THEOS SOFTWARE CORPORATION

WHITE PAPER

October 2003

Reasons to Use the THEOS Platform

Summary

- Ease Of Use & Maintenance
- Ease Of Development
- Virus and Worm Proof
- Hacker Resistant
- Crash Resistant
- Reliable File & Database
- Fast & Easy Networking

Overview

This overview summarizes some of the major reasons to adopt and implement the THEOS operating system. The overview is intended to be used by information technology staff at systems integrators, software houses, and end-user organizations. The more you get to know about THEOS, the more you will wonder why you haven't heard more about this remarkable operating system before. The simple answer is that management at the firm has historically emphasized software engineering excellence, has kept a low profile, hasn't invested in significant marketing efforts, and has relied upon VARs to perform most sales and marketing. As a result, what the public has seen of THEOS has to date largely been the applications marketed by VARs. At this point in time, THEOS supports thousands of mature applications, hundreds of thousands of installations worldwide, and over a million end-users.

1. Ease of Use and Maintenance

THEOS was designed to be a personal computer multi-user multi-tasking operating system that could be used by small business people without the need for in-house technical support. Depending on how developers structure applications, users can be entirely locked out of the operating system, instead operating only in menus which provide application-specific options. Many tasks that ordinarily require the support of technical staff, such as backing-up application files, can be handled in a truly user-transparent way by an application running under THEOS.

Many THEOS end-user organizations have no technical person on their staff. They are instead remotely serviced by a VAR who occasionally checks up on the system via a secure connection established via the Internet or a dial-up line. Installation of the operating system typically takes ten minutes. The process is so straightforward that VARs often talk non-technical end-user staff through the process via telephone. System start-up can be achieved in under ten seconds while system shut-down can be performed in under five seconds.

THEOS has a genuine commitment to full backwards compatibility of its operating system. This means that new upgrades to the operating system do not require additional application coding in order to take advantage of new operating system features. This commitment means that many incompatibilities that plague other operating systems are avoided, thus helping to ensure a stable and easy-to-use computing environment for end-users.

Although it does support all commonly-used operating system features and functions, THEOS does not support a very large number of unnecessary features and functions as Windows does. Many of these unnecessary features and functions end up causing Windows to become very large (“bloatware”). More importantly, many of these features and functions significantly increase the complexity of Windows, thus making it more difficult to maintain. A good number of VARs who support the same application under both Windows and THEOS have dropped support for Windows because Windows was too time consuming and expensive to support. Complexity also makes Windows more difficult to comprehensively test, and this means that Windows is more likely to have undetected security problems. The consistent internal design and the on-going discipline of a small number of developers at THEOS allow the operating system to be elegantly simple. This in turn makes THEOS easier to maintain, significantly more secure, and significantly less expensive to use.

2. Ease Of Development

Because THEOS is highly-structured, internally-consistent, and straightforward, developers find that putting together applications takes much less time with THEOS than it does with other operating systems. THEOS supports applications developed in BASIC as well as C language. Developers who have written in other languages report that it takes only a single day of training to get to the point where they can productively write code for THEOS.

Just because THEOS has been around for 25+ years doesn't mean that it doesn't support all the latest hardware interfaces and Internet communications protocols. For example, THEOS supports standards such as DHCP, DNS, HTTP, HTTPS, FTP, TFTP, POP3, SMTP, Telnet, SMB/NetBios, and many others. These standards allow THEOS developers to quickly build code to meet a wide variety of objectives. These objectives include: supporting network management, supporting remote system maintenance, functioning as a web server, acting as an email server, acting as an email post office, feeding firmware to embedded devices, supporting file transfers, and sharing system resources (like files) with other operating systems. Also notable is the fact that THEOS internally develops all of its drivers, thus helping to ensure another layer of compatibility and consistency that is often missing with other operating systems like Windows.

THEOS supports graphical user interfaces as well as text-based user interfaces. It can be used on a server with thin clients, and even traditional ASCII dumb terminals can be employed as client machines. MS-Windows connectivity is provided, and an emulation system allows all real-mode desktop Windows applications to be supported as clients to a THEOS-based server. THEOS can also serve as a secure client, connecting to a THEOS server. THEOS is designed to run with single Intel processors, and virtually any computer system based on Intel's personal computer systems can run THEOS. Any machine which uses standard interfaces can be a client communicating with a THEOS server. Applications can be web-based so that they have the same look-and-feel of a standard browser to end-users. The small space required for the operating system (100MB of disk space and 32MB RAM to execute) makes it a viable alternative for special purpose devices as well as small portable machines.

