REAL-TIME OPERATING SYSTEMS
It's a Dynamic World in There

The world of embedded systems is growing fast albeit without much ado. Almost every well-advertised development in the computing space, ranging from 64-bit computing to multi-core processors, has been reflected in the embedded systems space as well. As a result, embedded real-time operating systems have also grown correspondingly to enable developers and manufacturers to make the most of key developments in the hardware space

Robots assembling vehicle floors—a fine example of embedded real-time operating systems

■ **JANANI GOPALAKRISHNAN VIKRAM WITH INPUTS FROM SHWETA DHADIWAL**

Gadgets and machines are becoming increasingly powerful. This means that the computing systems embedded in them are also becoming more potent and smarter. Earlier, we used to associate multi-core and such concepts with supercomputers, but then they invaded the personal computing space, and now they are common in embedded systems as well. These changes show that how embedded systems are getting stronger today. Thus, as a complementary product, real-time operating systems (RTOS) are also becoming more powerful.

Let us take a quick look at the recent trends in the RTOS space—in terms of technology, what the industry majors are offering, and the changing requirements of the market.

Faster time-to-market

"New market scenarios indicate that the time-to-market should be the deciding factor in picking up a real-time operating system (RTOS). Although in the past, RTOS' selection was determined by an assessment of the functional characteristics of the candidate system, the increased complexity and the changes in the embedded market means that these factors are no longer applicable. With the emphasis on seizing windows of market opportunity, factors which affect time-to-market have become much more important," says Sunil Kumar Ross, country manager, Ammos Software Technologies Pvt Ltd, speaking for Express Logic, a leading player in the RTOS space.

RTOS must also be capable of meeting the functional requirements of the application. But, Ross notes that using an RTOS that is "overqualified" for an application may have a negative impact on time-to-market. The reason: the additional capabilities make the RTOS more complicated to use, thus pushing the learning curve for developers higher.

So, the bottom line according to him is to select an RTOS that suits project requirements.

He cites the interrupt response time requirements as an example. The common perception is that the fastest RTOS is the best. But some applications do not require such speed. For example, in a desktop system, the keyboard response of about 10 milliseconds is sufficient to keep up with human typing speed. A faster response provides no additional advantage, as the person doing the typing won't be

able to hit the next character for at least 10 milliseconds.

Modular RTOS

As far as embedded systems go, the requirements vary from application to application. A simple gadget may require a small and basic system, while mission-critical, advanced equipment may require more flexible and faster system. Earlier, developers would just choose between a 'light' RTOS or an 'advanced' one, depending on their requirements. But today, the choice is not that simple.

"While minimal configurations are popular for some applications, requirements of many applications are getting more complex. Developers who write their own RTOS or use a minimal one, end up writing more codes to fill the gap between their application needs and what the RTOS offers. This consumes more time and increases the engineering costs," says Brad Dixon, senior product marketing manager, Mentor Graphics.

Nowadays, companies are beginning to provide modular real-time operating systems, enabling developers to choose the parts that they require. Nucleus RTOS, for example, gives developers the freedom of choice. From small applications using minimal kernel and core services to more advanced applications utilising available middleware, Nucleus RTOS satisfies a wide range of customer requirements. The Nucleus RTOS is hardware independent and has been ported to a number of ARM, Power and MIPS architecture central processing units (CPUs).

"Mentor offers a spectrum of software to address many types of developer needs. In a minimal configuration, the core Nucleus RTOS services are very few. Nucleus middleware components can be included to provide additional services such as USB, networking and graphics. For applications that require more extensive system services, Mentor can bring Linux and Android technology to bear on these projects either independently or in conjunction with the Nucleus RTOS," explains Dixon.

Modular RTOS can serve diverse applications, ranging from general-purpose and household appliances to process control and digital signal processing.

