



ZEBOR TECHNOLOGY'S WHITE PAPER ON ZDOS®

Until now, developers of application platforms have been forced to make trade-offs between performance, ease of implementation, cost, and other factors. The result has often been expensive compromised solutions, far from the initial idea that license-free software should make a project quick and inexpensive.

New solutions and new ways of thinking are needed to lift developers from what we consider their three main burdens:

1) Slow and heavy application platforms have become a standard, and developers often find themselves forced to try to accept their evident drawbacks, while still expected to develop superior solutions.

2) What's "free" out there has over and over again proven to actually be more expensive, due to several unexpected costs. When a project is considered as a whole, and not just the very small cost or non-cost for software, the end-result often differs hugely from the initial budget.

3) Heavy and inefficient software development platforms create an automatic need for heavy and therefore expensive hardware. While great focus is put on whether a "free" or a proprietary software development platform is used or not, very little focus is put on where the real money can be saved – for example on the hardware.

This article aims to discuss these topics above, and shed some light on the real cost-drivers in application development, with the goal of illustrating paths forward which do not involve compromises, delivers expected results, on time - while cutting those costs that really matter.

1) Slow and chunky application platforms have become standard

Indeed new technologies and more efficient methods of doing things should be welcomed and embraced if proven to actually solve problems. GNU and Linux are obvious examples of greatly successful new software platforms that opened up for alternatives to Microsoft's near monopolistic position.

But with its origins in as far back as 1983 and 1991 respectively, even GNU and Linux must be considered old-timers in the game. Today, Linux is by many considered to be a patch-work of layer upon layer of old code, with bugs, errors and inefficiency incorporated into the very core. Also, the problem of "forking" is becoming increasingly evident – meaning that there is a lack of compatibility not only between different releases of Linux – but also between different versions of a particular release.

2. What's "free" out there has over and over again proven to actually be very expensive, due to several unexpected costs.

Blinded by the promise of "free" software many non-technical decision-makers often rush into a decision to develop on Linux thinking that choosing a license-free development platform will save money. In actual fact many studies prove the opposite. The only migration that seems to pay off on a corporate level are those from Unix to Linux. But when looking at the TCO (total cost of ownership) and all factors considered, a choice to develop a Linux-based solution often proves expensive. Main factors here are:

a. Heavy software requires heavy hardware. This means the BOM (Bill of Materials) will add up to heavy costs as expensive and for the task unnecessarily heavy hardware must be used. Consider for example the difference in cost per unit of a product when a 20 USD processor can be used instead of a 150 USD processor of higher capacity.

b. Although Linux has the advantage of being a wide-spread platform of choice for many developers, it is actually not a very easily implemented platform. This simply means that many software-engineers have to work many hours in order to develop new applications – creating costs.

c. Time to market – The sooner a completed product can reach the market, the sooner it can begin generating pay-back, and also speed ahead of the competition. But due to a combination of factors, such as the many versions and the patch-work-nature of most free software, projects more often than not exceed time-plans and budgets grossly – both increasing costs for the project itself, but more seriously delays the payback, and leaves space for competitors to get ahead.

d. Ease of copying – Because of the extensive knowledge of for example Linux, and its openness, it is far too often an easy job for competitors to reverse-engineer a complete product, copying a product, and even bring it to the market at a cheaper price – because they bare almost no development costs.

3. Far too often today, even the simplest of embedded systems come equipped with powerful processors and everything that goes with it. It may be that an embedded product is basically only going to be used to perform one task – let's say a paying-machine for parking-tickets – but since the application is built on Linux, it needs a full-fledged computer inside. This not only creates costs because the hardware itself is expensive, it also automatically generates other costs when fans are needed, or even air-conditioners – and

all of these consume electricity, resulting in both monetary and environmental costs.

ZDOS® is a light, highly efficient, environmentally friendly operating system

ZDOS® was developed to provide solutions to the problems listed above. The basis of ZDOS® was developed while special applications were built to handle important, fundamental IT-infrastructure roles in the Swedish society on a mass-level. Among these applications were the Stockholm Stock-Exchange information system SIX, the Swedish ATM system for the major bank Nordea, and the security/encryption system for the Swedish Military. Obviously, the demands on speed, reliability and scalability in these projects were as high as can be. Interestingly, none of the projects above were built on any existing software. Instead the developers built these applications from scratch, in Assembler. After having concluded these, the core of what was to become ZDOS® was born. This was then finalized into a useable software package, and is now available as ZDOS®.

