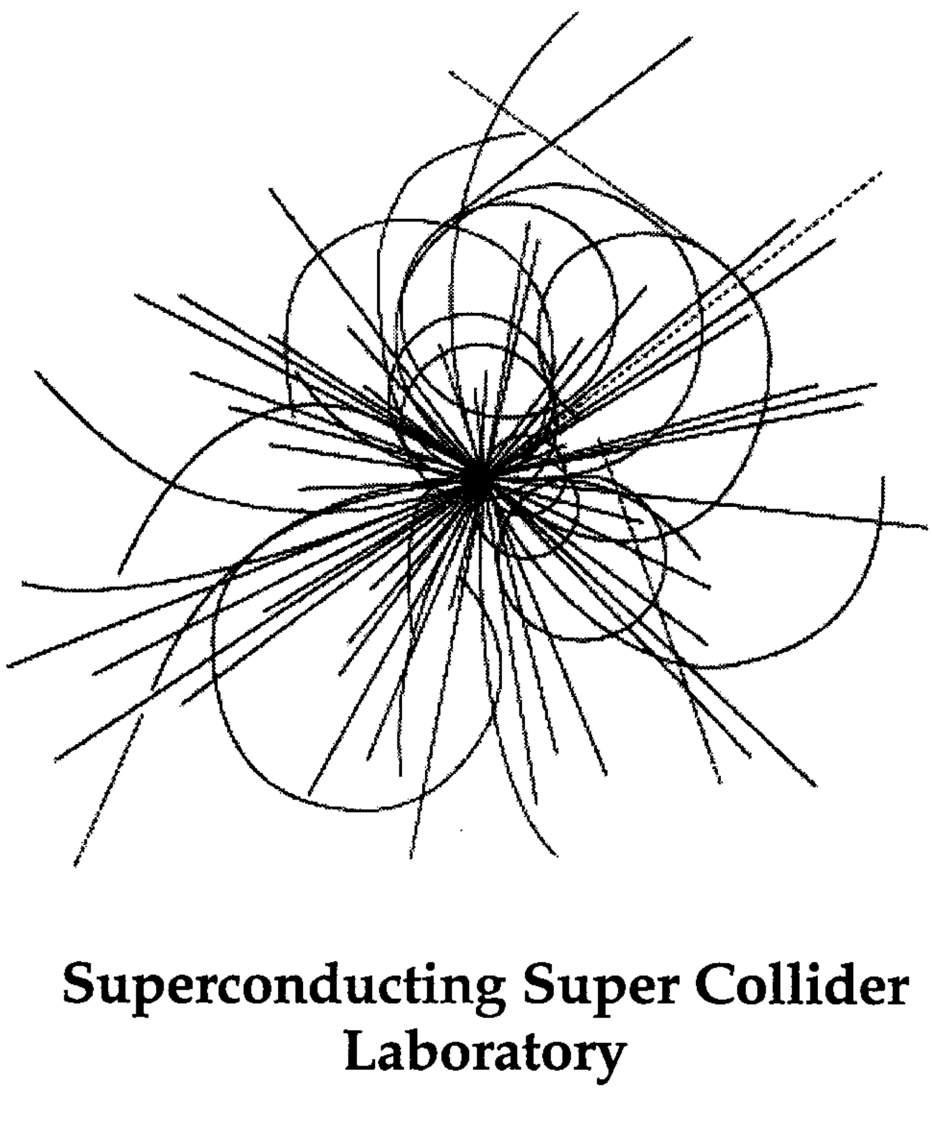
SSCL-N–859

SSCL-N-859
April 1994
Distribution Category: 400

H. Nguyen

Simple Network Management Protocol



Superconducting Super Collider Laboratory

SSCL-N-859

Simple Network Management Protocol*

H. Nguyen

Superconducting Super Collider Laboratory
2550 Beckleymeade Ave.
Dallas, TX 75237 USA

April 1994

*Operated by the Universities Research Association, Inc., for the U.S. Department of Energy under Contract No. DE-AC35-89ER40486.

Superconducting Super Collider Laboratory
Accelerator Division
2550 Beckleymeade Avenue, MS-1046
Dallas, TX 75237-3946

Document/Records Submission Form

Title: Simple Network Management Protocol

Originating Organization: ASD/Controls Department

Author(s) Title

Carl Kalbfleisch
Keng Low
D. Mathieson
Steve Hunt

Date: April 26, 1994

Submitted By: Huan Nguyen

Document/Record Status: **Released By:** **Draft:**

Control Status: **Document/Record Type:**

Document/Record Number:

WBS Number

Alternate Document/Record Number(s):

1)

2)

3)

4)

Keywords:

1) Communication Systems

2) Network Management

3) Management Information Base

4)

Content Description

This document will discuss the setup of a Simple Network Management Protocol (SNMP) for network management of the SSCL Controls system real time embedded processors.

Remarks

Included is the SNMP VxWorks Management Information Base source code.