Linux-based RTOS

When speaking about modularity, customisability, 'light' versions, choosing the right RTOS, etc, Linux-based RTOS definitely deserves a mention. Of late, quite a few Linux-based real-time operating systems have come up. Some of them are: Stable Hybrid Release (SHR, for the OpenMoko-based mobile phones), Red Hat Embedded Linux and eCos.

The SHR is currently undergoing an extensive development, and at the moment, there are two versions—SHR-Lite and SHR-Full—that only differ in the number of default applications installed. Soon, an SHR-terminal will be developed that ships only with the core basics, not even X11 or a Graphical Interface. It will be used by people who want to create their own RTOS distribution on top of SHR, or use it for other purposes that do not need an X server—from using it for fun applications like connecting the device to a robot controlled by USB and controlling the robot using SMS/phone calls, to using it as a core location-aware (GPS) and always-on (GSM) computer for various purposes.

"In the open source world, many times, less is more, as users usually like to pick their own software and customise the product as they like, and not as the developers like," say Julien Cassignol, Tom Hacohen, Sebastian Krzyszkowiak and Klaus Kurzmann, the core team of SHR.

"We prefer using simpler versions of applications for better performance (speed) and more battery time, as in cellphones, battery time is the most important feature. And, all our source code is available for download and customisation. Users also play an important role in designing our core user interface and applications through their feedback (wikis, mailing lists, Internet relay chat, etc)," they add. User contribution is one of the main benefits of Linux-based open source RTOS; it means that talented people from across the world are working on the software, and hence improving the quality and adding more features.

Real-time operating systems—A quick intro

If you have five tabs opened on your browser, then it is likely that the system will take longer than expected to open a webpage on a sixth tab. When you have a dozen services running on your computer, it is really difficult to say how long the next program might take to load and operate. That is one of the inherent problems in general-purpose computers and their operating systems—you do not know how long it might take to complete an instruction, since it is a function of several other factors such as the number of other services already running, resource utilisation by the other applications, etc.

Just imagine if the case were the same with the operating system running on a computer embedded in, say, a missile launcher or a robotic arm in an automotive assembly line! There will be numerous other machinery and processes dependent on these time-critical equipment or gadgets, and you really need to have an accurate estimate of how long the instrument might take to do its work. This is where a real-time operating system comes into play—it essentially ensures 'deterministic' timing, that is, the amount of time taken by the operating system to respond to an instruction is constant (and small), irrespective of the load, thereby enabling it to be used in real-time applications.

Real-time operating systems (RTOS) are different from the general-purpose operating systems running on our personal computers. Mainly, they are fast; the response times are highly predictable; they are small and occupy very little memory; they consume fewer resources, ensuring that these resources are available for the other operations of the device; they include significantly fewer functions—that is, only those relevant for the device; and are sometimes architecture or microprocessor-dependent.

Twelve questions to answer before you make your pick

1. Are you going to build your own RTOS, choose an open source option, or a proprietary one?
2. What is the total cost of ownership of the RTOS, including licence fees and royalties, if any, and the service or support charges? Is this affordable?
3. What is the memory footprint of the RTOS? Is it small enough?
4. Does the RTOS have all the features required by your application? On the other hand, is it too feature-rich, large and complicated? Is it possible to pick just what you want, and customise, if required?
5. Does it support the architecture you are working on? Multi-core, 64-bit support, etc, if required?
6. How good is the performance: code size and reliability, interrupt latency, context switch times, service overhead, etc?
7. Hope it does not guzzle too much power, memory or other resources?
8. Are there communications stacks (TCP/IP, USB), middleware, and other software available (flash file systems, graphics, etc)?
9. Is it easy to work on, without too sharp a learning curve?
10. Are there sufficient development tools? Are these easy to learn? Are development boards available?
11. Is it scalable and portable? Will it work well with other processors?
12. Does the vendor provide good support?

Although SHR is being made and tested for the ARM-based OpenMoko phones, the open source nature means that it can also be adapted for other architectures. In fact, the SHR team points out that many have shown interest in coming up with a version for the Palm Pre. It can be adapted for usage with personal digital assistants, navigation devices and Internet tablets.

