

**NAME**

**vm\_page\_insert**, **vm\_page\_remove** - add/remove page from an object

**SYNOPSIS**

```
#include <sys/param.h>
#include <vm/vm.h>
#include <vm/vm_page.h>
```

*void*

```
vm_page_insert(vm_page_t m, vm_object_t object, vm_pindex_t pindex);
```

*void*

```
vm_page_remove(vm_page_t m);
```

**DESCRIPTION**

The **vm\_page\_insert()** function adds a page to the given object at the given index. The page is added to both the VM page hash table and to the object's list of pages, but the hardware page tables are not updated. In the case of a user page, it will be faulted in when it is accessed. If the page is a kernel page, the caller is expected to handle adding the page to the kernel's pmap.

If PG\_WRITEABLE is set in the page's flags, OBJ\_WRITEABLE and OBJ\_MIGHTBEDIRTY are set in the object's flags.

The **vm\_page\_remove()** function removes the given page from its object, and from the VM page hash table. The page must be busy prior to this call, or the system will panic. The pmap entry for the page is not removed by this function.

The arguments to **vm\_page\_insert()** are:

*m*      The page to add to the object.

*object*   The object the page should be added to.

*pindex*

        The index into the object the page should be at.

The arguments to **vm\_page\_remove()** are:

*m*      The page to remove.

**IMPLEMENTATION NOTES**

The index of a page in a VM object is the byte index into the same object truncated to a page boundary. For example, if the page size is 4096 bytes, and the address in the object is 81944, the page index is 20.

**AUTHORS**

This manual page was written by Chad David <*davidc@acns.ab.ca*>.