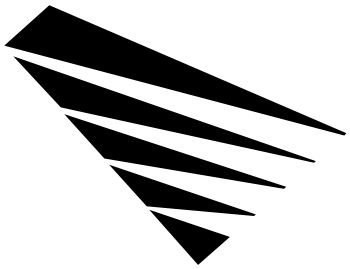

Chapter 5
Real-Mode DOS APIs

Get NIOS Real Mode API 140
DosInvokeCNlmApiHandler 141
DosInvokeRegNlmApiHandler 142
DosNiosFarCallHandler 143
RM_NIOS_BEGIN_USE_API 144
RM_NIOS_COPY_MEM 145
RM_NIOS_COPY_STRING 146
RM_NIOS_END_USE_API 147
RM_NIOS_MAP 148
RM_NIOS_UNMAP 150

Get NIOS Real Mode API

Description

The following steps are used to gain access to NIOS APIs available to real-mode applications. Using these APIs provides, among other things, a method to invoke most exported NLM APIs from real mode.

To locate the NIOS real-mode interfaces, issue an Int 2Fh as shown below. If AX returns set to 0000h, then NIOS is loaded.

On Entry

ax 0D8C1h

On Return

ax 0000h (NIOS is present)

bx Version of loaded NIOS module: bh has major version, bl has minor version

esi Seg:Off of NIOS Far Call Handler (refer to **DosNiosFarCallHandler** for more info)

ecx Seg:Off of NIOS function used to invoke "C" callable NLM functions (refer to **DosInvokeCNlmApiHandler** for more info)

edx Seg:Off of NIOS function used to invoke register-based NLM functions (refer to **DosInvokeRegNlmApiHandler** for more info)

All other registers preserved

See Also

DosNiosFarCallHandler
DosInvokeRegNlmApiHandler
DosInvokeCNlmApiHandler

DosInvokeCNlmApiHandler

Description

Real-mode applications use the **DosInvokeCNlmApiHandler** function to call (invoke) an exported NLM function that uses the "C" calling conventions.

Syntax

```
(*DosInvokeCNlmApiHandler)
    (UINT32   apiAddress,
    UINT32    apiParmCount,
    ...);
```

Parameters

apiAddress Address of NLM API to invoke. This value is obtained from the **RM_NIOS_RESOLVE_NLM_API** function.

apiParmCount Number of UINT32 stack parameters needed for call. This value defines the number of UINT32 values that need to be copied from the real-mode stack onto the protected-mode stack prior to invoking the specified NLM API.

... Parameters to NLM API.

Returns

Defined by NLM API
UINT32 values are returned in registers DX:AX

Remarks

The **DosInvokeCNlmApiHandler** far call address is obtained using the procedure outlined in the **Get NIOS Real Mode API** entry above.

Data pointer parameters to invoked NLM APIs must have been previously mapped using the RM_NIOS_MAP function before invoking the NLM API.

See Also

Get NIOS Real Mode API
DosInvokeRegNlmApiHandler

DosInvokeRegNlmApiHandler

Description

Real-mode applications use the **DosInvokeRegNlmApiHandler** function to call (invoke) an exported NLM function that uses register-based calling conventions.

Assumes

apiAddress	pushed onto the stack
eax,ebx,ecx,edx,esi,edi,ebp	set up as specified for the NLM API

Returns

General purpose regs set up as defined by NLM API
All segment registers preserved
apiAddress is removed from stack

Remarks

Obtain the **DosInvokeRegNlmApiHandler** far call address using the procedure outlined in the *Get NIOS Real Mode API* entry above.

Data pointer parameters to invoked NLM APIs must have been previously mapped using the **RM_NIOS_MAP** function before invoking the NLM API.

See Also

Get NIOS Real Mode API
DosInvokeCNlmApiHandler

DosNiosFarCallHandler

Description

DosNiosFarCallHandler is invoked by real-mode applications using the address obtained using the procedure outlined in the **Get NIOS Real Mode API** entry.

Syntax

```
#include <nlmapi.h>

UINT32
(*DosNiosFarCallHandler)(
    UINT32    function,
    ...);
```

Parameters

function	RM_NIOS_xxxx value (see NLMAPI.H and NLMAPI.INC)
...	Other parameters as needed

Returns

Values specific to each function	
0x80000000	Invalid function request value

Remarks

See Also

RM_NIOS_BEGIN_USE_API

Description

Determines the 32-bit flat linear address of the specified NLM API name. The returned address can then be used with the **DosInvokeCNlmApiHandler** or **DosInvokeRegNlmApiHandler** far call handlers to actually invoke the NLM function from real mode.

