NC STATE UNIVERSITY Message Board

Course: CSC501 - 001
Forum: PA3 discussion

Go to: Different Course | Forum List | Topic List

Author **Topic: BS and multiple processes**

adhanas Mon Mar 17 2:38 Who

In another thread (http://classic.wolfware.ncsu.edu/wrap-bin/mesgboard/csc:501::001:1:2014?task=ST&Forum=5&Topic=1) it was told that multiple processes doing a xmmap() could be mapped to the same BS, provided the BS doesn't belong to a vheap.

But, FAQ #3 says and I quote:
"3) About the mapping <pid, vpage, npages, store>
This mapping is maintained inside the kernel. Since the "store" can take only 16 values at the most (because there are only 16 backing stores possible for any user), and no store can be mapped to more than one range of virtual memory at any time, the table that contains these mappings will contain only 16 entries."

Going by what's given in FAQ, we can have only have a 1:1 mapping between a BS and a virtual memory range (and so, a pid). This seems to be contradicting the earlier argument.

What am I missing here? Thanks!

wzhang27 Mon Mar 17 8:54 Who

I don't quite get your previous arguments on the other post. I think there might be some wording problem.

But, in a newly created process, if you want to use BS, you should first get_bs(), and then xmmap().

adhanas Mon Mar 17 10:37 Who

---

1. FAQ #3 says that each BS will be mapped to only one virtual address range at any given time (by that, only to one process).

2. But, in the other thread, Hiep mentioned that the same BS, say BS0, can be used for two distinct processes doing an xmmap(), provided the BS0 is not being used to host a vheap.

These two points seems to contradict each other.

hcnguye3 Mon Mar 17 11:15 Who

@adhanas: BS will be mapped to only one virtual address range at any given time for EACH process. I agree that the FAQ#3 seems a bit confusing, I will try to reword it. You can take a look at FAQ#7 to see how two processes share the same BS

adhanas Tue Mar 18 11:53 Who

Below is the slightly modified version of FAQ #7 code snippet. Assume both the processes are working on the same BS.

```
procA() {
get_bs();
xmmap();
xunmap();
release_bs();
}

procB() {
get_bs();
xmmap();
xunmap();
release_bs();
}
```

Couple of questions:
1. What should happen if procB calls xmmap() before procA unmaps it?
2. What should happen if procB happens to release the BS before procA unmaps it? This could be dependent on the solution of question 1 above.

hcnguye3 Tue Mar 18 23:02 Who

You dont have to worry about that. That is users' responsibility if they want to have synchronization. You only asked to allow sharing backing store between processes

adhanas Sun Mar 23 11:18 Who

"@adhanas: BS will be mapped to only one virtual address range at any given time for EACH process."

Should our implementation include a check for this or is it assumed that the grading program will not map different VA ranges to the same BS?

rsinghc Tue Mar 25 21:36 Who

+1

mjoshi Thu Mar 27 6:50 Who

"You dont have to worry about that. That is users' responsibility if they want to have synchronization. You only asked to allow sharing backing store between processes"

I understand this. I think the question remains open.

Consider below scenario:

```
procA() {
xmmap(4096, 0, 100);

char *addr = 0x10000000;
*addr = 'A';
}
```

A page fault occurs here and page in brought from BS into main memory.

Please note that procA does not xmunmap the memory.

Now, if another process B executes

```
procB() {
xmmap(6000, 0, 100);

*addr = 0x10000000; // same as above

printf("%c\n", *addr);
}
```

Should the 'A' get printed in procB?

I am asking this since if procA does not xmunmap the memory, the data written in physical pages will not be returned to the backing store (my understanding). I think it should not get printed unless procA executes xmunmap.

Please correct me if I am wrong.

adhanas Fri Mar 28 1:40 Who

My previous question on the same thread:

"Should our implementation include a check for this or is it assumed that the grading program will not map different VA ranges to the same BS?"

Any suggestions?

hcnguye3 Fri Mar 28 9:42 Who

@mjoshi: It looks like you misunderstood. Note that each process has its own virtual address space. That means even A and B access the same virtual address, it does not mean they will get the same value. Sharing is done through mapping to the same page in the same backing store (eventually become sharing a physical frame)

@ adhanas: You should implement a check. You can return SYSERR in that case.

rrathale Fri Mar 28 10:03 Who

So suppose a process does the following:

Process A:
xmmap(7000,1,100);
.
. // some code
.

xmmap(8000,1,100);

Then should the second xmmap return SYSERR? Or should the mapping starting at vpno 7000 be replaced by mapping starting at vpno 8000?

hcnguye3 Fri Mar 28 10:41 Who

I assume that there is no xmumap in "some code".
return SYSERR

rpageda Tue Apr 1 0:04 Who

---

what will happen in the following scenario?

use case:

get_bs(1,100);
xmmap(4000,1,50);
xmmap(5000,1,50);

should the second xmmap return SYSERR?

Post New Topic Post a Reply

WolfWare - Message Board - last update 30-Aug-2001