MorphOS in Detail

By Nicholas Blachford

Contents

This document was written to describe MorphOS, how it came about, its workings, its current status and future plans. Special thanks go to bplan and the MorphOS developers for the great deal of assistance given.

Introduction

MorphOS is a new Operating System for PowerPC RISC microprocessors which runs on the Pegasos computer and PowerUP (CyberstormPPC, BlizzardPPC) expansion cards for the Amiga. It has also been tested on the Mai Logic Teron CX and Teron PX evaluation boards. In the future MorphOS will also run on other Open Firmware based PowerPC machines and possibly some PowerPC based Apple Macintosh computers as well.

It currently runs on 603e, 604e, 750 and 7400 (Original G4) PowerPC processors.

MorphOS is a well named combination of the old and the new. It originally started in 1995 with a plan to migrate Amiga to PowerPC but eventually 'morphed' into an entire Operating System in it's own right which includes Amiga compatibility. In the future it will change again becoming a truly modern, highly advanced operating system yet retaining compatibility with existing applications through it's system of OS boxes.

In The Beginning - The History of MorphOS

To add some perspective and for completeness, before going into detail, this section describes how the MorphOS project started and it's subsequent history.

The Amiga started on its long arduous path to the PowerPC processor in 1995. While there have been and still are other projects to do this, with MorphOS 1.0 the original project is finally being completed.

In September 1995 there was an agreement between Amiga Technologies and the German company phase5 to develop a migration path for the Amiga to move to the PowerPC from the Motorola 680x0 (68K) series processors which were coming to the end of their life, this was the genesis of the MorphOS project.

Unfortunately this agreement was short-lived as Amiga Technologies' parent company Escom went bankrupt. However the project did not die as phase5 decided to go ahead and continue with the migration process. This lead to phase5 launching in 1997 a series of PowerUP add-on cards for the Amiga providing RISC based acceleration for the first time. Programs ran on both the 68K and some functions were accelerated through the use of the PowerPC processor and a PPC native library.

There were many plans made and agreements between phase5 and other companies but as the numerous twists and turns in the Amiga story occurred companies went out of business and none of these plans came to fruition, eventually with the decline in the Amiga market phase5 themselves went bankrupt. Later a new company, bplan was formed and the project was picked up again but this time it was to be completed as an Operating System in it's own right without any components from the original. In 2000 as it moved towards completion, MorphOS was released as a public beta for PowerUP card owners.

The 2002 public MorphOS 1.0 release on the PowerPC based Pegasos represents the final chapter in the 7 year long transition to PowerPC from the original 68K based propriety hardware. MorphOS does not require the original hardware or include any of the original system software but it does very much include the same spirit and feeling - a computer which is fast and fun to use!

The Structure of MorphOS

The MorphOS System is based around the minimalist Quark microkernel.
On top of the kernel are currently two "Boxes" the first, currently in the final stages of development is the A-Box, the second box currently in development is the significantly more advanced Q-Box.

At the time of writing (November 2002) most development has focused on the A-Box but considerable design work has been done on the Q-Box.

The A-Box can run Amiga RTG (Re-Targetable Graphics) applications as it includes a complete PowerPC native clean-room reimplementation of version 3.1 of the Amigas' Operating System (herein AOS) and a JIT (Just In Time compiler) based 68K emulator.

The Q-Box on the other hand shall be new and will require it's own applications, but by including Amiga compatibility through the A-Box, MorphOS is able to start with an existing, mature application base while the Q-Box is in development.

It is also possible that in the future other boxes may be added allowing compatibility with applications from other operating systems. Possibilities here are to add Unix / Linux and BeOS application compatibility. It should however be noted that these are only possibilities at this point and various issues have to be considered before adding compatibility for these systems.

The A-Box

The original purpose of MorphOS was to keep the Amiga alive by providing an upgrade path to modern hardware.