Red Hat also has an RTOS based on the Linux kernel, which gives the user the ability to configure only what they find is necessary. It means that they may trim down the final binary to something that may fit their needs. Plus, the RTOS also inherits the benefits of the Linux parentage. "Any new feature of the Linux kernel becomes a new feature of our RTOS. Currently, we've included several tracing and performance utilities that will help out our customers in fine-tuning their system. We've also included more robust priority inheritance, mutual exclusion, application user interfaces…," says a Red Hat spokesperson.

Although Red Hat is focusing primarily on x86 architectures, the RTOS is available for ARM, PowerPC, SuperH, MIPS and more. Besides, it has been found suitable for a wide range of projects, from the Garmin navigation system to large-scale database centres and embedded systems.

Multi-core RTOS

"The recent trends that we have observed in the embedded RTOS space are around the multi-core and hypervisor technology. Another positive trend is around 64-bit applications in embedded," says Venkatesh Kumaran, country manager-India, Wind River Systems.

Speaking of multi-core processors, the first question that arises in your mind is—cannot any RTOS be used for embedded systems using multi-core processors? You need an RTOS capable of shared memory communication, integrated on-chip I/O support, intelligent cache utilisation strategies, inter-core interrupts, multithreading and multiprocessing—that is, seamless system calls across all processors, symmetric multiprocessing, optional asymmetric multiprocessing, etc.

Keeping this popularity around multi-core in mind, Wind River has initiated a number of solutions targeting this space. They have on-chip debugging solutions that can debug up to eight cores. And, with their hypervisor, you can seamlessly move legacy applications to newer platforms.

The hypervisor technology, at the basic level, can be seen as another name for virtualisation. It is a 'virtual machine monitor' that can be implemented at either or both the hardware and software levels to enable multiple operating systems to run concurrently on a host computer. Virtualisation is essential to ensure full CPU utilisation in a multi-core system.

The technology has recently been extended to embedded systems as well, to enable a high-level operating system interface for application programming, such as Linux or Microsoft Windows. And, at the same time, maintaining traditional real-time operating system application programming interfaces (APIs).

In accordance with the definition of real-time operating systems, embedded hypervisors should also be capable of real-time operation, be it small in size, or using less resources. Hypervisors can support only embedded architectures that allow memory protection, as it is the key requirement for virtualisation. Wind River, OK-labs, VMware, eSOL, etc, have hypervisor solutions for the embedded space.

64-bit RTOS

Regular appliances do not require high-end systems, and 8-bit and 16-bit frameworks are more than sufficient for them. However, there are more complex embedded systems that require greater degrees of computing and much faster processing. For such requirements, the embedded RTOS space is now opting for 64-bit platforms. 64-bit microcontroller units and 64-bit floating point units attain a level of precision that is unimaginable with 32-bit systems. This makes these high-end processors suitable for critical telecommunication systems, process control, and more.

"Today, there are large amounts of data stored in the memory for higher performance. We usually believe that only a 64-bit operating system can suit

Some popular embedded RTOSes

LynxOS. A Unix-like real-time operating system from LynuxWorks. It features full POSIX conformance and Linux compatibility

QNX Neutrino. Supports ARM, MIPS, PowerPC, SH and X86 processor families

VxWorks. A popular RTOS offered by Wind River, an Intel group company. Supports a range of architectures like ARM, PowerPC, ColdFire and MIPS

INTEGRITY. A hard RTOS from Green Hills Software Inc, that supports ARM, PowerPC, ColdFire, x86, and so on

ThreadX. Express Logic's advanced RTOS designed specifically for deeply embedded applications. It supports ARM, PowerPC, x86, etc

Windows CE. Microsoft's offering for embedded systems and minimalist computers, this is supported on Intel x86 and compatibles, MIPS, ARM, and Hitachi SuperH processors

