Controlling Laser Ranging with RTAI-based Real-Time Linux

Evan Hoffman (Honeywell), Randall Ricklefs (University of Texas)

ABSTRACT

Currently, many laser ranging systems, such as NASA's MOBLAS systems and the McDonald Laser Ranging Station (MLRS), use proprietary Unix-like real-time operating systems for time-sensitive ranging control software. Such OS's are expensive to maintain and often carry a risk of vendor lock-in. We outline a method of controlling an SLR system using the Linux operating system with the RealTime Application Interface (RTAI) hard real time extension. Linux provides a wide variety of software packages that have low operating cost, are under active development, and are open source. Two flavors of Linux are discussed: Arch Linux and CentOS. Both of these flavors have strengths and weaknesses when being used in a real time environment. Choosing which real time scheduler to use is important for programming considerations. Our approach uses the LXRT scheduler which allows real time control in user mode. We show effective control of an LR system using modest hardware. The current status of conversions of the Goddard Geophysical and Astronomical Observatory (GGAO) 48” telescope and MLRS is presented.

1 Introduction

Accurate satellite laser ranging requires a reliable and predictable method of controlling hardware. This is usually achieved through the use of software running on real time operating systems (RTOS), that is, operating systems whose response time to user-provided interrupts can be guaranteed to be in a given threshold regardless of other software running on the system. Currently, many SLR stations around the world use proprietary, Unix-like RTOS's for hardware control. At the time of implementation, these were the most efficient choices available. However, they are often times very expensive, and software developed for them is not always portable to other RTOS's, creating a risk of vendor lock-in. NASA's MOBLAS systems, and the McDonald Laser Ranging Station at the University of Texas, are examples of systems using proprietary RTOS's.

We propose replacing these proprietary RTOS's with an open source alternative, specifically, Linux. The 2.6 kernel, released in late 2003, provided a significant reduction of overhead thread processing over its 2.4 predecessor. In addition, it allowed processes running kernel mode to be partially preemptable [1]. An additional patch to the stock kernel, the R ealTime Application Interface (RTAI), allows complete premption of kernel running processes [2].

We intend to use the 48“ telescope facility at the Goddard Geophysical and Astronomical Observatory (GGAO) and the McDonald Laser Ranging Station (MLRS) as test stations using this new software.

2 Available Linux and Real Time Linux options

Linux has a large array of distributions suited for a variety of purposes. When selecting a distribution for our systems, we considered stability, quality of software repositories, and ease of maintainability. Arch Linux[3] was chosen for the 48” facility, while CentOS[4] was chosen for MLRS. Both have feature rich package managers for easy installation of software. A brief discussion of each follows. As development continues, these are subject to change.

2.1 Arch Linux

Arch Linux is a general purpose Linux distribution that focuses on simplicity and minimalism.

2.1.1 Advantages

- Arch is a rolling release distribution, that is, each software package is kept at the latest version available. This provides the system with the latest feature updates and bug fixes.
- Arch has an easy to use software package manager with a large library of free software.
- Arch is minimalist. Its base install has little software and low overhead. This allows the system to be lean and bloat free.
- Arch is designed for simple and centralized configuration (BSD style).

2.1.2 Disadvantages

- Being a rolling release, updates to the system should be done carefully as to not jeopardize the stability of the system. Latest software can introduce new bugs.
- Arch requires some knowledge about Linux/OpenBSD system management. It is not the most user friendly distribution on the market.

2.2 CentOS

CentOS is a distribution based on the popular Red Hat Enterprise Linux.

2.2.1 Advantages

- Similar to Arch Linux, CentOS has an easy to use software package manager with a large library of software.
- CentOS has a long term release based on Red Hat Enterprise Linux. It will be supported with security updates for years.

2.2.2 Disadvantages

- CentOS is not as lean as other distributions.
- The current CentOS kernel is 2.6.22.1, a couple versions behind the latest (as of this writing, 2.6.39.3).
- Latest software is not always available in the software repositories, as there is an emphasis on stability.
- Clean shutdown and reboot times are slower when compared to current proprietary RTOS's in use, such as LynxOS as well as other distributions such as Ubuntu.

2.3 Other Linux Options

A full discussion of available Linux options is beyond the scope of this paper. Several other popular distributions were considered. Ubuntu is very popular and user friendly, but suffers from significant bloat. Debian has a reputation for rock solid stability and a very long release time, though its software is usually held several versions back.

3 RTAI: What is it and how does it work?

There are several different real time Linux options. RTLinuxFree, Xenomai, RTAI are free examples, while commercial options such as RTLinuxPro and BlueCat Linux are also available. A complete discussion of them is beyond the scope of this paper. RTAI was selected as it has a very active user base and is free.

RTAI is an real time extension of the vanilla Linux kernel, allowing hard real time. After the patch is applied, it runs as a second kernel alongside the regular Linux kernel. In addition, a nanokernel abstraction layer called Adeos [5] runs as an interface between the Linux kernel and the RTAI kernel. This allows RTAI to take over real time interrupts, while passing regular ones to the regular Linux kernel. The RTAI API is POSIX compliant, which means that much code that has already been written for the proprietary RTOS's can be reused.

