

Small Computer Systems Interface (SCSI)

Rômulo Silva de Oliveira

www.das.ufsc.br/~romulo DAS - UFSC

www.hochfeiler.it/alvise/ASPI_1.HTM

www.cdrlabs.com/articles/index.php?articleid=3&page=1
scsifaq.org:9080/scsi_faq/

Rômulo Silva de Oliveira 16/12/02

1

• ASPI

Rômulo Silva de Oliveira 16/12/02

2

ASPI

- Adaptec facilitou a programação de dispositivos SCSI2 através dos "ASPI drivers"
 - ASPI significa "Advanced SCSI Programming Interface"
- ASPI fornece uma maneira fácil de:
 - Gerenciar dispositivos SCSI2
 - Enviar comandos SCSI2 para eles
- ASPI também pode ser usado na prática para gerenciar dispositivos SCSI3 e enviar comandos SCSI3 para eles

Rômulo Silva de Oliveira 16/12/02

3

ASPI no Windows

- Suporte para ASPI era fornecido no MS-DOS por um driver incluído no arquivo CONFIG.SYS
 - ASPI4DOS.SYS
- Em Windows 3.1x suporte era provido por
 - WINASPI.DLL
- Nas plataformas Win32 o suporte para programação ASPI é fornecido por
 - Windows 95, 98 e ME
 - WNASPI32.DLL, APIX.VXD, ASPIENUM.VXD
 - Windows NT, 2000 e XP
 - WNASPI32.DLL, ASPI32.SYS

Rômulo Silva de Oliveira 16/12/02

4

ASPI no Windows

- Adaptec costumava ajudar os programadores:
- ASPI32.EXE
 - Utilitário para atualizar ASPI em Windows 95 e NT
- ASPICHK.EXE
 - Utilitário para verificar a instalação, usado para determinar se suporte ASPI está apropriadamente instalado e também para descobrir qual versão está instalada
- ASPI SDK, "ASPI Software Development Kit"

Rômulo Silva de Oliveira 16/12/02

5

ASPI no Windows

- www.adaptec.com
- Atualmente menos coisas são fornecidas gratuitamente
- Texto com explicações e código:
 - ftp://ftp.adaptec.com/obsolete/adaptec/aspi_w32.txt

Rômulo Silva de Oliveira 16/12/02

6

ASPI no Windows

- O coração da programação ASPI no Windows é a biblioteca WNASPI32.DLL
- Suporte ASPI provido por WNASPI32.DLL consiste de 5 funções e 8 comandos ASPI para o programador usar
- Entre as 5 funções apenas 2 delas são realmente importantes
 - GetASPI32SupportInfo
 - SendASPI32Command

Rômulo Silva de Oliveira 16/12/02

7

ASPI no Windows

- GetASPI32SupportInfo
 - Não tem parâmetros
 - Usada para verificar se o hardware e software estão propriamente instalados no PC
 - Retorna uma doubleword composta por 3 pedaços
 - Bits 31-16 não são usados, zerados
 - Bits 15-8 representam o "status code"
 - Bits 7-0 retornam a contagem de host adapters
 - O número de host adapter instalados
 - Se um card é colocado no barramento do PC a contagem é 2
 - Pois o controlador IDE/EIDE é sempre "visto" pelo suporte ASPI

Rômulo Silva de Oliveira 16/12/02

8

ASPI no Windows

- Se o status code for SS_COMP
 - Suporte ASPI está OK
- Outros valores indicam a causa da ausência do suporte, por exemplo
 - SS_NO_ADAPTERS
 - Indica que está faltando o hardware
 - SS_NO_ASPI
 - Indica que o gerenciador ASPI não foi encontrado

Rômulo Silva de Oliveira 16/12/02

9

ASPI no Windows

- SendASPI32Command
 - É a função ASPI usada para executar os 8 comandos ASPI providos pelo suporte ASPI
- Trata-se da peça central na programação ASPI
- SendASPI32Command
 - é uma função de propósito geral
 - usada para executar os 8 comandos ASPI
- Importante:
- Comandos ASPI são DIFERENTES de comandos SCSI2

