Subject: Re: [PATCH 2/8] CGroup Files: Add write_string cgroup control file method
Posted by Paul Menage on Tue, 24 Jun 2008 23:26:21 GMT
View Forum Message <> Reply to Message

On Tue, Jun 24, 2008 at 4:19 PM, Andrew Morton
<akpm@linux-foundation.org> wrote:
>>   /*
>> +   * write_string() is passed a nul-terminated kernelspace
>> +   * buffer of maximum length determined by max_write_len.
>> +   * Returns 0 or -ve error code.
>> +   */
>> +   int (*write_string)(struct cgroup *cgrp, struct cftype *cft,
>> +   const char *buffer);
>
> Everything seems to use size_t (or ssize_t?) except for the ->write_string
> return value. Can any of this be improved?

What other things are you including as "everything"?

write_string() returns 0 on success or a -ve error code on failure -
it doesn't have the concept of writing some fraction of the passed
bytes.

The functions that deal in size_t/ssize_t (along with userspace
buffers, files and position pointers) are the glue that interfaces
with the filesystem layer. My aim (which is furthered by this patch
series) is to keep as much of that as possible in the cgroup layer
itself, and to reduce filesystem glue in the cgroup subsystems. The
raw file interface is still exposed by cgroups for those subsystems
that really need it, but it should be the exception rather than the
rule.

Paul

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers