SPACE SYSTEMS SYMPOSIUM (D1)
Space Systems Architectures (4)

Author: Mr. Edward Birrane
The John Hopkins University Applied Physics Laboratory, United States, Edward.Birrane@jhuapl.edu

Mr. Christopher Krupiarz
The John Hopkins University Applied Physics Laboratory, United States, Christopher.Krupiarz@jhuapl.edu
Mr. Alan Mick
The John Hopkins University Applied Physics Laboratory, United States, Alan.Mick@jhuapl.edu

HOMOGENIZING THE DEVELOPMENT AND RUNTIME ENVIRONMENTS OF FLIGHT SOFTWARE SYSTEMS

Abstract

The flight software environment is the combination of hardware, board support package, and operating system used to support application execution. Runtime environments are comprised of real-time operating systems and flight-ready hardware. Development environments are often a heterogeneous collection of non-real-time operating systems, brass-boards, and desktop computers purposed to approximate portions of the runtime system. This multiple-development-avenue approach to optimizing software for a particular mission's runtime system does not, by its nature, prioritize re-usable software. A concentrated effort to standardize environments, especially around open systems, enables multi-vendor and COTS integration beyond those specializing in niche real-time systems.

Development environments indistinguishable from the runtime environment allow software to be coded, tested, and profiled on powerful desktop computers and then cross-compiled directly to the run-time system as flight hardware becomes available. The operating system, as customized by device drivers and BSP, provides an insulating context for the execution of flight software. As such, flight software may now be run anywhere the flight operating system will run.

The Johns Hopkins University Applied Physics Laboratory, with funding provided through the Naval Research Laboratory, conducted a study into the technical feasibility of using real-time Linux as the operating system for a flight command and data handling (CDH) processor. The portions of this study relating to development environments, and our subsequent reference implementation, provides the content for this paper.

This paper describes the construction and technical evaluation of a Linux-based development and test laboratory. We discuss the efficiencies discovered through the process of homogenizing software environments, including reduced debugging and profiling times, the use of laptops to simulate flight hardware, the ability to distribute concurrent software development, and enhanced opportunities for technology transfer.

We present a compelling case for considering the software environment as a vehicle to reduce software development cost and schedule, improve software quality, and increase opportunities for software re-use. Additionally, we posit that the choice to standardize on a widely available operating system provides significant opportunity for collaboration among universities, enthusiasts, space-industry vendors, and non-space application domains such as the telecommunications industry.