Rômulo Silva de Oliveira 16/12/02

10

Comandos ASPI

- Comandos ASPI podem ser síncronos ou assíncronos
- **Comandos ASPI síncronos**
- Inquiry: obtém informação dos host adapters instalados
- Get device type: obtém o tipo de dispositivo em um endereço SCSI específico
- Abort SRB: cancela um comando ASPI prévio
- Get disk info: obtém informação do BIOS sobre discos SCSI2 (Win95)
- Rescan SCSI2 bus: solicita nova varredura do bus
- Get/set timeouts: gets/sets SRB timeouts com relação a dispositivos específicos

Rômulo Silva de Oliveira 16/12/02

11

Comandos ASPI

- Comandos ASPI assíncronos
- Exec SCSI2 command: envia um comando SCSI2 para um dispositivo SCSI2 alvo
- Reset device: envia uma mensagem "Bus Device Reset" para um dispositivo SCSI2 alvo
- O uso apropriado dos comandos ASPI é por conta do programador
- Os comandos ASPI mais utilizados são "Inquiry", "Get device type" e "Exec SCSI2 command"

Rômulo Silva de Oliveira 16/12/02

12

Comandos ASPI

- ASPI command
 - ASPI command code ASPI command type
- Inquiry
 - SC_HA_INQUIRY synchronous
- Get device type
 - SC_GET_DEV_TYPE synchronous
- Abort SRB
 - SC_ABORT_SRB synchronous
- Get disk info
 - SC_GET_DISK_INFO synchronous

Rômulo Silva de Oliveira 16/12/02

13

Comandos ASPI

- ASPI command
 - ASPI command code ASPI command type
- Rescan SCSI2 bus
 - SIZE=2>SC_RESCAN_SCSI_BUS synchronous
- Get/set timeouts
 - SC_GETSET_TIMEOUTS synchronous
- Exec SCSI2 command
 - SC_EXEC_SCSI_CMD asynchronous
- Reset device
 - SC_RESET_DEV asynchronous

Rômulo Silva de Oliveira 16/12/02

14

SendASPI32Command

- SendASPI32Command
 - Recebe um parâmetro
 - Um ponteiro para o SCSI2 Request Block
 - SRB
- O SRB é uma estrutura que deve ser preparada antes de chamar a função
- Existem tantos SRBs quantos forem os comandos ASPI, conforme tabela

Rômulo Silva de Oliveira 16/12/02

15

SendASPI32Command

- Inquiry SC_HA_INQUIRY
 - SRB_HAInquiry
- Get device type SC_GET_DEV_TYPE
 - SRB_GDEVBlock
- Abort SRB SC_ABORT_SRB
 - SRB_Abort
- Get disk info SC_GET_DISK_INFO
 - SRB_GetDiskInfo

Rômulo Silva de Oliveira 16/12/02

16

SendASPI32Command

- Rescan SCSI2 bus SC_RESCAN_SCSI_BUS
 - SRB_RescanPort
- Get/set timeouts SC_GETSET_TIMEOUTS
 - SRB_GetSetTimeouts
- Exec SCSI2 command SC_EXEC_SCSI_CMD
 - SRB_ExecSCSICmd
- Reset device SC_RESET_DEV
 - SRB_BusDeviceReset

Rômulo Silva de Oliveira 16/12/02

17

SendASPI32Command

- Os primeiros 8 bytes de qualquer SRB são sempre os mesmos
- Os bytes restantes são específicos de cada comando
- Códigos de comandos e estruturas SRB são definidas em WNAPI32.H
- Vamos revisar os 4 primeiros bytes

