Document: T10/97-267 revision 0
Date: 97-11-05
To: NCITS T10 SCSI Working Group
From: Keith W. Parker <diogenes@europa.com>
Subject: SCSI Socket Services (SSS) Review 97-11-05

SCSI Socket Services (SSS) Review 97-11-05

Project T10/1246-D

Status

Keeping a low profile until first draft released

Following John Lohmeyer's counsel
Studying to find answers before questions are asked

Finished Linux network socket code diving

SSS Remote Procedure Set requirements determined

Linux selected as development platform

Not part of standard, demonstration development platform
Free operating system, source code, & development tools
Free availability on Internet
Strong academic involvement
Strong international involvement
Free is very good for embedded systems

WinSock2 on top of BSD SSS appears viable

Simplifies minimum functionality requirements
Eliminates Proprietary vs. Standard issues
The WinSock2 wrapper will be a bit thicker than BSD

Schedule

98-01 Jan T10 reflector

First draft release

98-03 Mar T10 mtg

Review & editing

98-05 May T10 mtg

Review & editing
First operational demonstration

98-07 Jul T10 mtg

Review & editing
Earliest possible SSS standard approval

98-09 Sep T10 mtg

Target SSS standard approval
Available for 98-99 academic year

98-11 Nov T10 mtg

SCSI CAM IrDA Project Proposal

Linux Socket Code Diving

Finding what needs to be done and where to do it.

"Linux Kernel Internals" (LKI)

M. Beck, H. Bohme, M. Dziadzka, U. Kuntz, R. Magnus, D. Verworner
Addison-Wesley, ISBN 0-201-87741-4

Redhat 4.2 distribution of Linux

SSS RPS functions required

Socket functions

SEE: LKI A.3 Communication
socketcall() functions

SYS_SOCKET - socket()
SYS_SOCKETPAIR - socketpair()
SYS_GETSOCKOPT - getsockopt()
SYS_SETSOCKOPT - setsockopt()
SYS_BIND - bind()
SYS_CONNECT - connect()
SYS_LISTEN - listen()
SYS_ACCEPT - accept()
SYS_RECV - recv()
SYS_RECVFROM - recvfrom()
SYS_SEND - send()
SYS_SENDTO - sendto()
SYS_SHUTDOWN - shutdown()
SYS_GETPEERNAME - getpeername()
SYS_GETSOCKNAME - getsockname()

File functions

SEE: LKI 6.2.6 File operations
FROM: m:\live\usr\src\linux-2_.30\net\socket.c

```
static struct file_operations socket_file_ops = {
        sock_lseek,
        sock_read,
        sock_write,
        NULL,           /* readdir */
        sock_select,
        sock_ioctl,
        NULL,           /* mmap */
        NULL,           /* no special open code... */
        sock_close,
        NULL,           /* no fsync */
        sock_fasync
};
```

SCSI adapter manufacturers should support Linux SCSI CAM

The Ultimate Application Note

SCSI SSS Remote Procedure Set (RPS)

BSD API to WinSock API to WinSock SPI to SSS function mapping

BSD functions (22)	**WinSock API** functions (35)	**WinSock SPI** functions (32)	**Blk**	**SSS** functions (35/50)
	WSAStartup()	WSPStartup()		SSS_drvr__startup()
	WSACleanup()	WSPCleanup()		SSS_drvr__cleanup()
		WSCInstallProvider() DLL		SSS_drvr_provider_install()
		WSCDeinstallProvider() DLL		SSS_drvr_provider_deinstall()
	WSAEnumProtocols()	WSCEnumProtocols() DLL		SSS_drvr_proto_enum() SSS_drvr_proto_qry()
socket()	WSASocket()	WSPSocket()		SSS_sock_create()
socketpair()	WSADuplicateSocket()	WSPDuplicateSocket()		SSS_sock_create _pair()
ioctlsocket()	WSAIoctl()	WSPIoctl()	y	SSS_sock_vfs_ioctl()
select()		WSPSelect()	y	SSS_sock_vfs_select()
closesocket()		WSPCloseSocket()	y	SSS_sock_vfs_close ()
getsockopt()		WSPGetSockOpt()	*	SSS_sock_opt_get()
setsockopt()		WSPSetSockOpt()	*	SSS_sock_opt_set()
	WSAGetQOSByName()	WSPGetQOSByName()	*	SSS_sock_QOS_get_by_name()
bind()		WSPBind()		SSS_state_bind()
connect()	WSAConnect()	WSPConnect()	y	SSS_state_connect()
	WSAJoinLeaf()	WSPJoinLeaf()	y	SSS_state_connect_leaf()
listen()		WSPListen()		SSS_state_listen()
accept()	WSAAccept()	WSPAccept()	y	SSS_state_accept()
recv()	WSARecv()	WSPRecv()	y	SSS_xfer__recv()
recvfrom()	WSARecvFrom()	WSPRecvFrom()	y	SSS_xfer__recv_from()
send()	WSASend()	WSPSend()	y	SSS_xfer__send()
sendto()	WSASendTo()	WSPSendTo()	y	SSS_xfer__send_to()
			y	SSS_xfer_vfs_fasync()
			y	SSS_xfer_vfs_read()
			y	SSS_xfer_vfs_write()
shutdown()		WSPShutdown()		SSS_conn_shutdown()
	WSARecvDisconnect()	WSPRecvDisconnect()		SSS_conn_recv_disconnect()
	WSASendDisconnect()	WSPSendDisconnect()		SSS_conn_send_disconnect()
getpeername()		WSPGetPeerName()		SSS_name_peer_get()
getsockname()		WSPGetSockName()		SSS_name_sock_get()
htonl()	WSAHtonl()			SSS_conv_hst_to_net_l()
htons()	WSAHtons()			SSS_conv_hst_to_net_s()
ntohl()	WSANtohl()			SSS_conv_net_to_hst_l()
ntohs()	WSANtohs()			SSS_conv_net_to_hst_s()
----------	----------	----------		----------
	WSAGetLastError()			SSS_error_last_get()
	WSASetLastError()			SSS_error_last_set()
	WSAGetOverlappedResult()	WSPGetOverlappedResult()		SSS_event_ovrl_result_get()
	WSAEnumNetworkEvents()	WSPEnumNetworkEvents()		SSS_event_net_enum()
	WSAAsyncSelect()	WSPAsyncSelect()		SSS_event_net_async_select()
	WSAEventSelect()	WSPEventSelect()		SSS_event_net_select()
	WSACreateEvent()			SSS_event_sys_create()

