

## Re: statistics infrastructure (in -mm tree) review

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-06/msg03906.html>

---

- *From:* "Randy.Dunlap" <[rdunlap@xxxxxxxxxxxxx](mailto:rdunlap@xxxxxxxxxxxxx)>
  - *Date:* Tue, 13 Jun 2006 17:18:27 -0700
- 

On Tue, 13 Jun 2006 16:47:39 -0700 Greg KH wrote:

First cut at reviewing this code.

Initial impression is, "dammm, that's a complex interface". I'd really like to see some other, real-world usages of this. Like perhaps the io-scheduler statistics? Some other /proc stats that have nothing to do with processes?

Agreed with complexity.

And what does this mean for relayfs? Those developers tuned that code to the nth degree to get speed and other goodness, and here you go just ignoring that stuff and add yet another way to get stats out of the kernel. Why should I use this instead of my own code with relayfs?

Good questions.

And is the need for the in-kernel parser really necessary? I know it makes the userspace tools simpler (cat and echo), but should we be telling the kernel how to filter and adjust the data? Shouldn't we just dump it all to userspace and use tools there to manipulate it?

I agree again.

Code comments now:

```
diff -puN /dev/null include/linux/statistic.h
---- /dev/null 2006-06-03 22:34:36.282200750 -0700
+++ devel-akpm/include/linux/statistic.h 2006-06-09 15:22:58.000000000
```

Re: statistics infrastructure (in -mm tree) review

```
-0700
@@ -0,0 +1,348 @@
+/*
+ * include/linux/statistic.h
+ *
+ * Statistics facility

+/**
+ * struct statistic_info - description of a class of statistics
+ * @name: pointer to name name string
+ * @x_unit: pointer to string describing unit of X of (X, Y) data pair
+ * @y_unit: pointer to string describing unit of Y of (X, Y) data pair
+ * @flags: only flag so far (distinction of incremental and other statistic)
+ * @defaults: pointer to string describing defaults setting for attributes
+ *
+ * Exploiters must setup an array of struct statistic_info for a
+ * corresponding array of struct statistic, which are then pointed to
+ * by struct statistic_interface.
+ *
+ * Struct statistic_info and all members and addressed strings must stay for
+ * the lifetime of corresponding statistics created with statistic_create().
+ *
+ * Except for the name string, all other members may be left blank.
+ * It would be nice of exploiters to fill it out completely, though.
+ */
+struct statistic_info {
+/* public: */
+ char *name;
+ char *x_unit;
+ char *y_unit;
+ int flags;
+ char *defaults;
+};
```

The whole "public:" and "private:" thing in these structures is not needed. Just document it in the kernel-doc comments and you should be fine. This isn't C++ :)

but public: and private: are kernel-doc comments...  
Using "private:" causes those fields to be omitted from the generated documentation because those fields are for internal/private use of the (statistics) infrastructure code, not to be used by its clients (er, ugh, exploiters) etc.

```
--- /dev/null 2006-06-03 22:34:36.282200750 -0700
+++ devel-akpm/lib/statistic.c 2006-06-09 15:22:58.000000000 -0700
```

Re: statistics infrastructure (in -mm tree) review

```
@@ -0,0 +1,1459 @@  
+/*  
+ * lib/statistic.c  
+ * statistics facility  
+ *
```

Again with the verbose license :)

Well it's not uncommon in kernel source files.  
Where do we document how licenses should be written?

---

~Randy

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in  
the body of a message to majordomo@xxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>