated per Al's review comments (round 2). Patch has been compile, boot and stress tested.

Al, if this patch is ok with you, can you please ack it with Marcelo? If not, please let me know if there are other issues and I will take care of them tomorrow, as it is bedtime for me. Once again, thanks for reviewing this and taking time to point out the problems for me. I really appreciate it.

For reference, here are previous postings:

Previous 2.4 thread:

<http://marc.theaimsgroup.com/?t=108321757100001&r=1&w=1>

Previous 2.6 thread:

<http://marc.theaimsgroup.com/?t=108311959700002&r=1&w=1>

Thank you,
Michael Brown

```
# This is a BitKeeper generated diff -Nru style patch.
#
# ChangeSet
# 2004/04/29 23:58:04-05:00 michael_e_brown@meb-laptop.michaels-house.net
# changes per code review from Al Viro.
#
# drivers/char/smbios.c
# 2004/04/29 23:58:02-05:00 michael_e_brown@meb-laptop.michaels-house.net +11 -9
# Updates per comments from Al Viro. Fix stylistic issue, unsigned vs signed issue, and check return value
from put_user.
#
# ChangeSet
# 2004/04/29 21:11:19-05:00 michael_e_brown@meb-laptop.michaels-house.net
# Updates per Al Viro:
```

Linux-Kernel: [PATCH 2.4] add SMBIOS information to /proc/smbios -- UPDATE 2

```
# -- remove MOD_{INC,DEC}_USE_COUNT
# -- use normal read call instead of proc read call
#
# drivers/char/smbios.c
# 2004/04/29 21:11:17-05:00 michael_e_brown@meb-laptop.michaels-house.net +35 -47
# Updates per comments from Al Viro.
#
# ChangeSet
# 2004/04/29 00:16:35-05:00 michael_e_brown@meb-laptop.michaels-house.net
# rename smbios to proper smbios.c
#
# drivers/char/Makefile
# 2004/04/29 00:09:15-05:00 michael_e_brown@meb-laptop.michaels-house.net +1 -1
# rename smbios driver to 'smbios'.
#
# drivers/char/smbios.c
# 2004/04/29 00:08:51-05:00 michael_e_brown@meb-laptop.michaels-house.net +0 -0
# Rename: drivers/char/procsmbios.c -> drivers/char/smbios.c
#
# ChangeSet
# 2004/04/28 23:51:19-05:00 michael_e_brown@meb-laptop.michaels-house.net
# Add SMBIOS driver that exports system SMBIOS information as files in /proc/smbios.
#
# drivers/char/smbios.h
# 2004/04/28 23:50:15-05:00 michael_e_brown@meb-laptop.michaels-house.net +53 -0
#
# drivers/char/procsmbios.c
# 2004/04/28 23:50:15-05:00 michael_e_brown@meb-laptop.michaels-house.net +251 -0
#
# drivers/char/smbios.h
# 2004/04/28 23:50:15-05:00 michael_e_brown@meb-laptop.michaels-house.net +0 -0
# BitKeeper file /home/michael_e_brown/kernel/bk/subtrees/smbios-24-driver/drivers/char/smbios.h
#
# drivers/char/procsmbios.c
# 2004/04/28 23:50:15-05:00 michael_e_brown@meb-laptop.michaels-house.net +0 -0
# BitKeeper file /home/michael_e_brown/kernel/bk/subtrees/smbios-24-driver/drivers/char/procsmbios.c
#
# drivers/char/Makefile
# 2004/04/28 23:50:15-05:00 michael_e_brown@meb-laptop.michaels-house.net +1 -0
# Add SMBIOS driver to build system
#
# drivers/char/Config.in
# 2004/04/28 23:50:15-05:00 michael_e_brown@meb-laptop.michaels-house.net +1 -0
# Add config option to turn on SMBIOS proc driver
#
# Documentation/Configure.help
# 2004/04/28 23:50:15-05:00 michael_e_brown@meb-laptop.michaels-house.net +14 -0
# Add help text for SMBIOS driver
#
diff -Nru a/Documentation/Configure.help b/Documentation/Configure.help
--- a/Documentation/Configure.help Thu Apr 29 23:58:22 2004
```

Linux-Kernel: [PATCH 2.4] add SMBIOS information to /proc/smbios -- UPDATE 2

```
+++ b/Documentation/Configure.help Thu Apr 29 23:58:22 2004
@@ -19914,6 +19914,20 @@
```