Below is a presentation of how ZDOS® provides solutions to the problems listed above:

1. Slow and chunky application platforms have become standard

ZDOS® is not the standard today – far from it as too few people know about it. This is likely to change. What's important is that ZDOS® is not slow and chunky. Instead, ZDOS® is very fast and very small in size. The ZDOS® core is only 132K including the commando interpreter, while the Linux core is 1,7MB. For example, tests have proven that a platform running on ZDOS® and a 0,8GHz processor can deliver the same or better results than any other system running a 2,0GHz Core Duo processor. ZDOS® can also deliver the same or better results on a 3,5W processor, than Linux on a 55W processor – greatly reducing the costs for electricity, and completely eliminating the needs for fans and other cooling. Again, ZDOS® itself is an incredibly small Operating System – just 132kb in size.

2. What's "free" out there has over and over again proven to actually be very expensive, due to several unexpected costs.

ZDOS® is not free, instead it is acquired on a low-cost license-basis. This means that there is only one relevant version of ZDOS® – with backwards compatibility to previous versions, and with full support from the Zebor team. Instead of being free, ZDOS® saves costs by demanding much less expensive hardware, needing less development time,

fewer engineers – and getting the finalized product quicker to the market.

3) Heavy and inefficient software development platforms create an automatic need for heavy and therefore expensive hardware.

Again, ZDOS® is an incredibly light operating system with very low demands on processing power and resources. During the last decades there has been virtually no focus on increasing the effectiveness of the software. Instead all the focus has been on creating heavier and stronger hardware – to support the ever-increasing demands of the software. With the invention of dual core processors etc, hardware manufacturers are struggling to make these heavy soft wares run. In the 16 bit DOS-world – many developers find themselves locked by the 640K memory limit – something that ZDOS® has solved, being a 32-bit real-time operating system. Also, it is important to point out that the Zebor philosophy is to start from the core and build from there, on a mean and lean basis. Far too often engineers find themselves having to start from a much too big OS such as Linux, and then try to get rid of all the unnecessary parts to try to make it work for the specific application at hand.

Conclusion – cut costs where they matter

In conclusion, the 32-bit real-time operating system ZDOS® allows you to do more with less. It's a small and light operating system requiring a minimum of hardware. There is no learning-curve in beginning to use it, as applications can be programmed in standard C or C++.

Using ZDOS® eliminates the previous forced trade-offs between performance, ease of implementation, cost, and other factors, being fast and efficient, easy to use and inexpensive. Thus ZDOS® helps cut those costs that really do make a difference:

- Hardware costs – Being able to use less advanced hardware brings down cost per unit dramatically.
- Implementation-time – Being able to use fewer engineers, and fewer hours with these engineers is a huge cost-cutter. To get up to speed with ZDOS® is quick, and the tools are so straight forward to learn and use that applications are quickly and easily created. For example, the same run time library functions are used on both application and kernel driver levels.
- Time-to-market – Perhaps the most important point – Bringing your product quicker to the market helps you beat the competition, become the standard, generate quicker payback – and allowing you to move on to the next product before anyone else.

Examples of areas of usage

- Digital set-top boxes
- Advanced interactive TVs
- Mobile phones
- Hand-helds
- Industrial robots
- Home appliances
- Tools for measuring current, signal strength etc
- Network equipment, such as routers, switchers, etc
- In cars, planes, trains
- Industrial tools
- Mountain-drills
- Security systems, encryptions
- Desk-top computers if used with a web-OS
- Pagers
- Web-pads
- And more...

Technical Specifications

- Low demands on memory: ZBIOS®, ZDOS®, TCP/IP and applications fit and can boot/load from one 512KB flash memory, for example on-board memory of motherboard
- Real-time 32-bit operating system
- Runs on any 32-bit x86 platform with a 486 or later architecture
- Completely written in assembly language
- 20-100 times faster code than any other platform
- Highly efficient
- Low hardware demands (good performance on a 100MHz 486 CPU)
- Low power-demands (performs well on a CPU using only 0,5W)
- Rich, easy to use run time library with support for every single function
- TCP/IP network protocol stack written entirely in assembly language for performance
- Native and integrated support for parallel processor architectures (SMP)
- Minimum memory requirement 4MB
- Small footprint
- Preemptive multitasking
- 32 priority levels
- Almost instantly operational