

Re: VM question related to faults

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/hackers/2006-07/msg00476.html>

- *From:* "Attilio Rao" <attilio@xxxxxxxxxxxx>
 - *Date:* Mon, 31 Jul 2006 20:15:11 +0200
-

2006/7/30, Divacky Roman <xdivac02@xxxxxxxxxxxxxxxxxxxx>:

On Sun, Jul 30, 2006 at 12:57:32PM +0200, Divacky Roman wrote:

> hi,
>
> while working on SoC linuxolator project I am in a need of this:
>
> I need to do some operation on memory like mem1 = mem1 + mem2 etc.
> where the mem1/mem2 access can trigger fault. (memory not mapped or something)

to make it clear.. I am trying to access user-space memory from kernel.
This needs to be atomic (its an implementation of linux futexes)

I need to check from kernel if some memory is accessible and then perform an operation on this memory. All atomically.

hence I need two things – function which checks wheter the memory is accessible and something which makes it atomic (some mutex/something which prevents other process to enter VM to unmap/etc. the memory in question)

hope its a bit more clear now

You would use something like:

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/time.h>
#include <sys/lock.h>
#include <sys/mutex.h>
#include <sys/system.h>
#include <sys/resource.h>

#include <vm/vm_page.h>

...
int
lock_and_fetch(const void* mem1, const void *mem2)
{
```

Re: VM question related to faults

```
mtx_lock(&vm_page_queue_mtx);
if (fubyte(mem1) == -1 || fubyte(mem2) == -1) {
mtx_unlock(&vm_page_queue_mtx);
return(EINVAL);
}
/* Operations... */
mtx_unlock(&vm_page_queue_mtx);

return(0);
}
```

It prevents to virtual pages to be passed through queues.

Attilio

—

Peace can only be achieved by understanding – A. Einstein

freebsd-hackers@xxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-hackers>

To unsubscribe, send any mail to "freebsd-hackers-unsubscribe@xxxxxxxxxxx"