The THEOS operating system uses an active data dictionary, and this means that data input can be screened according to standardized cross-application tests before it is accepted by a variety of different applications. A built-in ISAM database management system provides a further level of integration between the operating system and the database management system. This integration also makes life easier for developers because they need to consult only one source of documentation, rather than the customary two sources. Likewise, incompatibilities between these two types of code are eliminated because the THEOS code all comes from one organization.

Unlike operating systems such as UNIX, there is only one current version of THEOS. There are no variants or derivative versions of THEOS. All code for THEOS has been developed internally instead of by third parties. All this means that interfaces to THEOS are cleaner, more predictable, and considerably less complicated than interfaces with many other operating systems. Because there has only been one organization developing THEOS, there are no legal questions about the use of THEOS code once a system has been licensed -- a situation distinctly different from that with LINUX.

3. Worm And Virus Proof

THEOS was designed much like a mainframe in that it can be configured so that it requires all executing code to have predefined privileges. Executable code introduced from external sources (worms, viruses, and other malware) will then not execute because it has not expressly been granted permission to do so. End-users who may be handling files containing worms or viruses do not have privileges to modify the list of programs that they can execute, so worms, viruses, and other malware will be contained within the domain of the end-user's restricted privileges. This restriction on the execution of externally-supplied code goes beyond attachments that come with email, and beyond background processes spawned by web surfing, it also applies to macros which come with spreadsheets and word processing packages. So even if a virus were designed with the unique executable format found in THEOS, if it was externally introduced, it would not have privileges to execute, and thus would be harmless.

To further ensure that THEOS boxes are not damaged by worms, viruses, and other malware, the THEOS operating system uses Intel's 32 bit segmented programming model. This means that a separation between executable code and data is always maintained. Thus spreadsheets and other data files are always simply data, and they cannot unexpectedly turn out to be executables. The THEOS file system furthermore has file type conventions which are defined at compile time, and which cannot be changed at run time. This further ensures that what appears to be data is not in fact a malicious piece of executable code.

Further making THEOS immune to worms and viruses, it is notable that Windows executable code will not run under THEOS. THEOS has a unique internal memory addressing scheme, a different stack handling process, and other unique features which mean that assembled code for Windows or other operating systems will not execute under THEOS. So even if an authorized user of a THEOS system were to place a Windows virus on a THEOS machine, perhaps using a THEOS box as a mail server, it would still have no effect on the THEOS machine.

THEOS client machines are by default locked down. This prevents end-users from changing system configurations. Of course it is still possible for the developer to allow end-users to change their own local hardware configurations, such as the printer they access. This lock-down approach prevents users from executing downloaded unauthorized software, such as software which exchanges music files. End-users are not given access to the command interpreter, and access to the command interpreter is required to execute anything other than the list of authorized applications expressly granted by a systems manager to an involved end-user. This default desktop lock-down also means that users cannot change operating system files in a way which would open THEOS up to external compromise. The THEOS configuration that is installed by a system manager is the version that end-users continue to employ. This helps to ensure that externally introduced code, such as worms and viruses, cannot disrupt the end-user processing environment.

Because a physical or virtual dongle is required for every THEOS machine, the operating system code checks that this is an authorized machine before it permits the execution of certain internal routines. The developer can also prevent the operating system from booting unless the correct serial number, which is found in the dongle, has been provided. This feature can provide still another mechanism helping to ensure that only authorized code executes on systems running THEOS, as well as ensuring that software piracy does not take place. This dongle can furthermore provide a mechanism to prevent unauthorized third parties from setting up a secret THEOS system, thus thwarting their efforts to surreptitiously develop and test THEOS attack methods.

The ability to detect virus- and worm-like behavior allows THEOS to block certain transmissions which might infect other machines, thus preventing a THEOS box from becoming a conduit of these attacks even though it will not be victimized thereby. Because THEOS is internally different from Windows, it is well suited for those applications and network services that absolutely must continue to be available, such as network logging. The

With THEOS, there is no need for an anti-virus system, and likewise there is no need for updates to anti-virus systems. Organizations using THEOS can thus avoid the time consuming and costly efforts associated with frequent updates to protect against the latest batch of viruses. Organizations using THEOS can also avoid time spent fixing problems caused by viruses and worms, which are collectively now the most frequently encountered computer security problem around the world. For the reasons mentioned in the prior paragraphs, the ongoing and increasingly-escalating war between virus writers and the anti-virus system developers can also be avoided altogether.