Syntax

```
UINT32
(*DosNiosFarCallHandler)(
    UINT32     RM_NIOS_BEGIN_USE_API,
    UINT8 far *apiName);
```

Parameters

apiName Name of the API you would like to call. This is a case-insensitive ASCIIZ string, for example, "CNWIpxSendPacket". Note that this is a segment:offset value.

Returns

0 API does not exist
!0 Linear address of API

Remarks

This function records a dependency for the NLM module in which the API function exists, so it is important that the DOS application use **RM_NIOS_END_USE_API** before the application terminates.

See Also

RM_NIOS_END_USE_API

RM_NIOS_COPY_MEM

Description

Copies the contents of the memory at the specified protected-mode linear address into the specified real-mode buffer for the given length.

Syntax

```
void
(*DosNiosFarCallHandler)(
    UINT32    RM_NIOS_COPY_MEM,
    void      far *destBuffer,
    UINT32    pmBuffer,
    UINT32    length);
```

Parameters

destBuffer Pointer to real-mode buffer to copy to. Note that this is a segment:offset value.

pmBuffer Linear address of protected-mode buffer to copy from

length Number of bytes to copy

Returns

Nothing

Remarks

See Also

RM_NIOS_COPY_STRING

Description

Copies the string pointed to by *pmBuffer* into the specified 16-bit seg:off buffer.

Syntax

```
void
(*DosNiosFarCallHandler)(
    UINT32    RM_NIOS_COPY_STRING,
    void      far *destBuffer,
    UINT32    pmBuffer);
```

Parameters

destBuffer Pointer to seg:off buffer to copy to. Note that this is a segment:offset value.

pmBuffer Linear address of string.

Returns

Nothing

Remarks

See Also

RM_NIOS_END_USE_API

Description Signals that the DOS application is no longer going to use the specified NLM API function. This deletes the dependency previously created using **RM_NIOS_BEGIN_USE_API**.

Syntax

```
void
(*DosNiosFarCallHandler)(
    UINT32    RM_NIOS_END_USE_API,
    UINT32    apiLinAddress);
```

Parameters *apiLinAddress* Linear address of NLM API function

Returns Nothing

Remarks

See Also RM_NIOS_BEGIN_USE_API

RM_NIOS_MAP

Description

RM_NIOS_MAP converts the specified segment:offset value into a flat linear address and locks the memory so that it can be accessed at interrupt time.

Syntax

```
void
*(*DosNiosFarCallHandler)(
    UINT32    RM_NIOS_MAP,
    void      far *segOff,
    UINT32    length);
```

Parameters

segOff Segment:Offset value to lock and return a linear address for

length Length of buffer to map

Returns

Flat linear address (returned in dx:ax)

Remarks

The returned pointer can be used as a pointer value parameter to any NLM function.

A DOS application can convert seg:off parameters to flat linear addresses directly if the NLM API function they intend to call does not require page locked memory. Converting seg:off pointers to linear addresses is accomplished by using the following formula ((UINT32)(seg << 4) + off).

Memory that is mapped using this function must be subsequently unmapped when the memory is no longer needed, such as when the real-mode application terminates. It is extremely important that mapped memory be unmapped.

For time-critical operations, the caller should use this function to lock and obtain the appropriate linear addresses during its initialization, and then pass the obtained values directly when invoking the NLM API, instead of calling the MAP/UNMAP functions each time the NLM API is invoked.

The reason is that this function is executed in protected mode and therefore incurs the overhead of a processor mode switch every time it is called, in addition to the memory locking/unlocking overhead.

See Also (*DosNiosFarCallHandler)(RM_NIOS_UNMAP)

RM_NIOS_UNMAP

Description

Unmaps (unlocks) a buffer that was previously locked with the **RM_NIOS_MAP** function.

Syntax

```
void
(*DosNiosFarCallHandler)(
    UINT32    RM_NIOS_UNMAP,
    UINT32    address,
    UINT32    length);
```

Parameters

address Linear address to unmap

length Length of buffer to unmap

Returns

Nothing

Remarks

Memory that is mapped using the **RM_NIOS_MAP** function must be subsequently unmapped when it's no longer going to be used, such as when the real-mode application terminates.

See Also

(*DosNiosFarCallHandler)(RM_NIOS_MAP)