Nucleus RTOS. A product of Mentor Graphics, Nucleus RTOS is a highly-modular, hardware-independent RTOS that has been ported to a number of ARM, Power, and MIPS architecture central processing units

Red Hat Embedded Linux. An open source offering based on the Linux kernel, this has been used for a wide range of purposes, including navigation systems

eCos. An open source RTOS that supports ARM, Hitachi H8300, Intel x86, MIPS, Matsushita AM3x, Motorola 68k, PowerPC, SuperH, SPARC, NEC V8xx, etc

FreeRTOS. FreeRTOS.org is a portable, open source, mini real-time kernel that supports more than a dozen architectures including ARM, AVR32 and MSP430

Symbian OS. A leading open operating system for advanced data-enabled mobile phones. Suits advanced 3G mobile phones also

Stable Hybrid Release. SHR, available in full and light versions, is a Linux-based RTOS developed and tested for ARM-based OpenMoko phones, although it can be ported to other architectures as well

Android. A mobile operating system based on the Linux kernel. Google changed the way the RTOS game was being played, with this successful release. Many consider it a market disruption. Several brands have launched Android phones now

such high-performance metrics and exploit the functionality of the silicon. But Wind River believes that RTOS requires at least 32-bit microprocessors, and we focus on this segment and higher. The most significant feature of our most recent release, VxWorks 6.7, is 64-bit computing," says Kumaran.

64-bit RTOSes usually support 32/64-bit embedded microcontroller units, such as ARM, MIPS, Power Architecture, 68K/Coldfire, x86, SuperH, etc. The company is also capable of working on such platforms.

Android in the RTOS space

"Any discussion around the biggest market disruptions that we have seen in the recent past would have to begin with the mention of Android. It is a game-changing initiative that also saw Wind River's serious foray into the mobile market," says Kumaran. "We have also seen embedded platforms moving to the enterprise computing segment, and the acceptance of Linux. Wind River is definitely gaining traction in this space as the market moves to advanced telecommunications computing architecture/high availability (ATCA/HA) platforms. From the defence perspective, a monumental change has been observed in certifying platforms on a VxWorks 653 operating environment."

Coming back to Android, it is a Linux-based open source operating system for mobile phones. It was launched by Google and then taken over by the Open Handset Alliance. There are now more than a dozen mobile phone models, including some by Samsung, LG and HTC, that use Android. As a result, it has now proved its worth, and is gaining traction. RTOS majors, including Wind River and Nucleus (Mentor Graphics), are now offering Android-based products including development tools.

With Linux and Android in the picture, open source is becoming more popular in the RTOS space.

IPv6 support and other advancements

Basically, embedded RTOSes enable developers and manufacturers to make the best use of the microprocessor board to make their device perform to the optimum level. Hence, if a new feature is required in a product, not only the microprocessor unit but also the embedded RTOS should be able to support it. In that sense, RTOSes are constantly evolving to support new standards and features, ranging from USB 3.0 to Internet Protocol version 6.0 (IPv6).

The Nucleus RTOS, for example, recently rolled out support for IPv6, including certification under the IPv6 Ready Logo (Phase 2) program. It means that the developers who need to integrate their products on IPv6 networks have a certified protocol stack that can be interoperable with other IPv6 networked devices.

More in store

Although hidden somewhere within modern devices, the world of embedded systems is growing, providing an impetus to the growth of the operating system that powers it.

For an example, we have seen that how 64-bit microprocessors that were once used in supercomputers and entered the desktop computing space around 2003, have already become a part of embedded systems. The RTOSes have also quickly adapted to make the most of these powerful systems. Another example of how fast the RTOS world reacts to market changes is its reaction to the success of Android. Almost every market leader has now accepted Android within its folds and offers products for, or based on it.

Therefore, although silent, the embedded RTOS space has shown and continues to show significant growth and innovation, and is worth a frequent look-up. ●

The author is a freelance writer based in Bengaluru. She writes on a variety of topics, her favourites being technology, cuisine, and life