Memory Management in Dux

Martin Brandenburg

Overview

Memory management is a core function of a computer operating system. Many computers today have over a billion bytes of random access storage. Memory management decides what goes where in memory.

User Memory

To put it simply, programs obtain more memory by requesting it. As the kernel, we are in charge of finding more memory for user consumption.

When a programs starts, it gets a new address space. The kernel maps itself and the executable to the address space and starts the program. The address space contains all code and data the program needs. In the event that it needs more memory, it can request more. Note that memory in unallocated sections of the address space will raise a fault.

During runtime, programs can call upon the kernel for more memory. The kernel picks a physical address and maps it to the program's address space.

When a program is closed, its' entire address space is freed for other programs and the kernel.

Kernel Memory

Th kernel allocates memory in the same way programs do—excepting that it asks itself.

During bootup, advanced memory management is not available. A simple placement-based allocator will be available to it during that time. Memory allocated from it can not be freed.

After the kernel isbooted it obtains memory from the heap, just like programs. Past this, simple implementations of `malloc()` and `free()` will be provided.
The kernel does not worry about freeing its memory upon shutdown.

Physical Memory

Memory allocation takes place in two domains—physical and virtual. Physical memory is erratic and full of holes. Good portions of it are stringed into a virtual space for both programs and the kernel.

The physical allocator allocates page frames. Allocated pages can be freed for use.