Rômulo Silva de Oliveira 16/12/02

18

SendASPI32Command

- Primeiro byte é o "ASPI command code"
 - Conforme lista mostrada antes
- Segundo byte (SRB_Status) é "ASPI command status"
 - Para ser verificado após qualquer chamada para a função SendASPI32Command
- Terceiro byte é o "ASPI host adapter number"
 - Deve conter o número do host adapter em questão
 - Esta numeração é atribuída pelo gerente ASPI para cada adaptador instalado
- Quarto byte é o "ASPI request flags"
 - Deve ser zero a não ser para comandos SC_EXEC_SCSI_CMD ou SC_GETSET_TIMEOUTS

Rômulo Silva de Oliveira 16/12/02

19

SendASPI32Command

- SendASPI32Command invocada para um comando síncrono somente retorna após o comando ter sido concluído
- O byte "ASPI_Status" deve ser testado para saber o resultado do comando

Rômulo Silva de Oliveira 16/12/02

20

SendASPI32Command

- SendASPI32Command invocado para um comando assíncrono retorna imediatamente o controle para o programa chamador
 - O byte "ASPI_Status" retorna com "SS_PENDING"
- Quando o comando é realmente concluído o byte "ASPI_Status" na estrutura SRB conterá o verdadeiro valor do "ASPI command status"
- Enquanto o comando não é concluído, a aplicação pode invocar outros comandos ASPI

Rômulo Silva de Oliveira 16/12/02

21

SendASPI32Command

- Como fazer para o programa tomar ciência de quando o comando ASPI assíncrono está concluído ?
- Um método trivial seria testar continuamente o byte "ASPI_Status byte"
 - Quando ele mudar de SS_PENDING para qualquer outra coisa significa que o comando ASPI foi concluído
 - E o byte representa o verdadeiro código de retorno da chamada SendASPI32Command
- Outro método seria enviar o comando ASPI e armar uma "função callback"
 - Que será chamada pelo gerenciador ASPI manager
 - Quando o comando for concluído

Rômulo Silva de Oliveira 16/12/02

22

SendASPI32Command

- O terceiro método, mais elegante, preferido:
 - Método de notificação de eventos
- Um "evento" é criado e atribuído ao campo "SRB_PostProc" do SRB do comando assíncrono
 - Reseta o evento
 - Chama SendASPI32Command
 - Aplicação espera: WaitForSingleObject
- Quando o comando ASPI conclui
 - o evento é sinalizado
 - Aplicação pode testar o "ASPI_Status byte" para determinar o resultado final da execução do comando
 - Apropriado para usar com threads

Rômulo Silva de Oliveira 16/12/02

23

Comando Exec SCSI2

- O comando "Exec SCSI2 command" é assíncrono
- Comando ASPI usado para enviar comandos SCSI2 aos dispositivos SCSI2
- Comandos SCSI2 divididos em 3 categorias
 - Comandos que NÃO são seguidos por transferências de dados
 - Comandos que SÃO seguidos por transferência de dados do dispositivo para o host
 - Comandos SCSI como READ, MODE SENSE, etc.
 - Comandos que SÃO seguidos por transferência de dados do host para o dispositivo
 - Comandos SCSI como WRITE, MODE SELECT, etc.

Rômulo Silva de Oliveira 16/12/02

24

SRB do Comando “Exec SCSI2”

```
typedef struct
{
    BYTE SRB_Cmd;           // ASPI command code = SC_EXEC_SCSI_CMD
    BYTE SRB_Status;        // ASPI command status byte
    BYTE SRB_Hald;          // ASPI host adapter number
    BYTE SRB_Flags;         // ASPI request flags
    DWORD SRB_Hdr_Rsvd;     // Reserved
    BYTE SRB_Target;        // Target's SCSI ID
    BYTE SRB_Lun;           // Target's LUN number
    WORD SRB_Rsvd1;         // Reserved for Alignment
    DWORD SRB_BufLen;       // Data Allocation Length
    ...
}
SRB_ExecSCSICmd, *PSRB_ExecSCSICmd, FAR *LPSRB_ExecSCSICmd;
```