If unsure, say N.

```
+SMBIOS table information in /proc
+CONFIG_SMBIOS
+ This driver creates two files in /proc, /proc/smbios/table_entry_point
+ and /proc/smbios/table. These two files contain the contents of the
+ BIOS-generated SMBIOS tables. Generally, PC-like architectures made
+ after 1997 have this, it is safe to enable on any system.
+
+ To compile this driver as a module (= code which can be inserted in
+ and removed from the running kernel whenever you want), say M here
+ and read <file:Documentation/modules.txt>. The module will be called
+ smbios.o.
```

+ If unsure, say N.

```
+
+ Sony Vaio Programmable I/O Control Device support
+CONFIG_SONYPI
```

This driver enables access to the Sony Programmable I/O Control

```
diff -Nru a/drivers/char/Config.in b/drivers/char/Config.in
--- a/drivers/char/Config.in Thu Apr 29 23:58:22 2004
+++ b/drivers/char/Config.in Thu Apr 29 23:58:22 2004
@@ -333,6 +333,7 @@
```

```
if [ "$CONFIG_EXPERIMENTAL" = "y" -a "$CONFIG_X86" = "y" -a "$CONFIG_X86_64" != "y" ]; then
    dep_tristate 'Sony Vaio Programmable I/O Control Device support (EXPERIMENTAL)'
```

```
CONFIG_SONYPI $CONFIG_PCI
fi
```

```
+tristate 'SMBIOS table information in /proc' CONFIG_SMBIOS
```

```
mainmenu_option next_comment
comment 'Ftape, the floppy tape device driver'
diff -Nru a/drivers/char/Makefile b/drivers/char/Makefile
--- a/drivers/char/Makefile Thu Apr 29 23:58:22 2004
+++ b/drivers/char/Makefile Thu Apr 29 23:58:22 2004
@@ -265,6 +265,7 @@
obj-$(CONFIG_HW_RANDOM) += hw_random.o
obj-$(CONFIG_AMD_PM768) += amd76x_pm.o
obj-$(CONFIG_BRIQ_PANEL) += briq_panel.o
+obj-$(CONFIG_SMBIOS) += smbios.o
```

```
obj-$(CONFIG_ITE_GPIO) += ite_gpio.o
obj-$(CONFIG_AU1X00_GPIO) += au1000_gpio.o
diff -Nru a/drivers/char/smbios.c b/drivers/char/smbios.c
--- /dev/null Wed Dec 31 16:00:00 1969
+++ b/drivers/char/smbios.c Thu Apr 29 23:58:22 2004
@@ -0,0 +1,241 @@
+/*
+ * linux/drivers/firmware/smbios.c
```

