Subject: Question: remap_pfn_range changes in 2.6.2x
Posted by arusev on Tue, 01 Apr 2008 08:46:35 GMT
View Forum Message <> Reply to Message

The current kernel code for remap_pfn_range now includes the check of
vm_area size vs. mapped physical page area.

In earlier versions when vm_area size was more than
pfn-s to map to, the rest of vm_area was attached to on-demand page
allocation/COW so the userspace which mapped more just obtained a piece
of "shared memory" anonymous pages.

What is the reason behind that strict check now included in
remap_pfn_range?

How to implement the feature again?

I need to map a few pages to physical PFN-s and the rest of
to ZERO_PAGE where userland may do what it whishes.

Which functions should I use in my filesystem driver to achieve the same
effect???

This is an issue for mmaping shared library objects by ld-linux.so
at XIP-ed FLASH filesystems.

When shared library marked with "chmod +t" is loaded, the LD-linker
tries to mmap more than it's size due to get some extra place for
dynamic relocations and other housekeeping.

(I'm looking into 2.6.24 mainstream tree)

```
int remap_pfn_range(struct vm_area_struct *vma, unsigned long addr,
unsigned long pfn, unsigned long size, pgprot_t prot)
{
   ...
 if (is_cow_mapping(vma->vm_flags)) {
        if (addr != vma->vm_start || end != vma->vm_end){
            return -EINVAL;
        }
        vma->vm_pgoff = pfn;
 }

 ...
```

Could anybody give me a hint how this problem should to be dealt?