Network Management of Real-Time Embedded Processors

C. Kalbfleisch, S. Hunt, K. Low, D. Mathieson
SSC Laboratory *
2550 Beckleymeade Ave.
Dallas, Texas 75237

Abstract

The Superconducting Super Collider Laboratory is a complex of particle accelerators being built in Ellis County, Texas. It will have a dedicated global communications network that will deliver control messages and provide for general data acquisition. This network will connect thousands of computer nodes over a very large geographic area. In order to meet the demanding availability requirements being levied on the system, it will need comprehensive network management. A large number of the computer nodes are embedded systems that traditionally do not support network management services. This presents unique challenges to standard network management practices. The Simple Network Management Protocol, SNMP, is widely accepted by industry as a tool to manage network devices. In this paper we will examine the performance characteristics and usefulness of an SNMP agent in a real-time environment.

I. Network Management Historical Perspective

The Internet Activities Board (IAB) has spent considerable time focusing on standards for network management. In 1987 a group of engineers implemented the Simple Gateway Management Protocol (SGMP). Around the same time, the OSI network management documents specified CMIP over TCP (CMOT). (CMIP is the OSI Common Management Information Protocol, and TCP is the Transmission Control Protocol). These two groups met to determine if a consensus could be reached on a network management approach. The result of the meeting was that the SGMP protocol would be extended to address the needs of network devices other than gateways. It would be called the Simple Network Management Protocol (SNMP). This was to become the short term solution for network management in the community while a second group worked on the OSI approach as a long term solution. A third group was to design a common framework for network management so as to make the migration from SNMP to CMOT easier. By the fall of 1989 a number of vendors had implementations of SNMP installed, and it became the de facto operational standard for network management of TCP/IP based networks[1].

II. Structure of Management Information

*Operated by the Universities Research Association, Inc., for the U.S. Department of Energy under Contract No. DE-AC02-89ER40486.

The Structure of Management Information (SMI) was initially deployed in order that the SNMP and the CMOT camps would have a common framework for identifiying managed objects[2]. A collection of managed objects is referred to as a Management Information Base (MIB)[3]. Essentially the SMI specifies a syntax for defining MIBs in Abstrax Syntax Notation One (ASN.1) macros and a base group of object types. Complex object types can be created using ASN.1. Additionally, a set of Basic Encoding Rules (BER) are defined to translate the ASN.1 instances into serialized octet strings that can be sent out onto a network.

Objects are defined with an associated Object Identifier (OID) in a tree structure. There are four branches in the tree that are of primary interest; Directory, Management, Experimental and Private. The Directory subtree is reserved for future use with the OSI network management model. The Management subtree is used to define objects in the Internet standard MIB. This consists of objects that are expected to be available on managed nodes running the Internet suite of protocols. The latest version of this MIB is referred to as MIB-II[4]. It contains 171 objects in a number of groups that are identified as System, Interfaces, Address Translation, IP, ICMP, TCP, UDP, EGP, transmission and SNMP. Each group is considered optional, but if any object in a group is implemented, then the whole group must be implemented. The Experimental subtree is used for conducting Internet experiments. The Private subtree allows any enterprise to register with the Internet community and build their own MIBs. MIBs developed by the SSC Lab fall into this category.

III. What is SNMP?

SNMP has become widely implemented in the network community as the accepted de facto standard network management protocol. Agents supporting SNMP are provided by many network device vendors. In addition, many workstation vendors provide an SNMP agent either as part of their standard operating system release or through third party vendors.

SNMP provides four operations: Get, GetNext, Set, and Trap[5]. It requires a connectionless transport service to be provided. TCP/IP implementations of SNMP use the User Datagram Protocol (UDP) as the transport mechanism. Community names provide for minimal authentication access to a managed nodes MIB. Different community names can be used for Get and Set to provide read-only access to some users and read-write access to others. Get

0-7803-1203-1/93$03.00

and Set operate on a specified OID. GetNext returns the OID and value of the next object in the MIB. This allows a Network Management Station (NMS) to "walk" through a managed nodes MIB by repeatedly issuing GetNext calls to that node. Traps are the means that the managed node can report that an event has occured or some threshold has been passed.

Many commercially available NMSs are available that use SNMP. NMS provides a user interface to the SNMP objects and allows for data collection, MIB browsing, trap handling and maintaining graphical maps of the network. The NMS can draw conclusions on the health of the network and its associated devices based on the MIB data it collects. Additionally, SNMP applications can be purchased or written to manipulate the data in a number of ways. e.g. a meter to show the network or CPU utilization of a particular device.

IV. SNMP Agent for Real-Time Systems

In order to provide SNMP services for real-time systems, SNMP agent software was purchased in source form. This software is the SNMP Universal Agent[6]. C source code is provided to implement the SNMP agent, and sample MIB interfaces which simulate MIB-II and the experimental Uninterruptible Power Supply (UPS) MIB. Each MIB adheres to a defined Agent-MIB interface so that new MIB modules can be added. This code was ported to VxWorks[7], a real-time operating system, by SSC Lab engineers.

VxWorks provides many data structures that contain data relevant to the status of the network. A mapping of this data to the MIB-II objects was performed by SSC Laboratory personnel resulting in a near complete implementation of MIB-II under VxWorks.

