Mission Critical DVR

White paper on RTOS and other key components of a reliable DVR

SILENT WITNESS®
Applications Specialists Group
604-574-1526, 604-574-1523
www.silentwitness.com
swsales@silentwitness.com

Preliminary

RTOS Overview:

Most complex devices use a hardware platform, on which an Operating System (OS) is used. The “application” or product software resides on top of all this, and the application dictates what the hardware will do via the OS. The type of operating system used can vary widely. Basing a product on a popular, general purpose OS allows faster development time, with well-supported resources. Using a narrow focus OS can allow other advantages.

To fully appreciate how the OS, Application Software and hardware interact, we must identify the common components. Within the OS itself, there’s common software processes such as: Graphic Subsystems, Device Drivers, I/O Managers, etc.. When the Application runs, it makes requests to the OS. The OS issues tasks via the subsystems, which in turn talks to the hardware. When all these work properly, the task is executed as defined!

Timed Tasks vs. Regular Tasks

A general purpose OS typically doesn’t have strict timed task requirements. If a task is issued, it is assigned a priority in relation to other tasks running. Generally speaking, the task with the highest priority is executed and everything works fine. Typically, a non-RTOS ***cannot*** guarantee that a task will finish at an exact time, because the OS or other components may interfere.

[1]The key characteristic that separates an RTOS from a conventional OS is the predictability of a specific set of tasks. With the RTOS, the device will function and respond exactly the same way in a predictable and repeatable manner. This deterministic behavior minimizes the chances of the system going awry in unforeseen circumstances. Non-RTOS systems may subtly respond differently, at different times with different stimuli.

A conventional OS attempts to use a "fairness" policy in scheduling threads and processes to the CPU. This gives all applications a chance to progress, but doesn't guarantee the priority of threads identified as “real time”, verses those that are not. And although those threads are progressing, their relative priorities may not be kept. This can result in unpredictable delays preventing an activity from completing on time.

Multi-tasking

A general purpose OS can execute tasks (or programs) concurrently. However, it is typical that the OS won’t ***truly*** execute tasks simultaneously; it merely time-shares CPU cycles to accomplish a similar thing. The OS allows one program to operate for a few milliseconds, then the next program a few more, etc. For general purpose use, this works fine. In some cases, the non-RTOS will arbitrarily allow some tasks more bandwidth then others. However, it may be that one of the tasks has to monitor something else in real time. So, the monitoring is intermittent and it forces the developer to work around this situation.

[1] http://www.qnx.com/developer/articles/dec1200b/index.html

A RTOS truly allows multiple programs and tasks to execute simultaneously. The main benefit to this is that critical tasks can execute continuously, minimizing unforeseen responses.

Pre-emptive Multi-tasking

As mentioned, any OS will support concurrent task execution. Throughout the operation, the priority of those processes or threads will change. [1]With the RTOS, scheduling is always performed at the thread level, and threads are always scheduled according to their fixed priority. A high-priority thread that becomes ready to run can preempt a lower-priority thread. **Pre-emptive Multitasking guarantees the correct priority when the time comes, and the OS will know if any tasks are not completed in an absolute time. In doing so, this give greater assurance that the system will perform as intended, in a dynamic environment.**

RTOS Summary

Most of the details described earlier are more important to the design engineers. The main idea here is to illustrate that the RTOS allows more precise control, and repeatable response than a non-RTOS system. It is because of this precision and response that the RTOS is the preferred OS solution when reliability is of top concern.

Realtime operating systems such as QNX are typically used in "mission-critical" environments requiring hard realtime capability, where failure to perform activities in a timely manner put persons or property at greater risk. Life monitoring equipment, driver information systems, [2]nuclear reactors and medical equipment, large telecom networks, are typical “mission critical” applications.

RTOS vs. General Purpose OS Security

A general purpose OS (GPOS) common to most PC’s have many wonderful, well supported features. It can easily support a diverse range of software from Office applications, image capturing and more. The integrated networking capabilities allow a broad range of connectivity option. However, the nature of most GPOS’s is to give priority to innovation over security and reliability.