Rômulo Silva de Oliveira 16/12/02

25

SRB do Comando “Exec SCSI2”

```
typedef struct
{
    ...
    BYTE FAR *SRB_BufPointer; // Data Buffer Pointer
    BYTE SRB_SenseLen;        // Sense Allocation Length
    BYTE SRB_CDBLen;          // CDB Length
    BYTE SRB_HaStat;          // Host Adapter Status
    BYTE SRB_TargStat;        // Target Status
    VOID FAR *SRB_PostProc;   // Post routine
    BYTE SRB_Rsvd2[20];       // Reserved, MUST = 0
    BYTE CDBByte[16];         // SCSI CDB
    BYTE SenseArea[SENSE_LEN+2]; // Request Sense buffer
}
SRB_ExecSCSICmd, *PSRB_ExecSCSICmd, FAR *LPSRB_ExecSCSICmd;
```

Rômulo Silva de Oliveira 16/12/02

26

SRB do Comando “Exec SCSI2”

- SRB_Cmd
 - ASPI command code: colocar aqui SC_EXEC_SCSI_CMD
- SRB_Status
 - ASPI command status byte: colocar zero
 - No retorno ele vai conter o status dado pelo gerente ASPI
 - SS_PENDING (0x00): o comando ASPI está em andamento
 - SS_COMP (0x01): completou com sucesso
 - SS_ABORTED (0x02): comando ASPI abortado em função de eventos externos
 - SS_ERR (0x04): um erro ocorreu em função do host adapter, dispositivo, ou barramento SCSI, detalhes em SRB_TargStat e SRB_HaStat

Rômulo Silva de Oliveira 16/12/02

27

SRB do Comando “Exec SCSI2”

- SRB_Status: ASPI command status byte
 - No retorno contem o status dado pelo gerente ASPI
 - SS_INVALID_CMD (0x80): SRB_Cmd invalido
 - SS_INVALID_HA (0x81): número de host adapter inválido
 - SS_NO_DEVICE (0x82): não existe dispositivo no SCSI-ID/LUN informado
 - SS_INVALID_SRB (0xE0): Request Block mal formado
 - SS_BUFFER_ALIGN (0xE1): don't know
 - SS_ILLEGAL_MODE (0xE2): tentativa de usar ASPI for Win32 em ambiente 16-bit

Rômulo Silva de Oliveira 16/12/02

28

SRB do Comando “Exec SCSI2”

- SRB_Status: ASPI command status byte
 - No retorno contem o status dado pelo gerente ASPI
 - SS_NO_ASPI (0xE3): suporte ASPI não instalado corretamente
 - SS_FAILED_INIT (0xE4): Falha geral interna ASPI
 - SS_ASPI_IS_BUSY (0xE5): Recursos insuficientes para completar o comando SendASPI32Command
 - SS_BUFFER_TOO_BIG (0xE6): don't know
 - SS_MISMATCHED_COMPONENTS (0xE7): Suporte ASPI não instalado corretamente
 - SS_NO_ADAPTERS (0xE8): suporte ASPI não reconhece nenhum host adapter conectado no barramento do PC
 - SS_INSUFFICIENT_RESOURCES (0xE9): Recursos do sistema baixos para inicializar o ASPI

Rômulo Silva de Oliveira 16/12/02

29

SRB do Comando “Exec SCSI2”

- SRB_Hald – Número ASPI do host adapter
 - Colocar o número do host adapter que o dispositivo alvo está conectado
 - O número é atribuído pelo gerenciador ASPI para cada SCSI2 host adapted

Rômulo Silva de Oliveira 16/12/02

30

SRB do Comando “Exec SCSI2”

- SRB_Flags - ASPI request flags (OR de várias)
 - SRB_DIR SCSI (0x00): don't know
 - SRB_POSTING (0x01):
 - Habilita ASPI posting (chamadas assíncronas)
 - SRB_ENABLE_RESIDUAL_COUNT (0x04):
 - Habilita relatório de contagem de bytes residuais
 - SRB_DIR_IN (0x08):
 - Transferência de dados para o host
 - SRB_DIR_OUT (0x10):
 - Transferência de dados para o dispositivo
 - SRB_EVENT_NOTIFY (0x40):
 - Habilita ASPI event notification (chamadas assíncronas)