V. SSCL Real-Time MIB

With the standard MIB-II objects in place, it is apparent that extensions are needed in order to manage real-time systems. Objects that are not part of any existing standard MIB are desired. For instance, from the central management station, it may be desireable to change a nodes configuration, determine the current software verison, modify the state of tasks in the system, determine the CPU load or even reboot the system.

A MIB was designed to address these unique requirements for management of real-time systems, referred to as the SSCL Real-Time MIB. There are four main groups in the MIB: Real-Time Operating System (RTOS), SNMP Daemon (SNMPD), CPU Idle (IDLE) and System.

The RTOS group contains objects relevant to each real-time operating system that the agent may be ported to. For example, a VxWorks sub-group could contain objects representing system memory usage, system tasks and boot parameters. The current implementation allows the VxWorks boot parameters to be interrogated and modified using SNMP. Due to the distribution of real-time systems at the SSC over many miles, this functionality could be invaluable.

The SNMPD group contains information relevant to the tasks used to support SNMP on the target. These objects include the version of the Agent core, version of the Operating System Port, and the task priority of the daemon task. The task priority is settable using SNMP.

The CPU idle group contains information regarding the utilization of the CPU. Objects provide the CPU percent idle at various time intervals as well as a user settable time interval. The values can be queried to monitor the system performance.

The system group is intended to monitor and control the real-time system as a whole. It contains objects that allow the user to start reboot sequences or abort reboot sequences to a target. When these values are set, SNMP Traps can be sent to a NMS to advise that the system is being rebooted. These traps contain the system being rebooted, how long until the reboot will occur and which system caused the reboot to occur.

VI. SSCL T1 MIB

The communications for the SSC controls system will consist largely of point-to-point links to satisfy the throughput and response time requirements[8]. These point-to-point links will be provided directly into the real-time system by means of a fractional T1 interface. This interface is being developed by the SSC to implement standard protocols such as HDLC, PPP, and TCP/IP. Some of these interfaces are already managed by standard MIBs defined by the Internet community. Where applicable, those MIBs will be used to manage the point-to-point links. In addition to that, a SSCL T1 MIB will be designed to implement direct driver level statistics about the T1 interfaces. This MIB will allow network managers to determine the operational status of the interfaces, and verify that the T1 communications are set up properly in accordance with ADM and SONET equipment used to transport the T1. For example, the T1 channels involved in a point-to-point link could be verified.

VII. Real-Time Performance issues

When real-time systems are operational, there are essentially three types of operations: Interrupt Service Routines (ISR), real-time tasks, and non real-time tasks. The goal of the real-time operating system is to schedule the real-time tasks in a deterministic nature. The non real-time tasks are tasks that are generally not mission critical (e.g. a user level shell), they may use the remaining CPU time, or if there is no remaining time, they are postponed until CPU resources are available.

Since the SNMPD task is not mission critical to the embedded system, but rather provides support information to the management station, it is considered part of the non real-time group of tasks. To address this issue, the SNMPD task has been designed to run at any priority level. The system designer assigns a priority when the SNMPD task is initialized, or also while it is running through the use of SNMP.

The local CPU utilization is only one concern for the impact of the SNMP task on the real-time system. The other consideration is the bandwidth requirement added to the network. In the SSC control system, each real-time system will have a fractional T1 interface. This means that a given system will have a dedicated bi-directional network bandwidth between 64 kbps and 1.554 Mbps. (i.e. the total bandwidth is available for transmitting and receiving data simultaneously). Although the exact traffic patterns for the SNMP data are not known at this time, estimates can be made based on experience using an existing NMS to manage ethernet networks. The NMS could query a node at some user configurable rate for the network utilization, CPU utilization and a few other parameters. This total of about 15 objects could be sufficient to determine if the system is functioning.

It is estimated that each object requires a packet of 128 bytes to be transmitted on the network with a response of the same size. Based on this packet size, SNMP would require 1920 bytes for a complete data acquisition cycle. If the polling period is set to one minute, then the theoretical bandwidth of a 64kbps channel is 480KB/minute, and the bandwidth of a T1 is 11.7MB/minute. The SNMP traffic is then 0.39 percent of the 64 kbps channel or only 0.016 percent of the full T1.

VIII. Future directions

The Real-Time MIB will be expanded to provide additional functionality in terms of Operating System configuration, utilization and task maintenance. Some of these areas will be threshold monitoring for various portions of the MIB including sending out SNMP traps when the CPU utilization passes certain thresholds. It is desireable to be able to set task priorities, interrogate memory, and possibly to use SNMP to implement some debug capabilities for the real-time system. Each of these areas are under investigation.

In the areas of network management protocols, the Internet Engineering task force is nearing completion of the next generation of SNMP, SNMP Version Two, commonly referred to as SNMPv2. SNMPv2 addresses manager-to-manager communications, bulk data transfer, and security enhancements. It is widely expected to be the replacement for SNMPv1. Because of this development, plans are already in place to move the SSC Lab real-time SNMP Agent to SNMPv2 compatibility.