RTAI has two process schedulers available to it. The one we have selected, LXRT, provides hard real time in user space, allowing non-root users the ability to run hard real time dependent programs. The advantage is that it allows programs to run without touching the kernel code, which allows for easier programming. It also more closely models how current proprietary RTOS's are used at the NASA stations.

4 Advantages and Disadvantages of Real Time Linux

Here we briefly discuss predicted advantages and disadvantages to using RTAI based real time Linux, in an SLR context, over current proprietary RTOS's.

4.1 Advantages

- RTAI is completely free and open-source. This will allow SLR stations greater control over software and give them the ability to modify it to suit specific needs that may be unique to SLR work. It also reduces possible licensing costs associated with proprietary software.
- Many distributions of Linux have a proven record of stability and security.
- There is a large (and often free) software library available for Linux. Since RTAI runs alongside the regular Linux kernel, a system using it can run any software that was designed for Linux.
- RTAI has an active community for collaboration and support.

4.2 Disadvantages

- Commercial support is unavailable.
- Up to date documentation can sometimes be difficult to find. Conflicting versions of API documents can cause confusion.

5 At MLRS

The Linux/RTAI upgrade at MLRS steals time when the ranging crews are not at work to slip a disk drive with the new operating system and software into the operational ranging system. Thus far, the RTAI/Linux system properly controls all the existing hardware from meteorological sensors, to timing electronics, to the laser and telescope. Hardware-based simulation of tracking can run for hours without unexpected problems. Tests tracking stars and satellites show the system can properly handle the telescope mount. Real satellite ranging was only attempted once so far, with internal calibrations working properly. No satellite data was obtained, but that was not surprising since the engineering staff and not an observing crew made the attempt.

A couple major issues remain. The software could probably be used in production as is, although that won't be done because the CAMAC device driver is not yet running in hard real time mode. The result is that any major disk accesses, such as starting a web browser or compiling software will temporarily drop real time interrupts and therefore prevent ranging.

The other issue is that shutting down or booting CentOS takes much longer than with the proprietary real time OS or other Linux distributions such as Fedora or Ubuntu. Tests will be made with the latest Ubuntu long term support (LTS) version with the aim at replacing CentOS. Initial results are encouraging.

6 At GGAO

The upgrade to RTAI for the 48” facility is part of an ongoing modernization effort. Currently the facility uses a CAMAC interface with ISA computer boards for its hardware/software interface. Using a modern computer with a PCI and PCI Express bus, it is intended that the entire console and control logic be replaced, as well as the optical encoders and the servo pre-amplifiers. PCI and PCI express I/O cards are planned to replace the entire CAMAC interface.

Much of the software code structure is being maintained for the new system, however, the old Motif GUI is being replaced using the GTK+ toolkit. In addition, as RTAI runs as its own kernel, it must communicate with the regular Linux kernel through shared memory and RTAI FIFOs. A major software re-write is underway.

Currently, much of the GUI software has been written. Timing signals and analog out signals are being controlled and read in real time using RTAI. Encoders have been read in soft real time, and conversion to hard real time is underway. These are significant steps to producing a real time system capable of controlling the telescope; an encoder-in analog-out system provides the basic feedback loop necessary.

Due to the large scale of the upgrading effort, progress with the RTAI portion has been slower than hoped, but steady. New optical encoders have been installed into the system, and a logic interface is being tested to read them reliably with the new hardware.

Conclusion

We believe that, given the results of our experimients at the 48“ facility and MLRS, RTAI based real time Linux is a good candidate for replacement of proprietary RTOS's involving the control of SLR systems. There is still much work to be done to prove the concept. Progress speed is dependent on available funding and hours to devote to the project. The ultimate goal will be to acquire good data using an RTAI based system. We hope to make significant progress towards this goal in the coming months.

Acknowledgments

Ricklefs would like to thank NASA for support through NASA contract NNG06DA07C and others in the laser ranging community for their encouragement.

Hoffman would like to thank Jan McGarry, Tom Zagwodski, Jack Cheek, Howard Donovan, and many others at NASA , HTSI, and the ILRS community for their continuing support.

References

[1] "Real-Time and Performance Improvements in the 2.6 Linux Kernel | Linux Journal." *Linux Journal | The Original Magazine of the Linux Community*. Web. 29 July 2011. <http://www.linuxjournal.com/article/8041>.

[2] *Di Milano, Politecnico.* RTAI - Official Website. Web. 29 July 2011. <http://www.rtai.org>.

[3]*Arch Linux*. Web. 31 July 2011. <http://www.archlinux.org/>.

[4]*The Community ENTerprise Operating System*. Web. 31 July 2011. <http://www.centos.org/>.

[5] "Adeos." Web. 31 July 2011. <http://home.gna.org/adeos/>.

Correspondence

Evan Hoffman
Honeywell Technology Solutions, Inc.
7515 Mission Drive, Lanham, MD 20706
Evan.Hoffman@Honeywell.com

Randall Ricklefs
The University of Texas at Austin, Center for Space Research
3925 W. Braker Lane, Suite 200, Austin, TX 78759-5321
ricklefs@csr.utexas.edu