

**NAME**

**vm\_page\_free**, **vm\_page\_free\_toq**, **vm\_page\_free\_zero**, **vm\_page\_try\_to\_free** - free a page

**SYNOPSIS**

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

```
#include <vm/vm.h>
```

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

*void*

```
vm_page_free(vm_page_t m);
```

*void*

```
vm_page_free_toq(vm_page_t m);
```

*void*

```
vm_page_free_zero(vm_page_t m);
```

*int*

```
vm_page_try_to_free(vm_page_t m);
```

**DESCRIPTION**

The **vm\_page\_free\_toq()** function moves a page into the free queue, and disassociates it from its object. If the page is held, wired, already free, or its busy count is not zero, the system will panic. If the **PG\_ZERO** flag is set on the page, it is placed at the end of the free queue; otherwise, it is placed at the front.

If the page's object is of type **OBJT\_VNODE** and it is the last page associated with the object, the underlying vnode may be freed.

The **vm\_page\_free()** and **vm\_page\_free\_zero()** functions both call **vm\_page\_free\_toq()** to actually free the page, but **vm\_page\_free\_zero()** sets the **PG\_ZERO** flag and **vm\_page\_free()** clears the **PG\_ZERO** flag prior to the call to **vm\_page\_free\_toq()**.

The **vm\_page\_try\_to\_free()** function verifies that the page is not held, wired, busy or dirty, and if so, marks the page as busy, drops any protection that may be set on the page, and frees it.

**RETURN VALUES**

**vm\_page\_try\_to\_free()** returns 1 if it is able to free the page; otherwise, 0 is returned.

**SEE ALSO**

vm\_page\_busy(9), vm\_page\_hold(9), vm\_page\_wire(9)

**AUTHORS**

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