It is not clear if or when the SSC control system network will migrate to an OSI-based network, but we believe that CMOT for real-time systems is feasible.

IX. Conclusions

SNMP seems quite well suited for managing real-time systems in the SSC Lab control system. It provides a common management protocol for traditional network devices as well as UNIX systems, real-time systems and possibly other control system components. SNMP's wide vendor acceptance provides a common ground for network management. This allows the SSC to manage its devices while still using commercial network management packages. With a sufficiently advanced NMS, proactive network management should be feasible. The background work put in by the Internet community will allow following the direction of the community as the standards migrate to newer versions of technology. This will certainly provide some interesting challenges in implementing the network management of the SSC controls system network.

References

[1] M. Rose, *The Simple Book*, Prentice-Hall, Inc., 1991, ISBN 0-13-812611-9.

[2] M. Rose and Keith McGloghrie. *Structure and Identification of Management Information for TCP/IP base Internets*, May 1990.

[3] M. Rose and Keith McGloghrie. *Concise MIB Definitions*, Request for Comments 1212, March 1991.

[4] M. Rose and Keith McGloghrie. *Management Informanation Base for Network Management of TCP/IP-based internets: MIB-II*, Request for Comments 1213, March 1991.

[5] J. Case, M. Fedor, M Schoffstall, J. Davin, *A Simple Network Management Protocol (SNMP).* Request for Comments 1157, May, 1990.

[6] Paul Freeman Associates, 14 Pleasant Street, P.O. Box 2067, Westford, MA 01886.

[7] Wind River Systems, Inc., 1010 Atlantic Ave., Alameda, CA 94501.