Users can already use emulators to run original applications on other platforms but these work by emulating the 68K processor and the custom chips. While this emulation is complete and accurate, a great deal of potential performance is lost in doing all the emulation.

While MorphOS has moved on from what it was originally built for it still retains compatibility with AOS 3.1. However it uses a different technique which allows software to take full advantage of modern hardware. The A-Box includes a clean-room implementation of the AOS 3.1 API created using publicly available documentation. This has been written and compiled on the PowerPC so it is nearly 100% native. There is also a 68K emulator for existing applications, the performance of which is boosted significantly by a JIT (Just In Time compiler) which converts 68K code into native PowerPC code then caches it largely removing the overhead emulation usually incurs. In addition to being native, the A-Box is faster than the original at running applications due to different techniques and being used within the system. This provides a further boost in system performance.

For maximum performance however it is a simple process to compile code natively to the PPC. This allows Amiga applications to take full advantage of the PowerPC CPUs' performance.

One important point to make however is that there is no emulation of the Amiga custom chips. Consequently MorphOS cannot itself run applications which require them. This is not as much of a problem as one might imagine since applications from AOS 2.0 onwards have been able to use API calls which are not hardware dependant. If a user wants to run applications which require the custom chips (e.g. many games), they can use UAE (Ubiquitous Amiga Emulator), this provides emulation of the full Amiga hardware and runs on various platforms including MorphOS.

AOS originally ran on what are now obsolete systems with only a fraction of the computing power available in today's modern microprocessors. Amiga speeds were quoted in MIPS (millions of instructions per second), the MIPS rating for a 1Ghz IBM G3 is over 2,000 times higher than the original 68000 based Amiga. Despite this the original Amigas responsiveness was - and still is - very high. Consequently, given that the A-Box provides a more efficient PPC native implementation, it's responsiveness will appear surprisingly fast to any PC owner especially given the relatively low clock speed (600MHz) of the initial systems.

File Systems Support
MorphOS supports a number of File Systems and others can be supported via plug-ins:
OFS (Original File System)
FFS (Fast File System)
FFS 2 (Fast File System 2)
PFS 3 (Professional File System 3)
SFS (Smart File System)

Chill Out - The Ambient Desktop
The MorphOS desktop replacement for Workbench is called Ambient and being part of the A-Box feels like the environment it emulates. It is able however to take advantage of the changes and advancements in the graphics system so unlike the Amigas' Workbench it supports 24bit displays and blending. Displaying the contents of a window while it's moving is another feature provided by the graphics system used by Ambient. MUI (Magic User Interface) is also incorporated making Ambient highly customizable.

Other Graphical Enhancements
Unlike the original, the graphics system is now fully abstracted away from the hardware, it has also been significantly enhanced with many new features such as transparency and overlays. Accelerated 3D graphics are implemented using the Rave3D API and a wrapper for the Warp3D API is in development, OpenGL compatibility is also in the pipeline.

The Audio system has also been improved with the addition of AHI, this is a retarget table API for Audio so Applications can take advantage of systems with audio hardware better than the original (these days, all of them).

When launched AOS was a highly advanced operating system, it included pre-emptive multitasking in 1985 - a feature not added to other desktop Operating Systems (i.e. to Windows or MacOS) until much later. Compared to modern Operating Systems however, the Amiga has it's fair share of limitations, i.e. there is no memory protection or virtual memory present as standard.

Because it includes a faithful re-implementation of the AOS 3.1 API specification the A-Box design was largely fixed and could not be made radically different. Consequently the A-Box has many of the same basic limitations present in the original. Some of these have been worked around with extensions but in some cases the limitations are either too difficult or impossible to remove without completely breaking compatibility with applications. These however are limitations of the A-Box, not MorphOS as a whole, the Q-Box shall not have these problems.

Q - The Future of MorphOS

The Q-Box

Up to now the development has been concentrated on the A-Box. In the future this focus shall switch to Q.