A popular GPOS offers many ways for applications to automatically execute received files. VBS, Java, ActiveX are just of few examples. Even the integrated networking modules support a wide range communications protocols. FTP, HTTP, FTPS, SMTP and NNTP are some examples. [3]Some of these are enabled by default, and have to be manually disabled if unused.

A GPOS may be prey to an unseemingly endless array of Viruses, Trojan Horse and other malicious executables. If the GPOS based DVR is operated as a closed system (No remote communications allowed, no connection to the Internet, no ability to load a floppy or optical disk), then it can be as secure as any other platform. However, many Security applications require remote access and administration. In order to minimize vulnerability, frequent updating and patches may be required of the GPOS. Additionally, you should run Virus Protection software concurrently, which increases system overhead and robs system resources. These could introduce unforeseen instabilities due to frequent changing of the core OS or Virus Protection program.

[1] http://www.qnx.com/developer/articles/dec1200b/index.html

[2]

http://www.qnx.com/company/index.html

[3] http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/tools/locktool.asp

So the bottom line is: **Do you want your DVR based on an OS that may require frequent updating and patching? Do you or the operator have the time, patience or expertise to install these patches, and verify the system works properly after the changes? Can you risk running a DVR that's not optimized for reliability?**

Purpose Built Platform

A DVR is most easily engineered on a PC platform. In fact, many if not most of the Security DVRs on the market today are just that. The main reasons for this is that the time-to-market is faster, as many components do not have to be custom designed. In fact, there are PCI based video capture cards specially designed so you can develop a DVR yourself. Therefore, it can take very little expertise to cobble together a DVR from off the shelf parts.

Although an RTOS such as QNX can be used on a PC platform, it is well suited for embedded systems. A purpose built, or embedded[4] platform is designed and optimized for a specific application. Properly executed, it can have higher performance and greater reliability than it's PC based counterpart. Another significant advantage is some RTOS's and attendant applications are small enough to be stored away from the primary hard drives. In some cases it can be stored in Solid state Flash memory, which is far more reliable than any hard drive. This can allow for additional fault tolerance, as the DVR can still provide limited functionality even if the hard drive(s) fail. This architecture can allow the "brains" to still work, and possible alert a remote site that the drive(s) have failed. And you might still be able to remotely log in and see live video and identify the problem. If the OS is stored on the primary hard drive, failure of that drive is catastrophic. The system would not be available to query, and it would be unusable until repair.

One common caveat frequently regurgitated by the PC Platform camp is that "Embedded DVRs can't be upgraded". This may be true of some, but the better embedded DVRs ***are*** upgradeable. Most use standard PC based IDE or SCSI hard drives, and frequently can be upgraded. And the best embedded systems allow Flash upgrading of the internal firmware, to allow performance enhancements and add new features.

File System

Most DVR's use a conventional file system like Fat 16, Fat 32 or NTFS. These have some significant advantages in a DVR, in that you can dynamically manage the files. This makes it easy to move files from one place to another, or erase specific, non-contiguous files.

The downside to this technique is that these file systems naturally fragment. If fragmentation becomes excessive, system performance decreases. Because of this, disk utilities may be required to "defrag" the drive, and possibly fix disk errors. If defragmentation is not done periodically, then system performance can be severely degraded, and the DVR may grind to a halt.

Some embedded DVR's use a contiguous file system that does not naturally fragment. That means that you don't have to perform periodic disk maintenance, assuring top performance with less downtime. Another benefit is that a contiguous file system can be formatted in a matter of seconds; FAT systems can take hours to format a large capacity drive. One limitation to the contiguous format is that you cannot selectively "erase" files. However this is generally unnecessary, as the DVR is used to record continually for extended periods, automatically erasing the oldest files first. And, it is undesirable to give anyone the capability to selectively erase a video clip, as it would compromise the concept of a "secure, unalterable" digital record of events.