WSACloseEvent()			SSS_event_sys_close()
WSASetEvent()			SSS_event_sys_set()
WSAResetEvent()			SSS_event_sys_reset()
WSAWaitForMultipleEvents()		y	SSS_event_sys_wait_multiple()
WSAIsBlocking()			SSS_block_ing_is()
WSASetBlockingHook()			SSS_block_hook_set()
WSAUnhookBlockingHost()			SSS_block_host_unhook()
WSACancelBlockingCall()	WSPCancelBlockingCall()		SSS_block_call_cancel()

SSS Related Projects

CAM Target Mode Access/Extension
BIOS Embedded SCSI CAM (BESC)
SSS Processor Modules
SSS enhanced SCSI Disk Drives

CAM Target Mode Access/Extension

SCSI CAM as alternate API access to existing drivers
Does not conflict with existing (ASPI, etc.) drivers
SCSI CAM standard defines Target Mode API
Platform / Device Independently (PDI)
SCSI CAM is more than just parallel
SSA, Fibre Channel, IEEE-1394 (FireWire), GPP (IrDA)
Sell millions of chips instead of a few SDK's

BIOS Embedded SCSI CAM (BESC)

SCSI CAM active while BIOS ROM Extensions are called
SSS devices can communicate before OS loaded
First extension device installs CAM-XPT and first CAM-SIM(s)

```
if( ! CAM_XPT_detected()  )
      CAM_XPT_install();
CAM_SIM_register (  CAM_SIM_list );
```

Other extension devices register additional CAM-SIM(s)

```
CAM_SIM_register (  CAM_SIM_list );
```

BESC able to pass control to/from OS Protected Mode drivers
Eventually CAM to be added to Virtual Machine Manager (VMM)
Ideal for embedded systems
Sell millions of chips instead of a few SDK's

SSS Processor Modules

Foundation of many new devices / products / markets
Cost-Effective systems upgrading
Highly scaleable
Symmetric Multi-Processing
Embedded systems applications
Application Modules
Software and hardware applications integrated inside SSS Processor Modules
With any laptop computer design:
Delete Expensive Display
Delete Expensive Keyboard/Pointer
Delete Expensive Laptop Packaging
Add SCSI Chip(s)
Add BIOS Embedded SCSI CAM (BESC) - opt.
Add Low-Cost Stacking Expansion
Add Low-Cost 5.25" SCSI Drive Packaging
Battery backed-up SSS Processor Module very useful
Choice of Laptop or SSS Processor Module packaging

Whole new markets with little additional investment
Physically Secure with Tamper Protection - opt.
SCSI only cabinets, w/o Motherboards, become the norm
Combo SCSI / LPT port
Much more effective use of laptop printer ports as SCSI
Laptop market virtually untapped by SCSI
SSS Processor Modules for any processor type / power
From microcontrollers to super-micros
8051, 65X, 68X, X86, Power-PC, Sparc, Alpha, DSP
User I/O optional
Video, keyboard, pointer(s), audio
Other I/O optional
Via. low-cost stacking expansion ports
PC/104-Plus

SSS enhanced SCSI Disk Drives

New Features
Physical Partitioning into multiple LUN's
Carve large disk drives into manageable pieces
Appear to be separate SCSI drives to OS(s)
Write Protection of LUN partitions
Hiding of LUN partitions
FTP access to Secure File Systems via SSS
Self-contained file system
No external OS support required
Self-protected file system
Will not accept invalid commands
Platform / Device Independent (PDI)
Highly transportable between systems
Physically Secure with Tamper Protection (optional)
All writes to non-volatile storage are encrypted
Tampering removes power to crypto keys RAM
Automatic Secure Archive
Automatic Secure Transaction Logging
Key Switch(es) support
Enhancements accessed via SSS
Web Browser
Configuration
User monitor / control
FTP
Other useful RFC's
Development Cycle
Prototyped on PC as a SCSI Old Computer Trick
Initial Product: a board to mount under 3.5"x1" SCSI disk
Finished Product: totally integrated inside SCSI disk drive

Keith W. Parker <diogenes@europa.com>

Technical Editor, NCITS T10/1246-D
SCSI Socket/SSL Services (SSS) Command Set

Diogenes' Unofficial SCSI Socket Services (SSS)
http://www.europa.com/~diogenes/SSS/

Diogenes' SCSI Magic File Drive Systems Manifesto
http://www.europa.com/~diogenes/DSMFDSM/