Abstract

In the early 1990’s, MLRS as well as the rest of the NASA SLR network converted their real-time ranging control computers from a set of disparate minicomputers to commodity PCs using the Unix-like real-time operating system LynxOS. We needed to service interrupts in various subsystems in a timely fashion. After a market survey, there were only a couple of affordable contenders that provided a portable interface using POSIX and X11 libraries to provide a common graphical user interface. [1] Recently, as LynxOS became more expensive for one station (we are no longer able to share a development environment with the rest of the NASA network), we needed to find a viable alternative. We have had a great deal of experience working in a Linux environment, and became acquainted with the various open source real-time options for Linux. Ultimately, RTAI (Real Time Application Interface) was chosen [2] It includes patches to the Linux kernel and supplies a set of real-time libraries and schedulers. Later, when it became clear that the device driver for our primary bus system (CAMAC) needed to have real-time capability, we also adopted RTDM (Real Time Device Driver). This is a library of wrapper routines to make writing a real-time device driver simpler and more transportable between real-time implementations like RTAI and Xenomai.

The chosen Linux distribution, after experimenting with a couple others [2], is Ubuntu 12.04 LTS. The boot and shutdown times are better than for most of the other distributions, and this version will be supported with security and bug patches until 2017. Shorter boot times help if the system must be rebooted during a pass, and the long term stability will ensure that the system will remain safely patched for some years – and time will not need to be spent upgrading to a new release every 12-18 months.

The architecture of the software remains the same for the Linux version as for the LynxOS version. There is a multi-threaded main program, called the Monitor, that handles all interactions with the station hardware – mount, timing, ranging hardware, control of the sensors, xy stage, and so forth. This program forks several other programs, only one of which can be active at a single time, and communicates with them through shared memory. These are the star calibrated antenna program, the satellite tracking program, and programs to calculate the TDI ranging timer and the absolute and incremental encoders. Each of these programs is also multi-threaded. All were converted for real-time use with RTAI. All programs are written in c, and heavy use is made of the X11/Motif graphical user interface libraries.

This conversion was undertaken as a low priority project without interference with the usual ranging activities. There were 3 phases to creating the real-time Linux controller system. Each brought us closer to a real-time response.

Production Experience

The Linux real-time system has been in use at MLRS since late August, 2013. Observer training took little time, since there is virtually no difference in the graphical user interface of the ranging programs. In addition, the Linux desktop environment is familiar to the observers because the day to day computing computer also runs Linux. Instead of launching the Monitor program from the root window menu under LynxOS, the observer double clicks an icon on the desktop.

Weather and some software problems limited the number and quality of passes taken early on. Later, a couple of annoyances (problems that proved to limit data and frustrate the observers were remedied, and tracking is now proceeding normally. These problems involved identifying lost interrupts and limiting their detrimental effects. Although the problem of missed interrupts is not new to this real-time Linux implementation, there may still be some tuning of the RTAI/Linux ranging code that can further reduce this problem.

Although Ubuntu boots fairly quickly compared to other distributions, it is slower than LynxOS. There are software or hardware glitches that require a reboot in the middle of a pass, added time rebooting can be frustrating for the operators and can result in the loss of data. To mitigate this slowdown, the hard disk drive was replaced with a 240GB Solid State Drive (SSD). The boot time was cut almost in half, and is comparable to LynxOS boot times.

The choice of desktop environment is evolving. Only a fairly simple desktop is needed to launch the ranging programs and carry out system administration. Production use started with Gnome Classic. This has a bug which often prevents shutdown and reboot from the desktop menus. The Xfce desktop may be used in place of.

Conclusion

The conversion from LynxOS to RTAI/Linux at MLRS was a low priority/low risk venture that bore fruit after a couple years of part time work. To gain experience and control, the project was broken up into several phases, each getting closer to the goal of real-time response. Although the learning curve for RTAI and RTDM was challenging, it was comparable to other unfamiliar environments. Now that the real-time Linux software is in place, it provides an effective, low cost, stable, and tool-rich environment for further development and maintenance.

References