Architectural ideas behind osFree

Compatibility with OS/2 (Intel)

osFree combines OS/2 (Intel) ideas with those from OS/2 (PowerPC). It must run programs using OS/2 API, both console and PM ones. So, on Intel architecture, it must be binary compatible with IBM's OS/2. But on other architectures it will have different ABI [1)], but the same portable API [2)].

Also, like it was in OS/2 (PowerPC), the binary compatibility with OS/2 (Intel) is possible (even on non-Intel architectures). It includes the translator of PowerPC instructions to Intel ones. The Instruction Set Translator (IST) is a special DLL. It could be made on the base of QEMU binary translation.

LX executable format is unlikely to be used on other architectures, because it is tied to the Intel one. So, most possibly, the ELF format will be used. It is cross-platform, extensible and simple. It supports both a 32-bit and a 64-bit variant. IBM Microkernel used ELF too. And more, IBM used an extended ELF format with special binary sections semantics. It supported inline resources, imports and exports, for example. So, it was possible to use UNIX-style shared objects, as well as OS/2-style DLL's.

So, the userland must be compatible with OS/2 (Intel). This includes some support of 16-bit applications. Although there is a very little of pure 16-bits apps (like cmd.exe and hiew.exe), but some 16-bit API's are widely used today. For example, Kbd/Mou/Vio API's (I.e., Console API) are still 16-bits in OS/2 (Intel), though on OS/2 (PowerPC) they are made 32-bits, because binary compatibility makes no sense on other arch's (only source one makes sense).

TODO: Thunking. Getting rid of thunks in 32-bits executables calling the 16-bit API's.

Kernel-related parts of OS/2

The most of 'kernel' area was decided to be moved to userland of L4 microkernel. So, some services, commonly meant as kernel parts, like filesystems, drivers, memory management, thread scheduling etc. are moved outside the kernel. So, they are not significantly differ from usual applications, and confined inside their address spaces.

Personality-Neutral Services

OS frameworks: l4env, l4re and Genode

Device Driver Environment

The OS/2 personality main server (or simply "OS/2 Server")

Filesystem server

Installlable file systems (IFS)

Filesystem router

IFS helpers

Exec server

Virtual Memory Arenas management

Instalable eXecutable Formats (IXF)

Shared memory management

Multiple Virtual Machines (MVM) server

A historical note

The term “MVM” goes from OS/2 (PowerPC). It replaces the term “MVDM” (Multiple Virtual **DOS** Machines). The PowerPC platform could not so easily emulate the Intel processor real mode, so it required more emulation. Though, we must say that it worked just fine, fast, and theoretically may be used not for DOS only. (IBM had plans to provide a binary compatibility with OS/2 (Intel), on the base of the same component which was used to run DOS Apps).

Virtual Device Drivers (VDD's)

Virtual Machine Monitor

MVM server

The DOS emulation kernel (doskrnl)

Instruction Set Translator (IST)

VM86 on Intel, and Hardware-assisted virtualization

Microkernels as Hypervisors

Other OS's personalities

"The one Ring to Bind Them all" (OS/2 as an integration platform for different types of applications)

The current OS/2 personality prototype

osFree PM

Graphical Program Interface (GPI)

Graphics Runtime Environment (OS/2 PM GRE) and Presentation Drivers

The GRADD model

Video Protected Mode Interface (PMI)

VIO/KBD/MOU (Console API)

VIO and BVH's (Base Video Handlers)

Framebuffer Interface

Virtual Framebuffer over GRADD driver

BVH and PM GRE on top of a Framebuffer interface

The current osFree PM prototype

[1)]
Application Binary Interface
[2)]
Application Program Interface

From:
https://osfree.org/doku/ - **osFree wiki**

Permanent link:
https://osfree.org/doku/doku.php?id=en:docs:general:architecture&rev=1402008317

Last update: **2014/06/05 22:45**