Q shall consist of an enhanced Quark kernel, a set of servers to provide functionality and the Q-Box in which applications run.
The Quark Kernel itself is very small providing a hardware abstraction layer, drivers, memory management and message passing. Most of the real work will be done in the servers, i.e. Networking, File System, GUI, Media, Security, 2D/3D Graphics etc.

Applications shall run in the Q-Box and make API calls via a message passing system. The API shall not be locked to any single programming language so application developers will not be required to learn a new language. The message passing is also extremely fast, instead of transferring the data in the message, the memory location of the message is remapped to the application where the message is due. Small messages are directly copied as this method is faster for very small amounts of data.

Scalability and Other Bits

The message passing system used within MorphOS means the system can be highly scalable. The components sending and receiving messages can be running on different processors or even physically different computers and the messages will still get there. A windowing system over a network (a-la The X Windowing System) could be implemented in this manner but the result is likely to be a great deal more responsive. An X Windows compatible wrapper is planned so X Windows applications can be ported but the resulting speed on the desktop will be very different from those using current X Windows implementations.

The distributed capability of Q messaging means clustering will be possible without having to rewrite the entire OS. Applications requiring massive numbers of processors will be relatively simple to write under Q. One requirement of large scale applications is a large memory footprint, workstations and large servers have been 64 bit for many years now but this is yet to reach consumer level Operating Systems. Q was designed with 64 bit capability in mind from the start, it shall not be a bolt-on to an existing 32 bit OS. Applications will, from the start have access to address ranges much, much larger than existing desktop Operating Systems.

Microkernel Vs Macro Kernel

A common problem encountered in the development of microkernel Operating Systems is speed. This is due to the CPU having to context switch back and forth between the kernel and user processes, context switching is expensive in terms of computing power. The consequence of this has been that many Operating Systems have switched from their original microkernel roots and become closer to a macrokernel by moving functionality into the kernel, i.e. Microsoft moved graphics into the Windows NT kernel, Be moved networking inside, Linux began as a macrokernel so includes everything. This technique provides a speed boost but at the cost of stability and security since different kernel tasks can potentially overwrite one another's memory.

Given the above, one might wonder why Q can be based on a microkernel (strictly speaking it's only "microkernel like") and still expected to perform well. The answer to this lies in the fact that MorphOS runs on PowerPC and not x86 CPUs. It is a problem with the x86 architecture that causes context switches to be computationally expensive. Context switching on the PowerPC is in the region of 10 times faster, similar in speed to a subroutine call. This means PowerPC Operating Systems can use a microkernel architecture with all it's advantages yet without the cost of slow context switches. There are no plans for an x86 version of MorphOS, if this changes there will no doubt be internal changes to accommodate the different processor architecture.

Transition

At time of writing the Quark kernel exists but is incomplete. The rest of Q is still only at the planning stage, the large details have been worked out but the fine details have yet to be filled in. Drivers are for instance included in the A-Box where they can get to the hardware directly. In the future the drivers shall be moved into the Quark kernel where they will be independent of the A-Box and indeed of any other boxes, this has the advantage of making them usable by all the boxes. This move has been pre-planned so drivers were written with it in mind and the move shall be a relatively trivial operation.

Vapour Where?

Almost all the information here about the Q-Box is in the future and this could lead to accusations of us generating vapourware. We are only setting out our goals here and how we plan to achieve them. However as is normal in the world of computing things change so this is all subject to change. Development of complex systems always takes a great deal of time, we are looking at ways of getting the system into the market as soon as possible but the final product will of course take time.

The future will expand on the work already done and provide MorphOS users with a truly modern Operating System designed and built by now long experienced developers who know the benefits and advantages of the past but also have the knowledge and experience of what not to do and the paths not to follow.

Further Information

For details of MorphOS news and links see:
www.morphos-news.de

For more details of the Pegasos main board see:
www.pegasosppc.com