4. Hacker Resistant

THEOS is aware of no incidence in which its operating system has ever been hacked. A quick review of the vulnerability statistics compiled by the Computer Emergency Response Team at Carnegie-Mellon University reveals no reported vulnerabilities for the THEOS operating system. This is in marked contrast to other operating systems which have hundreds of reported vulnerabilities, and many thousands of reported compromises (see www.cert.org).

All privileges are denied under THEOS unless expressly permitted, a security philosophy distinctly different from other personal computer operating systems. A number of security-jeopardizing actions that other operating systems allow are not possible under THEOS. For example, users are by default required to change their password when they first log-in with a new user-ID (also called expired passwords). This means that only the end-user will know his or her password. Likewise, systems administrators cannot set up shared user-IDs -- instead, each user must have their own unique user-ID. These and other features have been constructed to ensure that logs will definitively reveal the actions of each individual user.

Default access control is provided by a robust user-ID and fixed password system. This subsystem includes a variety of password management functions such as the ability to screen user-chosen passwords to make sure they are at least a certain length, are made up of both numbers and letters, have not recently been chosen by the involved user, are not in the dictionary, and are not equivalent to a user-ID. In addition, user-IDs that have been dormant for a certain period of time can have their privileges automatically revoked to prevent intruders from quietly using these accounts for unauthorized activity. The password subsystem also includes support for encrypted sessions so that fixed passwords cannot be intercepted when in transit. Also included are standard password protection routines such as a timeout after a certain number of failed log-in attempts in order to prevent password guessing attacks. Beyond the ability to support two layers of passwords, access to a THEOS box can also be restricted by originating client machine IP address. Support for high-speed hardware encryption applicable to remote sessions is also provided. Where a more stringent user authentication system is needed, a THEOS-based fingerprint biometric access control system is currently available.

Unlike some other operating systems such as UNIX, applications running under THEOS do not need system manager privileges in order to execute. This means that if a hacker were to be able to gain the privileges allocated to an application, that the hacker would not be able to take over the entire system. To further protect against unauthorized executables, parts of the THEOS operating system, such as the web server, can be configured so that access controls force all code to be located in certain protected directories (such as a CGI bin) in order to be executed. Because a hacker who took over a user account has no access to these directories, he or she could not introduce unauthorized executable code that would run. This is because system manager privileges are always required in order to configure and/or install soft-

A robust access control system allows developers to define many different types of access. The standard read, write, execute type of privileges can be defined for internal system resources like files and applications. It is easy to implement separation of duties and/or dual control (where two people are simultaneously required to perform a certain action). It is also easy to configure the system so that the system manager account is segmented into several different types of accounts, thus reducing the probability that any one person in a position of computer-related trust will exploit their privileges. The system manager account can for example be segmented so that one individual can install patches, another can display but not change internals files, and another can run utilities such as a defragmentation routine.

Since it's early days, access to THEOS source code has been restricted to a very small number of developers (according to policy, VARs do not have access to source). Documentation is also restricted to those who have a need to know. The unusual internal structure of THEOS executable code further makes it unlikely that outsiders intent on breaking into a THEOS system would be able to figure out how to compromise the system.

If management suspects that one of its users is engaging in unauthorized or abusive behavior, a THEOS routine which allows management to peek at a user's real-time session can be deployed. Snapshots of the session can be saved for further investigation, or for building a documented case for disciplinary action or termination. VARs can use these same routines to check-up on end-users and to perform help desk type support for remote users.

5. Crash Resistant

Because THEOS has been a commercial product for over 25 years, the development staff has had plenty of time to polish the code. This has allowed development staff to correct bugs and glitches, and to make the OS truly stable and reliable. Servers at customer sites are regularly reported to be running for many months without the need for restart, and for many years without the need for replacement. Some customers are still running THEOS systems on 286 machines, systems that were built around 1990. A recent survey of over 1000 THEOS users reported that in an average of nine years of operation, they had no virus attacks, no hacker attacks, and only eight crashes, and all the crashes were traceable to hardware failures.

Because THEOS is so stable, upgrades and new releases come out quarterly, and many users don't bother to implement these changes. The whole "patch or perish" approach to upgrades and new releases is entirely unnecessary when using THEOS. With other operating systems, particularly Windows, incompatibilities and other problems occasioned by patches and upgrades have also been a major source of downtime and crashes. This source of trouble as well as the need to test a large number of patches and fixes is eliminated when you go with THEOS.

THEOS has many built-in stability and internal integrity checks. For example, operating system files are automatically subjected to cyclic redundancy checks (CRCs) to ensure that they have not been corrupted or tampered with. Support for a variety of high-availability systems is also provided. For example, the operating system supports RAID 1-7 disk arrays, disk mirroring through a controller, an uninterruptible power supply (UPS) monitor, automatic reboot, and hot-pluggable boards (which allows processing to continue while circuit boards are changed). The operating system also uses segmented memory within Intel chips which means that certain types of buffer overflows are prevented, and also that one application can crash, but other applications can continue to be supported by the operating system.

