

partition logical address space as far as User/Kernel space is concerned, especially when it comes to deciding what gets mapped to

# How does Linux partition logical address space as far as User/Kernel space is concerned, especially when it comes to deciding what gets mapped to virtual versus physical RAM?

*Source:* <http://linux.derkeiler.com/Newsgroups/comp.os.linux/2003-09/0447.html>

---

*From:* Dustin Darcy ([aktuld\\_at\\_hotmail.com](mailto:aktuld_at_hotmail.com))

*Date:* 09/20/03

Date: Sat, 20 Sep 2003 20:24:50 GMT

The norm for a memory allocation scheme is to use physical memory before dumping the overflow to disk. However with Linux it's unique in that there's a set limit for user and kernel space (aka monolithic kernel). Which leads me to believe that the logical address space is set aside ahead of time. Meaning that if a person has 2GBs of physical RAM and chooses option 3GBs/1GBs of user/kernel space, that either 1GB of kernel space and 1GB of user space will be allocated to physical RAM; OR, 2 GBs of user space will go to physical RAM while the remaining 2 GBs (1 user, the other kernel) will be virtual. This seems rather asinine, so I hardly believe this is how it works.

It was also proposed to me that the system only considers how much it – will– give to either user or kernel space. Meaning that at boot up, if a system only uses 2 MBs of Kernel space, only 2 MBs gets mapped from the the total available memory for Kernel space (as there's a cap). Also meaning that those first 2 MBs would be allocated to physical memory. Then as more would be requested more would be allocated for either user or kernel space, straight from physical memory, till it ran out. This sounds like a more solid schema, but at the same time it seems silly to do things on a first come, first serve basis. It seems more reasonable that it would partition each application such that each has some physical memory to work in, and then to use a memory map for anything additional to disk.

So the question is, how does Linux delegate it's memory with regards to user/kernel space? Does Linux treat User/Kernel space just as a hard limit, allocating memory as it's needed until it hits the cap for a particular "type" of memory? Or does it preordain that kernel space will be in physical memory while user space will get the remainder, whether it be physical or not? Is there possibly another partitioning scheme, similar to the one I suggested above?

How does Linux partition logical address space as far as User/Kernel space is concerned, especially when i

partition logical address space as far as User/Kernel space is concerned, especially when it comes to deciding what gets m

I'm beginning to believe it's a simple tallying scheme, checking to see how much kernel space or user space has been allocated.

ie/

3GB/1GB of User/Kernel space is available.

2 MBs gets allocated at boot-time for the kernel and it's modules –  
(1.000GBs–0.002GBs=NewAvailableKernelSpace)

50 MBs gets allocated to X in user space – (3.000GBs–  
0.050GBs=NewAvailableUserSpace)

So on and so forth.

I imagine that the logical address space is just a series of pointers telling the system where everything is. For example, the first logical address might point to physical memory for kernel space, while the second logical address unit might point to user space in virtual memory (similar to the example above). I imagine that's the whole value of logical address space – it provides the HAL.

Any reference material or solid answers surrounding the mysteries of Linux's memory allocation would be appreciated. :)

Thanks,  
Dustin

How does Linux partition logical address space as far as User/Kernel space is concerned, especially when i