Console Input

```
; Read From the Console (ReadConsole.asm)
INCLUDE Irvine32.inc

BufSize = 80

.data
buffer BYTE BufSize DUP(?), 0, 0
stdInHandle HANDLE ?
bytesRead DWORD ?

.code
main PROC
    ; Get handle to standard input
    INVOKE GetStdHandle, STD_INPUT_HANDLE
    mov stdInHandle, eax

    ; Wait for user input
    INVOKE ReadConsole, stdInHandle, ADDR buffer, BufSize, ADDR bytesRead, 0

    ; Display the buffer
    mov esi, OFFSET buffer
    mov ecx, bytesRead
    mov ebx, TYPE buffer
    call DumpMem

    exit
main ENDP

END main
```

Console Input Buffer: In Win32 console programming, there exists an input buffer that stores input event records. These input events include keystrokes, mouse movements, and mouse-button clicks.

High-level input functions like ReadConsole process this input data and return a stream of characters to the program.

ReadConsole Function: The ReadConsole function is a Win32 API function used to read text input from the console and store it in a buffer.

It takes several parameters, including the console input handle, a pointer to a character buffer, the number of characters to read, a pointer to store the count of characters read, and a reserved

parameter.

The provided code is an example program that demonstrates how to use the ReadConsole function to read characters entered by the user in a console application.

It starts by defining the size of the buffer (BufSize) and declaring the necessary data variables, including a buffer for storing the input, a handle for standard input (stdInHandle), and a variable for the number of bytes read (bytesRead).

In the main procedure, it retrieves the standard input handle using the GetStdHandle function, which returns a handle to the standard input.

It then calls the ReadConsole function to read input from the user.

The function parameters include the standard input handle, the buffer for storing input, the maximum number of characters to read, a pointer to store the count of characters read, and a value of 0 for the reserved parameter.

After reading the input, it displays the content of the buffer using the DumpMem function, which is part of the Irvine32 library.

The DumpMem function is used to display the buffer's content in both hexadecimal and ASCII representations.

The program can read and display user input, including any end-of-line characters (0Dh and 0Ah) inserted when the user presses the Enter key.

In summary, this code demonstrates how to read and display user input from the console using the ReadConsole function in a Win32 console application. It provides an example of handling console input in assembly language.