Other THEOS features help to preserve a stable internal environment. These include a system log that does not overwrite itself, instead creating additional log files. Other routines automatically synchronize internal machine clock times across several THEOS boxes with an authoritative external time source. After a inappropriate shut-down (for example one caused by a power outage), THEOS automatically runs internal utilities which check the file system, and then, if necessary, automatically corrects many types of errors such as missing or inaccurate pointers. Also included are routines which detect a denial of service attack in progress, which filter inbound packets so that service can continue. The ability to support DNS black lists and spamcop, as well as permit connection only to named email servers, further protects a THEOS system against spam and other threats that might jeopardize its continued availability.

6. Reliable File and Database

The very reliable THEOS ISAM (Indexed Sequential Access Method) technology is the number one reason THEOS has been around over 25 years and actively supporting thousands of mature applications around the globe. THEOS ISAM is an integral part of the OS file system and has proved to be a stable and powerful data management tool for thousands of THEOS-based applications and used by millions of end-dusers.

Basically there are three types of Data Base files supported natively by THEOS: Index, Direct and Keyed.

These three database file types essentially consist of a key and a record, which can be defined, in just one sentence, by a programmer to fit any structure desired. This basic, well thought out simplicity is the inherent strength of the THEOS file structure for all applications running on THEOS today.

In addition to THEOS ISAM, we now have TDB, the THEOS Data Base Server, which is a DBMS that allows you to define your existing flat & ISAM files into a well organized Data Base. Once the definitions are completed, your data is ready to be queried with SQL or ODBC from Windows world.

However, after 25 years of hands on experience, THEOS contends that a mission critical application does not necessarily need a complete Data Base Engine with all the extraneous overhead required to support the applications. For over 25 years, hundreds of THEOS developers have been able to manage data in a very transparent and direct way using THEOS ISAM. And they have reported that their data, no matter how big or complex the files were, has proven to be very stable, without the data corruption or misplaced data

7. Fast and Easy Networking

THEOS Networking capabilities are embedded into the Operating System providing a faster implementation. A complete TCP/IP Stack of protocols makes THEOS Networking easy and straight forward. Here are the most common protocols and services supported by THEOS:

- FTP, File Transfer Protocol, allows file exchanging between systems. THEOS implements both client and server.
- TFTP, Trivial File Transfer Protocol, allows uploading firmware to devices. THEOS implements both client and server.

- HTTP(S), Hyper Text Transfer Protocol, secured with SSL or standard. Allows developer to build a Web Server, and develop CGIs and fully functional Web based applications. THEOS implements the server side. For the client all major browsers work nicely.
- SMTP/POP3, full email server with anti-spam and anti-virus capabilities on the server and client side. THEOS O/S implements server and the THEO+Mail client.
- Telnet, allows local or remote network computers using a Telnet client to connect to the THEOS server and login as a user. THEOS implements both client and server.
- LPD/LPR, allows to share Network printers. THEOS Implements both servers and clients.
- TFNS/SMB (NetBios). This server and client allows THEOS to share hard drives, folders or files between THEOS, Linux or Windows or any SMB device.

Proprietary Protocols:

- NetLogin, similar to Telnet, but it is secure and has file transfer embedded within this type of emulation. THEOS implements both client and server.
- Twindows, similar to NetLogin, but for Windows platforms. This server enables the THEOS WorkStation (a terminal emulator package for Windows), to connect to THEOS servers. This allows file transfer, images, distributed user interfaces, etc. This is the preferred THEOS terminal solution.
- TDB, THEOS Data Base Server, a DB engine with SQL conforming ANSI'92 standard and ODBC for Windows.
- TNFS/SMB (NetBios), extensions to the SMB by a THEOS native dialect, allows THEOS Servers to share disk resources and files using its native file system, for example, automatic record locking reading ISAM files.

In a THEOS server, the network can be started and stopped without rebooting the system. It is fast, simple and well integrated into the rest of the operating system. THEOS Network environment includes a set of standard commands that anyone familiar with Windows and Linux would feel comfortable using such as ping, finger, net ipcfg, telnet, sendmail etc.

THEOS SOFTWARE CORPORATION

1801 Oakland Blvd., Suite 315
Walnut Creek, CA 94596
Phone: 925-935-1118
Fax: 925-935-1177
Email: sales@theos.us