Linux-Kernel: [PATCH 2.4] add SMBIOS information to /proc/smbios -- UPDATE 2

```
+ * Copyright (C) 2002, 2003, 2004 Dell Inc.
+ * by Michael Brown <Michael_E_Brown@dell.com>
+ * vim:noet:ts=8:sw=8:filetype=c:textwidth=80:
+ *
+ * BIOS SMBIOS Table access
+ * conformant to DMTF SMBIOS definition
+ * at http://www.dmtf.org/standards/smbios
+ *
+ * This code takes information provided by SMBIOS tables
+ * and presents it in procfs as:
+ * /proc/smbios
+ * |--> /table_entry_point
+ * |--> /table
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License v2.0 as published by
+ * the Free Software Foundation
+ *
+ * This program is distributed in the hope that it will be useful,
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
+ * GNU General Public License for more details.
+ *
+ */
+
+#include <linux/module.h>
+#include <linux/init.h>
+#include <linux/proc_fs.h>
+#include <asm/io.h>
+#include <asm/uaccess.h>
+#include "smbios.h"
+
+#define SMBIOS_VERSION "1.0"
+#define SMBIOS_DATE "2004-04-29"
+
+MODULE_AUTHOR("Michael Brown <Michael_E_Brown@Dell.com>");
+MODULE_DESCRIPTION("procfs interface to SMBIOS information");
+MODULE_LICENSE("GPL");
+EXPORT_NO_SYMBOLS;
+
+struct smbios_device {
+ struct smbios_table_entry_point table_eps;
+ unsigned int smbios_table_real_length;
+};
+
+/* there shall be only one */
+static struct smbios_device the_smbios_device;
+
+static struct proc_dir_entry *smbios_dir, *table_eps_file, *table_file;
+
```

```

+static __init int
+checksum_eps(struct smbios_table_entry_point *table_eps)
+{
+ u8 *p = (u8 *)table_eps;
+ u8 checksum = 0;
+ int i=0;
+ for (i=0; i < table_eps->eps_length && i < sizeof(*table_eps); ++i) {
+ checksum += p[i];
+ }
+ return(checksum == 0);
+}
+
+static __init int
+find_table_entry_point(struct smbios_device *sdev)
+{
+ struct smbios_table_entry_point *table_eps = &(sdev->table_eps);
+ u32 fp;
+ for (fp = 0xF0000; fp < 0xFFFFF; fp += 16) {
+ isa_memcpy_fromio(table_eps, fp, sizeof(*table_eps));
+ if (memcmp(table_eps->anchor, "_SM_", 4)!=0)
+ continue;
+ if (checksum_eps(table_eps))
+ return 0;
+ }
+
+ printk(KERN_INFO "SMBIOS table entry point not found in "
+ "0xF0000 - 0xFFFFF\n");
+ return -ENODEV;
+}
+
+static __init int
+find_table_max_address(struct smbios_device *sdev)
+{
+ /* break out on one of three conditions:
+ * -- hit table_eps.table_length
+ * -- hit number of items that table claims we have
+ * -- hit structure type 127
+ */
+
+ u8 *buf = ioremap(sdev->table_eps.table_address,
+ sdev->table_eps.table_length);
+ u8 *ptr = buf;
+ int count = 0, keep_going = 1;
+ int max_count = sdev->table_eps.table_num_structs;
+ int max_length = sdev->table_eps.table_length;
+ while(keep_going && ((ptr - buf) <= (max_length-1)) && count < max_count){
+ if (ptr[0] == 0x7F) /* ptr[0] is type */
+ keep_going = 0;
+
+ ptr += ptr[1]; /* ptr[1] is length, skip structure */
+ /* skip strings at end of structure */

```

```

+ while((ptr-buf) < max_length && (ptr[0] || ptr[1]))
+ ++ptr;
+
+ /* string area ends in double-null. skip it. */
+ ptr += 2;
+ ++count;
+ }
+ sdev->smbios_table_real_length = (ptr - buf);
+ iounmap(buf);
+
+ if (count != max_count)
+ printk(KERN_INFO "Warning: SMBIOS table structure count"
+ " does not match count specified in the"
+ " table entry point.\n"
+ " Table entry point count: %d\n"
+ " Actual count: %d\n",
+ max_count, count);
+
+ if (keep_going != 0)
+ printk(KERN_INFO "Warning: SMBIOS table does not end with a"
+ " structure type 127. This may indicate a"
+ " truncated table.");
+
+ if (sdev->smbios_table_real_length != max_length)
+ printk(KERN_INFO "Warning: BIOS specified SMBIOS table length"
+ " does not match calculated length.\n"
+ " BIOS specified: %d\n"
+ " calculated length: %d\n",
+ max_length, sdev->smbios_table_real_length);
+
+ return sdev->smbios_table_real_length;
+}
+
+/* simple procfs style. Print the whole thing and let core
+ * handle splitting it out for userspace and setting eof.
+ */
+static ssize_t
+smbios_read_table_entry_point(char *page, char **start,
+ off_t off, int count,
+ int *eof, void *data)
+{
+ unsigned int max_off = sizeof(the_smbios_device.table_eps);
+ memcpy(page, &the_smbios_device.table_eps, max_off);
+ return max_off;
+}
+
+static ssize_t smbios_read_table(struct file *file, char *buf,
+ size_t count, loff_t *ppos)
+{
+ unsigned long origppos = *ppos;
+ unsigned long max_off = the_smbios_device.smbios_table_real_length;