[8] S. Hunt, C. Kalbfleisch, K. Low, D. Mathieson, *"BUBBANET" - A High Performance Network for the SSC Accelerator Control System*, these proceedings.

```
SSCL-RT-MIB DEFINITIONS ::= BEGIN

   -- Title:    SSCL real-time MIB
   -- Date:     March 15, 1993
   -- By:       Carl W. Kalbfleisch <cwk@irrational.ssc.gov>
   --
   -- This MIB describes OID's that may be supported in SSCL
   -- real-time systems.
   --
   --
   -- Copyright (C) 1993, University Research Association
   -- All Rights Reserved
   --
   --                              DISCLAIMER
   --
   -- The software is licensed on an "as is" basis only.  Universities Research
   -- Association, Inc. (URA) makes no representations or warranties, express
   -- or implied. By way of example but not of limitation, URA makes no
   -- representations or WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY
   -- PARTICULAR PURPOSE, that  the use of the licensed software will not
   -- infringe any patent, copyright, or trademark, or as to the use (or the
   -- results of the use) of the licensed  software or written material in
   -- terms of correctness, accuracy, reliability, currentness or otherwise.
   -- The entire risk as to the results and performance of the  licensed
   -- software is assumed by the user.  Universities Research Association, Inc.
   -- and any of its trustees, overseers, directors, officers, employees,
   -- agents or contractors shall not be liable under any claim, charge, or
   -- demand, whether in contract, tort, criminal law, or otherwise, for any
   -- and all loss, cost, charge, claim, demand, fee, expense, or damage of
   -- every nature and kind arising out of, connected with, resulting from or
   -- sustained as a result of using this software. In no event shall URA be
   -- liable for special, direct, indirect or consequential damages, losses,
   -- costs, charges, claims, demands, fees or expenses of any  nature or kind.
   -- ----------------------------------------------------------------------
   --

IMPORTS
        enterprises, Gauge
            FROM RFC1155-SMI

        OBJECT-TYPE
            FROM RFC-1212;

  sscl          OBJECT IDENTIFIER ::= { enterprises 535 }

  realTime      OBJECT IDENTIFIER ::= { sscl 1 }

--
--  rtos group
--  define OIDs for each Real Time Operating Systems supported
--
  rtos          OBJECT IDENTIFIER ::= { realTime 1 }
  psos          OBJECT IDENTIFIER ::= { rtos 1 }
  vxworks       OBJECT IDENTIFIER ::= { rtos 2 }
  lynxos        OBJECT IDENTIFIER ::= { rtos 3 }

--
--  snmpd group
```

```
--
  snmpd          OBJECT IDENTIFIER ::= { realTime 2 }

  agentVersion OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
            "An ASCII string containing the version number of the SNMP
            Universal Agent core."
        ::= { snmpd 1 }

  portVersion  OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
            "An ASCII string containing the version number of the Port
            of this agent to this operating system."
        ::= { snmpd 2 }

  taskPriority OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
            "The task priority of the SNMPD task in the real time OS.
            Setting this value causes the task to change its priority
            to the value specified."
        ::= { snmpd 3 }

--
--  idle group
--
  idle           OBJECT IDENTIFIER ::= { realTime 3 }

  currentIdle  OBJECT-TYPE
       SYNTAX  Gauge
       ACCESS  read-only
       STATUS  mandatory
       DESCRIPTION
           "An integer value that corresponds to the percentage
           of the CPU time that was idle in the last second"
       ::= { idle 1 }

  tenSecondIdle OBJECT-TYPE
       SYNTAX  Gauge
       ACCESS  read-only
       STATUS  mandatory
       DESCRIPTION
           "An integer value that corresponds to the percentage
           of the CPU time that was idle in the last ten seconds"
       ::= { idle 2 }

  sixtySecondIdle OBJECT-TYPE
       SYNTAX  Gauge
       ACCESS  read-only
       STATUS  mandatory
       DESCRIPTION
```

```
        "An integer value that corresponds to the percentage
        of the CPU time that was idle in the last minute"
    ::= { idle 3 }

userIdle OBJECT-TYPE
    SYNTAX  Gauge
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "An integer value that corresponds to the percentage
        of the CPU time that was idle in the last user
        defined interval.  If the user-interval value is zero,
        then this value is zero."
    ::= { idle 4 }

userInterval OBJECT-TYPE
    SYNTAX  INTEGER (0..1024)
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION
        "Defines a number of seconds for the user-idle calculation.
        The value must be in the range 0 thru idle.historySize seconds."
    ::= { idle 5 }

calibration OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "This is the count as calculated during the calibration of
        the cpu idle time.  This value can be used to compare the
        calibration between different CPUs."
    ::= { idle 6 }

historySize OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "This is the size of the idle history buffer."
    ::= { idle 7 }

historyValid OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
        "This is the number of valid entries in the history buffer."
    ::= { idle 8 }



--
-- system group
--
  system        OBJECT IDENTIFIER ::= { realTime 4 }

  reboot        OBJECT-TYPE
```

```
     SYNTAX  Gauge
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
         "Setting this value to be non-zero cause the local CPU to be
         reset after the number of seconds specified.  When reading this
         this value, it specifies the number of seconds until the reboot
         will occur, or zero (0) if the reboot sequence has not been
         activated.  The reboot sequence can be aborted by setting the
         abortReboot value to non-zero."
     ::= { system 1 }

rebootVME   OBJECT-TYPE
     SYNTAX  Gauge
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
         "Setting this value to be non-zero cause the VME sysReset line
         to be asserted after the number of seconds specified.  When reading
         this this value, it specifies the number of seconds until the reboot
         will occur, or zero (0) if the reboot sequence has not been
         activated.  The reboot sequence can be aborted by setting the
         abortReboot value to non-zero."
     ::= { system 2 }

abortReboot OBJECT-TYPE
     SYNTAX  Gauge
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
         "Setting this value to be non-zero cause the reboot or rebootVME
         sequence to be aborted."
     ::= { system 3 }

silent OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
         "When set to non-zero, no messages pertaining to the reboot
         are displayed on the console port.  Otherwise warning messages
         are displayed when the reboot sequence is initiated, when it
         is aborted, and once a second for each of the last 5 seconds
         before the reboot."
     ::= { system 4 }

spuriousInts OBJECT-TYPE
      SYNTAX INTEGER
      ACCESS read-write
      STATUS mandatory
      DESCRIPTION
          "Contains a count the number of spurious interrupts that have
          occurred either since the snmpd task was started, or since the
          value was last cleared."
      ::= {system 5}

--
--  define the RTOS groups here
--
```

```
--
--  pSOS+
--
--  This section defines OIDs for the pSOS+ real time executive, and its
--  components
--
--
   nodeAnchor OBJECT-TYPE
       SYNTAX  INTEGER
       ACCESS  read-only
       STATUS  mandatory
       DESCRIPTION
           "Contains the value of the system NODE ANCHOR."
       ::= { psos 1 }

--
--  Node Configuration Table
--
   nodeConfig OBJECT IDENTIFIER ::= { psos 2 }

   cpuType OBJECT-TYPE
       SYNTAX  INTEGER
       ACCESS  read-only
       STATUS  mandatory
       DESCRIPTION
           "A bit mask which defines the CPU type."
       ::= { nodeConfig 1 }

--
--  Multiprocessor Config Table
--
   mpCT    OBJECT IDENTIFIER ::= { nodeConfig 2 }

--
--  pSOS Config Table
--
   psosCT  OBJECT IDENTIFIER ::= { nodeConfig 3 }

   psosCode OBJECT-TYPE
       SYNTAX  INTEGER
       ACCESS  read-only
       STATUS  mandatory
       DESCRIPTION
           "start address of pSOS+."
       ::= { psosCT 1 }

   rgnZeroAddr OBJECT-TYPE
       SYNTAX  INTEGER
       ACCESS  read-only
       STATUS  mandatory
       DESCRIPTION
           "region 0 start address."
       ::= { psosCT 2 }

   rgnZeroLen OBJECT-TYPE
       SYNTAX  INTEGER
       ACCESS  read-only
       STATUS  mandatory
```

```
        DESCRIPTION
            "region 0 length."
        ::= { psosCT 3 }

    rgnZeroUnit OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
            "region 0 unit size."
        ::= { psosCT 4 }

--
--  pROBE Config Table
--
    probeCT OBJECT IDENTIFIER ::= { nodeConfig 4 }

--
--  pHILE Config Table
--
    phileCT OBJECT IDENTIFIER ::= { nodeConfig 6 }

--
--  pREPC Config Table
--
    prepcCT OBJECT IDENTIFIER ::= { nodeConfig 7 }

--
--  pNA Config Table
--
    pnaCT   OBJECT IDENTIFIER ::= { nodeConfig 5 }


--
--  vxWorks
--
--  This section defines OIDs specific to the vxWorks real time
