

NAME

PAE - Physical Address Extensions

SYNOPSIS

options PAE

DESCRIPTION

The PAE option provides support for the physical address extensions capability of the Intel Pentium Pro and above CPUs, and allows for up to 64 gigabytes of memory to be used in systems capable of supporting it. With the PAE option, memory above 4 gigabytes is simply added to the general page pool. The system makes no distinction between memory above or below 4 gigabytes, and no specific facility is provided for a process or the kernel to access more memory than they would otherwise be able to access, through a sliding window or otherwise.

SEE ALSO

`smp(4)`, `tuning(7)`, `config(8)`, `bus_dma(9)`

HISTORY

The PAE option first appeared in FreeBSD 4.9 and FreeBSD 5.1.

AUTHORS

Jake Burkholder <jake@FreeBSD.org>

BUGS

Since KLD modules are not compiled with the same options headers that the kernel is compiled with, they must not be loaded into a kernel compiled with the PAE option.

Many devices or their device drivers are not capable of direct memory access to physical addresses above 4 gigabytes. In order to make use of direct memory access IO in a system with more than 4 gigabytes of memory when the PAE option is used, these drivers must use a facility for remapping or substituting physical memory which is not accessible to the device. One such facility is provided by the **busdma** interface. Device drivers which do not account for such devices will not work reliably in a system with more than 4 gigabytes of memory when the PAE option is used, and may cause data corruption. The *PAE* kernel configuration file includes the PAE option, and explicitly excludes all device drivers which are known to not work or have not been tested in a system with the PAE option and more than 4 gigabytes of memory.

Many parameters which determine how memory is used in the kernel are based on the amount of physical memory. The formulas used to determine the values of these parameters for specific memory configurations may not take into account the fact there may be more than 4 gigabytes of memory, and

may not scale well to these memory configurations. In particular, it may be necessary to increase the amount of virtual address space available to the kernel, or to reduce the amount of a specific resource that is heavily used, in order to avoid running out of virtual address space. The `KVA_PAGES` option may be used to increase the kernel virtual address space, and the `kern.maxvnodes` `sysctl(8)` may be used to decrease the number of vnodes allowed, an example of a resource that the kernel is likely to overallocate in large memory configurations. For optimal performance and stability it may be necessary to consult the `tuning(7)` manual page, and make adjustments to the parameters documented there.