# Ideas

## Dynamic Linking

Most OSs (Including Linux and Windows XP) use a runtime linkage table to achieve dynamic linking. They work by being effectively an extra level of indirection. This process is very slow, both when loading the program, and running it. My idea is to use an address table. This table would contain byte offsets into the raw machine code of the executable file. There would be a separate dependency table for each dynamic library the executable depended on. Upon loading the library, if there is a dependency table for that library, a special function would be called which goes through every address in the table, and at the memory indicated by the address it would add on the offset at which the dynamic library was loaded. There would also be a dependency table for the executable itself, allowing the OS to load it at any address, and have it still function. The dynamic libraries themselves would count as executable files and so the same would apply to them (and any dependencies they had).