--  operating system and its environment
--

--
--  Memory Stuff
--
vxMemory  OBJECT IDENTIFIER ::= { vxworks 1 }

--
--  Current Memory
--
currentMemory  OBJECT IDENTIFIER ::= { vxMemory 2 }

  bytes OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
          "The current number of bytes allocated"
        ::= { currentMemory 1 }

  blocks OBJECT-TYPE
```

```
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
          "The current number of blocks allocated"
        ::= { currentMemory 2 }

  avgerage OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
          "The average size of the blocks allocated"
        ::= { currentMemory 3 }

  maximum OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
          "The maximum size block allocated."
        ::= { currentMemory 4 }

--
--  Free Memory Stats
--
freeMemory  OBJECT IDENTIFIER ::= { vxMemory 3 }
  bytes OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
          "The current number of bytes free"
        ::= { freeMemory 1 }

  blocks OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
          "The current number of blocks free"
        ::= { freeMemory 2 }

  average OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
          "The average size of the blocks free."
        ::= { freeMemory 3 }

--
--  Cumulative Memory Stats
--
cumulativeMemory  OBJECT IDENTIFIER ::= { vxMemory 4 }

  bytes OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
```

```
        STATUS  mandatory
        DESCRIPTION
          "The cumulative number of bytes allocated"
        ::= { cumulativeMemory 1 }

  blocks OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
          "The cumulative number of blocks allocated"
        ::= { cumulativeMemory 2 }

  average OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
          "The cumulative average size of the blocks allocated"
        ::= { cumulativeMemory 3 }

--
--  Network Stuff
--
vxNetwork  OBJECT IDENTIFIER ::= { vxworks 2 }

vxTcp  OBJECT IDENTIFIER ::= { vxNetwork 1 }

--
--  Task Stuff
--
vxTask  OBJECT IDENTIFIER ::= { vxworks 3 }

vxNumTasks OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
          ""
        ::= { vxTask 1 }

--
--  an entry for each task
--
vxTaskTable OBJECT-TYPE
        SYNTAX SEQUENCE OF VXTaskEntry
        ACCESS not-accessible
        STATUS deprecated
        DESCRIPTION
          "The vxTaskTable contains an entry for each task in the
           system."
        ::= { vxTask 2 }

  vxTaskEntry OBJECT-TYPE
        SYNTAX VXTaskEntry
        ACCESS not-accessible
        STATUS deprecated
        DESCRIPTION
          "Each entry contians one task"
```

```
        INDEX { taskID }
        ::= { vxTaskTable 1 }

 VXTaskEntry ::=
        SEQUENCE {
            name
                OCTET STRING,
            entryPoint
                OCTET STRING,
            taskID
                INTEGER,
            priority
                INTEGER,
            status
                INTEGER,
            programCounter
                INTEGER,
            stackPointer
                INTEGER,
            errno
                INTEGER,
            delay
                INTEGER
        }

name OBJECT-TYPE
      SYNTAX  OCTET STRING
      ACCESS  read-only
      STATUS  deprecated
      DESCRIPTION
        ""
      ::= { vxTaskEntry 1 }

entryPoint OBJECT-TYPE
      SYNTAX  OCTET STRING
      ACCESS  read-only
      STATUS  deprecated
      DESCRIPTION
        ""
      ::= { vxTaskEntry 2 }

taskID OBJECT-TYPE
      SYNTAX  INTEGER
      ACCESS  read-only
      STATUS  deprecated
      DESCRIPTION
        ""
      ::= { vxTaskEntry 3 }

priority OBJECT-TYPE
      SYNTAX  INTEGER
      ACCESS  read-write
      STATUS  deprecated
      DESCRIPTION
        ""
      ::= { vxTaskEntry 4 }

status OBJECT-TYPE
      SYNTAX  INTEGER
```

```
        ACCESS  read-write
        STATUS  deprecated
        DESCRIPTION
          ""
        ::= { vxTaskEntry 5 }

  programCounter OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  deprecated
        DESCRIPTION
          ""
        ::= { vxTaskEntry 6 }

  stackPointer OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  deprecated
        DESCRIPTION
          ""
        ::= { vxTaskEntry 7 }

  errno OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  deprecated
        DESCRIPTION
          ""
        ::= { vxTaskEntry 8 }

  delay OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  deprecated
        DESCRIPTION
          ""
        ::= { vxTaskEntry 9 }



--
--  Boot Param Stuff
--
vxBootParams  OBJECT IDENTIFIER ::= { vxworks 4 }

  bootDev OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
          "boot device code"
        ::= { vxBootParams 1 }

  hostName OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
          "name of host"
```

```
        ::= { vxBootParams 2 }


 targetName  OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
          "name of target"
        ::= { vxBootParams 3 }

 ethernetAddr  OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
          "ethernet internet addr"
        ::= { vxBootParams 4 }

 backplaneAddr  OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
          "backplane internet addr"
        ::= { vxBootParams 5 }

 hostAddr  OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
          "host internet addr"
        ::= { vxBootParams 6 }

 gatewayAddr  OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
          "gateway internet addr"
        ::= { vxBootParams 7 }

 bootFile  OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
          "name of boot file"
        ::= { vxBootParams 8 }

 startupScript  OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
          "name of startup script file"
        ::= { vxBootParams 9 }
```

```
   userName OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
          "user name"
        ::= { vxBootParams 10 }

   password OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
          "password"
        ::= { vxBootParams 11 }

   other OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
          "available for applications"
        ::= { vxBootParams 12 }

   processorNumber OBJECT-TYPE
        SYNTAX  Integer
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
          "processor number"
        ::= { vxBootParams 13 }

   flags OBJECT-TYPE
        SYNTAX  Integer
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
          "configuration flags"
        ::= { vxBootParams 14 }

--
--  vxKernel
--
--  Information about the kernel configuration
--
--
vxKernel  OBJECT IDENTIFIER ::= { vxworks 5 }

--
--  vxClock
--
--  Information about the clock configuration
--
vxClock   OBJECT IDENTIFIER ::= { vxKernel 1 }

  sysClkRate OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
```

```
        DESCRIPTION
            "The rate in ticks per second of the vxWorks clock."
        ::= { vxClock 1 }

  ticks OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-write
        STATUS  mandatory
        DESCRIPTION
            "The number of elapsed clock ticks."
        ::= { vxClock 2 }

--
--  lynx OS
--
--  This section defines OIDs specific to the Lynx OS real time
--  operating system, and its environment
--

END
```

```
SSCL-T1-MIB DEFINITIONS ::= BEGIN

   -- Title:    SSCL T1 MIB
   -- Date:     June 2, 1993
   -- By:       Carl W. Kalbfleisch <cwk@irrational.ssc.gov>
   --
   -- This MIB describes OID's that may be supported in SSCL
   -- systems using T1 communications.
   --
   --
   -- Copyright (C) 1993, University Research Association
   -- All Rights Reserved
   --
   --                              DISCLAIMER
   --
   -- The software is licensed on an "as is" basis only.  Universities Research
   -- Association, Inc. (URA) makes no representations or warranties, express
   -- or implied.  By way of example but not of limitation, URA makes no
   -- representations or WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY
   -- PARTICULAR PURPOSE, that  the use of the licensed software will not
   -- infringe any patent, copyright, or trademark, or as to the use (or the
   -- results of the use) of the licensed  software or written material in
   -- terms of correctness, accuracy, reliability, currentness or otherwise.
   -- The entire risk as to the results and performance of the  licensed
   -- software is assumed by the user.  Universities Research Association, Inc.
   -- and any of its trustees, overseers, directors, officers, employees,
   -- agents or contractors shall not be liable under any claim, charge, or
   -- demand, whether in contract, tort, criminal law, or otherwise, for any
   -- and all loss, cost, charge, claim, demand, fee, expense, or damage of
   -- every nature and kind arising out of, connected with, resulting from or
   -- sustained as a result of using this software. In no event shall URA be
   -- liable for special, direct, indirect or consequential damages, losses,
   -- costs, charges, claims, demands, fees or expenses of any  nature or kind.
   -- ---------------------------------------------------------------------------
   --

  sscl          OBJECT IDENTIFIER ::= { enterprises 535 }

  t1            OBJECT IDENTIFIER ::= { sscl 2 }

END
```

