Subject: Re: [PATCH 2/6] BC: beancounters core (API)
Posted by Matt Helsley on Wed, 23 Aug 2006 22:05:55 GMT
View Forum Message <> Reply to Message

```
On Wed, 2006-08-23 at 17:49 +0400, Kirill Korotaev wrote:
> Alexey Dobriyan wrote:
> > On Wed, Aug 23, 2006 at 03:03:07PM +0400, Kirill Korotaev wrote:
> >
> >
> >>--- /dev/null
> >>+++ ./include/bc/beancounter.h
> >
> >
> >>+#define BC_RESOURCES 0
> >
> >
> > Do you want userspace to see it?
> yep.
>
> >>+struct bc_resource_parm {
> >>+ unsigned long barrier; /* A barrier over which resource allocations
> >>+     * are failed gracefully. e.g. if the amount
> >>+     * of consumed memory is over the barrier
> >>+     * further sbrk() or mmap() calls fail, the
> >>+     * existing processes are not killed.
> >>+     */
> >>+ unsigned long limit; /* hard resource limit */
> >>+ unsigned long held; /* consumed resources */
> >>+ unsigned long maxheld; /* maximum amount of consumed resources */
> >>+ unsigned long minheld; /* minumum amount of consumed resources */
> >
> >
> > Stupid question: when minimum amount is useful?
> to monitor usage statistics (range of used resources).
> this value will be usefull when ubstat will be added.
> this field probably would be more logical to add later,
> but since it is part of user space interface it is left here
> for not changing API later.
```

Then I think it belongs in a separate patch. Add it and the scattered bits and pieces that use it with the same patch. Then folks can clearly see what it's for, where it impacts the code, and how it works. Yes, factoring it out causes the API to evolve over the course of applying the patch series -- IMHO that evolution is useful information to convey to reviewers too.

Cheers,
-Matt Helsley