

NAME

pmap_map - map a physical memory range into kernel virtual address (KVA) space

SYNOPSIS

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

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

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

```
vm_offset_t
```

```
pmap_map(vm_offset_t *virt, vm_paddr_t start, vm_paddr_t end, int prot);
```

DESCRIPTION

The **pmap_map**() function maps a range of physical addresses into kernel virtual address (KVA) space, from *start* to *end*, with protection bits *prot*.

The value passed in **virt* is treated as a hint for the virtual address of the beginning of the mapping.

IMPLEMENTATION NOTES

The *prot* argument is currently ignored by machine-dependent implementations.

Architectures which can support a direct mapped physical to virtual region can return the appropriate address within that region, leaving **virt* unchanged.

RETURN VALUES

The **pmap_map**() function returns the virtual address of the beginning of the mapping, if the mapping was successfully made; **virt* will also be updated with the first usable address after the mapped region.

If the function is unsuccessful, NULL is returned.

SEE ALSO

pmap(9)

AUTHORS

This manual page was written by Bruce M Simpson <bms@spc.org>.