Rômulo Silva de Oliveira 16/12/02

31

SRB do Comando “Exec SCSI2”

- SRB_Hdr_Rsvd
 - Reservado, colocar zero
- SRB_Target
 - ID SCSI do dispositivo
- SRB_Lun
 - Número LUN do dispositivo
- SRB_Rsvd1
 - Reservado para alinhamento, colocar zero

Rômulo Silva de Oliveira 16/12/02

32

SRB do Comando “Exec SCSI2”

- SRB_BufLen
 - Data Allocation Length
 - Colocar zero a não ser que o comando SCSI2 seja seguido de uma transferência de dados do dispositivo para o host ou do host para o dispositivo
 - Neste caso, número de bytes de dados que é esperado na transferência
- FAR *SRB_BufPointer
 - Data Buffer Pointer
 - Idem anterior, aponta o buffer a ser usado na transferência de dados

Rômulo Silva de Oliveira 16/12/02

33

SRB do Comando “Exec SCSI2”

- SRB_SenseLen - Sense Allocation Length:
 - Colocar SENSE_LEN (as defined in WNASPI32.H or WNASPI32.INC depending on the programming language you are using, C++ or ASM)
- SRB_CDBLen - CDB Length:
 - Colocar o tamanho do SCSI2 Command Descriptor Block
 - Comandos SCSI2 são usualmente de tamanho 6, 10 ou 12 bytes

Rômulo Silva de Oliveira 16/12/02

34

SRB do Comando “Exec SCSI2”

- SRB_HaStat - Host Adapter Status
 - Colocar zero
 - No retorno terá o ASPI status
- HASTAT_OK (0x00): host adapter status OK (no error)
- HASTAT_TIMEOUT (0x09): bus transaction timeout
- HASTAT_COMMAND_TIMEOUT (0x0B): SRB expired while waiting to be processed
- HASTAT_MESSAGE_REJECT (0x0D): MESSAGE REJECTED received while processing the SRB
- HASTAT_BUS_RESET (0x0E): unexpected bus reset

Rômulo Silva de Oliveira 16/12/02

35

SRB do Comando “Exec SCSI2”

- HASTAT_PARITY_ERROR (0x0F): parity error occurred
- HASTAT_REQUEST_SENSE_FAILED (0x10): REQUEST SENSE failed
- HASTAT_SEL_TO (0x11): selection of the target timed out
- HASTAT_DO_DU (0x12): data underrun or data overrun
- HASTAT_BUS_FREE (0x13): unexpected bus free
- HASTAT_PHASE_ERR (0x14): bus phase sequence failure

Rômulo Silva de Oliveira 16/12/02

36

SBR do Comando “Exec SCSI2”

- SRB_TargStat - Target Status
 - Colocar zero
 - No retorno contém ASPI status:
 - STATUS_GOOD (0x00): target status OK (no error)
 - STATUS_CHKCOND (0x02): target check condition occurred; the sense data is automatically retrieved for you by the ASPI manager; check it in the SenseArea of the SRB
 - STATUS_BUSY (0x08): specified target/LUN is busy
 - STATUS_RESCONF (0x18): target reservation conflict

Rômulo Silva de Oliveira 16/12/02

37

SBR do Comando “Exec SCSI2”

- FAR *SRB_PostProc - Post routine
 - Aponta para a “callback procedure” ou
 - Para o evento a ser emitido
- SRB_Rsvd2[20]
 - Reserved: colocar zeros
- CDBByte[16] - SCSI CDB
 - Este é o lugar onde vai o SCSI2 Command Descriptor Block
 - Pode ter qualquer tamanho, SRB_CDBLen
 - Descreve o comando SCSI2 a ser enviado
- SenseArea[SENSE_LEN+2] - Request Sense buffer

Rômulo Silva de Oliveira 16/12/02

38