SNMP Agent for VxWorks Documentation

Name

snmpd - Simple Network Management Protocol (SNMP) daemon for vxWorks.

Introduction

Simpe Network Management Protocol (SNMP) is becoming the industry standard for network management. As the name implies, it is fairly simple. In addition, it is extensible. SNMP is a client-server network management archetecture. A server requests information from a clients Management Information Base (MIB). A particular MIB defines some number of Object Indentifiers (OIDs). These OIDs are what is used to request information from a client.

Standard MIBs

There are many standard MIBs defined for various network archetectures. MIB-II is defined by RFC1213. It is propably the most widely supported MIB. It defines groups for the system description, and each layer of an IP based network. These network groups define the interface, IP layer, UDP layer, TCP layer and SNMP group in terms of configuration and statistics. Refer to RFC1213 for the fields that are defined.

Proprietaty MIBs

As mentioned previously, SNMP is extensible. In fact, much of its power is gained by these extensions. An SSCL proprietary MIB has been designed to address OIDs pertaining to real time systems. These OIDs range from statistics and configuration of the real time operating system to the system CPU idle time to facilities to reboot the system.

Loading SNMPD for vxWorks

The agent has been prepared for MVME147, MVME167 and in generic form. The generic form does not support all of the functionality of the others, but allows most functionality. To load the agent, add the following lines to your startup.cmd file.