[4] http://www.microsoft.com/windows/embedded/ce.net/howtobuy/confirm/default.asp

System Recovery after Power Loss

How gracefully the DVR recovers from a Power loss is dependant on many factors. A GPOS based PC is arguably most vulnerable if a power outage takes place. The main reason is that most GPOS's expect to have an orderly shutdown, initiated through the OS GUI. This gives the OS times to stop key processes in a specific order such that data remains intact and the hardware can be shut down safely. If power is interrupted during a hard disk write, the active file will surely be corrupt. If the [5]File Allocation Table (FAT) or the System Registry is being written to during power interruption, the system may not recover at all. If you're lucky, a Scandisk type utility will run during reboot, and offer a chance for the system to eventually recover. But, this could take a while to happen, and is certainly not guaranteed.

Because this is ***not*** an unlikely scenario over the course of operation, many manufacturers recommend using a Uninterruptible Power Supply (UPS) with the GPOS PC based DVR. However, in order to have the best protection, you need a UPS that uses a serial output back to the PC to signal an orderly shutdown. And the DVR must support this for full protection. If the system doesn't have this capability, the UPS will be useless if the power outage lasts longer than the Battery reserves. Not only can this type of subsystem be costly, but it adds another point of failure to the recording system.

A well designed embedded RTOS DVR can avoid some of these problems. And can be more easily designed to gracefully recover after power loss. In fact, it is not uncommon that a UPS is unnecessary with the embedded DVR. Many can recover faster, and more reliably.

Summary

A Video Recording system used in Security applications is expected to operate uninterrupted for long periods of time. Most applications are greatly dependant on the archived video recordings for many reasons. Shortages, shoplifting, false "Slip and Fall" claims can be identified and minimized. Theft, robberies, assaults and other violent crimes can be prosecuted, if the video can be retrieved.

The importance and the value of the recording solution is not apparent to the user until a major event occurs. It is then that the retrieval of video becomes of utmost importance. Failure of the recording system could mean the loss of tens of thousands of dollars, perhaps the loss of the entire business. Worse yet, violence against employees or customers could go unprosecuted. A properly designed, reliable DVR will become invaluable when disaster strikes. And can prove an effective deterrent to crime in general.

The nature of these concerns should make the reliability of the DVR of utmost priority. Too frequently, cost and superfluous features take precedence over core functionality.

Disclaimer: This document is accurate to the best of the author's knowledge, as of June 21, 2002. However, as a Preliminary release, it has not undergone our complete process of qualification to assure accuracy. As such, Silent Witness Enterprises (herein referred as SWE) assumes no responsibility for the consequences of any errors or omissions. In no event shall SWE be liable to you or any third party for any damages or losses of any kind arising out of or relating to the use of this document or its contents, including but not limited to any lost profits, lost opportunity, special, incidental, indirect, consequential or punitive damages, even if SWE has been advised of such damages or losses. No statement within is to be construed as a recommendation to purchase or not purchase any technology discussed. If you find any information to be incorrect, please notify Silent Witness Enterprises. We will review your claim(s) as soon as possible, and if necessary amend this document.

[5] The File Allocation Table is an internal log of all files on disk.

Gyyr DVMS by Silent Witness

After careful consideration, a Real Time Operating System was chosen for the DVMS Digital Video Recorder. Coupled with this are other significant design details:

- Purpose built, non-PC hardware platform
- Real Time OS by QNX, with deterministic behavior and true pre-emptive multitasking
- Ultra reliable Solid state FLASH Memory storage for the OS and application
- Upgradeable Storage both internal and external (SCSI RAID)
- Upgradeable Firmware for future enhancements and new features
- Non-fragmenting file system for video storage
- Power Fault tolerant architecture; Redundant FAT storage and automatic recovery. System operation resumes within 60 seconds of power resumption.

Truly, this is an instance where the whole is greater than the sum of the parts. No other DVR in this market has this synergistic combination of fault tolerant design parameters and future capabilities.