Electrical Engineering Department University of Delaware
Technical Report 94-10-1 October 1994

Unix Kernel Modifications for Precision Time Synchronization

David L. Mills

Abstract

This report is a substantial revision of RFC-1589, "A Kernel Model for Precision Timekeeping," March, 1994. It includes several changes to the daemon and user interfaces, as well as a new feature which disciplines the CPU clock oscillator in both time and frequency to a source of precision time signals, as well as provisions to operate with good accuracy at much higher poll intervals in the order of several hours.

This report is included (in ASCII format) as the README.kern file in the NTP Version 3 distribution for Unix, as well as distributions for SunOS, Ultrix and OSF/1 kernel modifications which support precision time functions. Detailed technical information, including source code segments implementing these functions, is also available. Availability of the kernel distributions, which involve licensed code, will be announced separately.

Keywords: operating systems, computer clock, time synchronization, disciplined oscillator.

Sponsored by: Advanced Research Projects Agency under NASA Ames Research Center contract NAG 2-638, National Science Foundation grant NCR-93-01002 and Northeastern Center for Engineering Education contract A30327-93.

Overview

This report describes an engineering model which implements a precision time-of-day function for a generic operating system. The model is based on the principles of disciplined oscillators using phase-lock loops (PLL) and frequency-lock loops (FLL) often found in the engineering literature. It has been implemented in the Unix kernels for several workstations, including those made by Sun Microsystems and Digital Equipment. The model changes the way the system clock is adjusted in time and frequency, as well as provides mechanisms to discipline its frequency to an external precision timing source. The model incorporates a generic system-call interface for use with the Network Time Protocol (NTP) or similar time synchronization protocol. The NTP Version 3 daemon xntpd operates with this model to provide synchronization limited in principle only by the accuracy and stability of the external timing source.

This report does not propose a standard protocol, specification or algorithm. It is intended to provoke comment, refinement and implementations for kernels not considered herein. While a working knowledge of NTP is not required for an understanding of the design principles or implementation of the model, it may be helpful in understanding how the model behaves in a fully functional timekeeping system. The architecture and design of NTP is described in [MIL91], while the current NTP Version 3 protocol specification is given in RFC-1305 [MIL92a] and a subset of the protocol, the Simple Network Time Protocol (SNTP), is given in RFC-1361 [MIL92c].

The model has been implemented in the Unix kernels for three Sun Microsystems and Digital Equipment workstations. In addition, for the Digital machines the model provides improved precision to one microsecond (us). Since these specific implementations involve modifications to licensed code, they cannot be provided directly. Inquiries should be directed to the manufacturer's representatives. However, the engineering model for these implementations, including a simulator with code segments almost identical to the implementations, but not involving licensed code, is available via anonymous FTP.

The NTP Version 3 distribution and technical information distributions can be obtained via anonymous ftp from louie.udel.edu in the directory pub/ntp. The compressed tar archive xntp3.v.tar.Z contains the NTP Version 3 distribution, where v is the version identifier and may be incremented in future versions. In order to utilize all features described in this report, the NTP version identifier should be 4f or later. The compressed tar archive kernel.tar.Z contains additional technical information, as well as this file.

Table of Contents

1. Introduction 1
2. Design Approach 1
2.1. Mechanisms to Adjust Time and Frequency 2
2.2. Daemon and Application Interface 3
2.3. Precision Clocks for DECstation 5000/240 and 3000 AXP Alpha 4
2.4. External Time and Frequency Discipline 5
2.4.1. PPS Signal 5
2.4.2. External Clocks 6
2.4.3. External Oscillators 7
2.5. Other Features 7
3. Technical Summary 7
3.1. PLL/FLL Simulation 8
3.2. The hardupdate() Routine 8
3.2.1. The hardclock Fragment 9
3.2.2. The second_overflow Fragment 10
3.2.3. The hardpps() Fragment 11
3.2.4. External Clock Interface 13
3.3. Leap Seconds 13
3.3.1. Clock Status State Machine 14
4. Programming Model and Interfaces 16
Table 1. Comparison of UTC and NTP Timescales at Leap16
4.1. The ntp_gettime() System Call 17
4.2. The ntp_adjtime() System Call 18
5. References 24

List of Tables

Table 1. Comparison of UTC and NTP Timescales at Leap16