```

Linux-Kernel: [PATCH 2.4] add SMBIOS information to /proc/smbios -- UPDATE 2

```
+ u8 *ptr;
+ int ret;
+
+ if(*ppos >= max_off || *ppos < 0)
+ return 0;
+
+ if(count > (max_off - *ppos))
+ count = max_off - *ppos;
+
+ ptr = ioremap(the_smbios_device.table_eps.table_address, max_off);
+ if (ptr == NULL)
+ return -ENXIO;
+
+ while (*ppos < max_off) {
+ ret = put_user(readb(ptr + *ppos), buf);
+ if( ret )
+ return ret;
+ ++(*ppos); ++buf;
+ }
+ iounmap(ptr);
+ return *ppos - origppos;
+}
+
+static struct file_operations proc_smbios_table_operations = {
+ .read = smbios_read_table,
+};
+
+static int __init
+smbios_init(void)
+{
+ int rc=0;
+
+ printk(KERN_INFO "SMBIOS facility v%s %s\n",
+ SMBIOS_VERSION, SMBIOS_DATE);
+
+ rc = find_table_entry_point(&the_smbios_device);
+ if (rc)
+ return rc;
+
+ find_table_max_address(&the_smbios_device);
+
+ rc = -ENOMEM;
+ smbios_dir = proc_mkdir("smbios", NULL);
+ if (smbios_dir == NULL)
+ goto out;
+
+ smbios_dir->owner = THIS_MODULE;
+
+ table_eps_file = create_proc_read_entry("table_entry_point",
+ 0444, smbios_dir,
+ smbios_read_table_entry_point,
```

Linux-Kernel: [PATCH 2.4] add SMBIOS information to /proc/smbios -- UPDATE 2

```
+ NULL);
+ if (table_eps_file == NULL)
+ goto no_table_eps_file;
+
+ table_eps_file->owner = THIS_MODULE;
+
+ table_file = create_proc_entry("table", 0444, smbios_dir);
+ if (table_file == NULL)
+ goto no_table_file;
+
+ table_file->proc_fops = &proc_smbios_table_operations;
+ table_file->owner = THIS_MODULE;
+
+ rc = 0;
+ goto out;
+no_table_file:
+ remove_proc_entry("table_entry_point", smbios_dir);
+
+no_table_eps_file:
+ remove_proc_entry("smbios", NULL);
+
+out:
+ return rc;
+}
+
+static void __exit
+smbios_exit(void)
+{
+ remove_proc_entry("table_entry_point", smbios_dir);
+ remove_proc_entry("table", smbios_dir);
+ remove_proc_entry("smbios", NULL);
+}
+
+module_init(smbios_init);
+module_exit(smbios_exit);
diff -Nru a/drivers/char/smbios.h b/drivers/char/smbios.h
--- /dev/null Wed Dec 31 16:00:00 1969
+++ b/drivers/char/smbios.h Thu Apr 29 23:58:22 2004
@@ -0,0 +1,53 @@
+/*
+ * linux/drivers/firmware/smbios.c
+ * Copyright (C) 2002, 2003, 2004 Dell Inc.
+ * by Michael Brown <Michael_E_Brown@dell.com>
+ * vim:noet:ts=8:sw=8:filetype=c:textwidth=80:
+ *
+ * BIOS SMBIOS Table access
+ * conformant to DMTF SMBIOS definition
+ * at http://www.dmtf.org/standards/smbios
+ *
+ * This code takes information provided by SMBIOS tables
+ * and presents it in sysfs.
```

Linux-Kernel: [PATCH 2.4] add SMBIOS information to /proc/smbios -- UPDATE 2

```
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License v2.0 as published by
+ * the Free Software Foundation
+ *
+ * This program is distributed in the hope that it will be useful,
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
+ * GNU General Public License for more details.
+ *
+ */
+
+#ifndef _LINUX_SMBIOS_H
+#define _LINUX_SMBIOS_H
+
+#include <linux/types.h>
+
+struct smbios_table_entry_point {
+ u8 anchor[4];
+ u8 checksum;
+ u8 eps_length;
+ u8 major_ver;
+ u8 minor_ver;
+ u16 max_struct_size;
+ u8 revision;
+ u8 formatted_area[5];
+ u8 dmi_anchor[5];
+ u8 intermediate_checksum;
+ u16 table_length;
+ u32 table_address;
+ u16 table_num_structs;
+ u8 smbios_bcd_revision;
+} __attribute__((packed));
+
+struct smbios_structure_header {
+ u8 type;
+ u8 length;
+ u16 handle;
+} __attribute__((packed));
+
+#endif /* _LINUX_SMBIOS_H */
```

—
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@vger.kernel.org
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>