Motorola MVME147

```
cd "directory containing snmpd.conf"
< /usr/local/comms/vxWorks/startup/snmpd_m147
```

Motorola MVME167

```
cd "directory containing snmpd.conf"
< /usr/local/comms/vxWorks/startup/snmpd_m167
```

GENERIC

```
cd "directory containing snmpd.conf"
< /usr/local/comms/vxWorks/startup/snmpd_generic
```

snmpd.conf file

The snmpd.conf file configures two values in the MIB-II system group. The first string found in the file becomes the agent's sysContact value. The second string is the sysLocation. An example file is:

```
####################################
####################################
##  System group
####################################
####################################

"Elvis"                          # sysContact
"The dark side of the moon"      # sysLocation
```

If the values need to be reconfigured, you can cd to a directory containing an snmpd.conf file and execute init_system () from the vxWorks shell.

If the snmpd.conf file does not exist, the the values are set to:

sysContact: my dear contact
sysLocation: my dear system location

VxWorks Task uasge

The SNMP agent and its support code creates 3 tasks. They are called LOAD, LAVG and SNMPD. The LOAD and LAVG tasks are used to calculate the system idle time. LOAD runs at priority 0 once a second. LAVG runs at priority 254 and uses all CPU time not used by other tasks or ISRs. DO NOT start another task at priority 254. The SNMPD task is the agent code that supplies OID values to management consoles. The scripts defined above start this task at priority 20. The priority can be changed while the task is running by using SNMP.

A task REBOOT or REBOOT_VME will be spawned if a reboot sequence is activated. These tasks are created at priority 0.

NOTE: VxWorks task priorities range from 0 (highest) to 254 (lowest).

Support Entry Points

There are a number of support entry points that can be activated from the vxWorks shell. These are as follows:

init_system ()

The init_system () entry point is used to re-read snmpd.conf. This may require to have an NFS mounted file system in order to access the file.

idleShow

The idleShow entry point will display the current, ten second average and 60 second average of the system idle time for the user.

systemShow

The systemShow () entry poiny will display the SNMP MIB-II group OIDs and their values.

Host management tools (on Lepew)

MIB browser

The MIB browser is an X window application that allows a user to interactively view and modify the values of an agents MIB. Invoke the MIB browser by running xnmbrowser on lepew. The path /usr/OV/bin should be added to your path, and your DISPLAY variable set appropriate to your physical location. Once xnmbrowser comes up on you screen, you may specify the target you wish to view in the (Name/IP address field). Move around the MIB with the (Up/Down Tree) buttons. The MIB-II values are in the tree in the .iso.org.dod.internet.mgmt.mib-2 area. The SSCL Real Time MIB is in the tree in the

snmpget, snmpset, snmpwalk

There are man pages for these applications that are available as part of HP OpenView.

Additional Information

Refer to the file /usr/local/comms/vxWorks/obj/README for information pertaining to the functionality of SNMP agent for vxWorks and the related support code.

License

The SSCL is authorized to run unlimited copies of this code. It may not be distributed or run off site of the SSCL.

Glossary Of Terms

SNMP Simple Network Management Protocol

MIB Management Information Base

OID Object Identifier

NMS Netork Management Station

agent

code that responds to SNMP requests

OpenView

A NMS from Hewlett Packard that is currently used by ASD Controls. The software is installed on lepew (an HP 425S).

See Also

snmpget, snmpset, snmpwalk, sscl_rt

Full Internet Standards

The following are full Internet Standards

- rfc1155 – Structure and Identification of Management Information for TCP/IP–based Internets
- rfc1157 – A Simple Network Management Protocol (SNMP)
- rfc1212 – Concise MIB Definitions
- rfc1213 – Management Information Base for Network Management of TCP/IP–based Internets: MIB–II

Draft Standards

The following are Draft or Proposed Internet Standards

- rfc1271 – Remote Network Monitoring (RMON)
- rfc1285 – FDDI Management Information Base
- rfc1406 – Definitions of Managed Objects for the DS1 and Es1 Interface Types
- rfc1441 – Introduction to version 2 of the Internet–standard Network Management Framework
- rfc1442 – Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)

Network Management Glossary of Terms

agent

code that responds to SNMP requests

ASN.1

Abstract Syntax Notation One

BER

Basic Encoding Rules are defined to translate ASN.1 representation of a MIB into a serialized octet string suitable for transmission on the network.

CMIP

Common Management Information Protocol

CMOT

CMIP over TCP/IP

MIB

Management Information Base

NMS

Network Management Station

OID

Object Identifier

OpenView

A NMS from Hewlett Packard that is currently used by ASD Controls. The software is installed on lepew (an HP 425S).

SGMP

The Simple Gateway Management Protocol was the predecessor to SNMP.

SMI

Structure of Management Information. For SNMP, this is defined in rfc1155. For SNMPv2, this is defined in rfc1442.

SNMP

The Simple Network Management Protocol protocol provides four operations: Get, GetNext, Set and Trap. SNMP refers to the first version of SNMP.

SNMPv2

Simple Network Management Protocol version 2 provide security and bulk data transfer enhancements to SNMP.

Spectrum

A NMS from Cabletron Systems.