Microprocessors (0630371)
Fall 2010/2011 – Lecture Notes # 16

PROCEDURES

Objectives of the Lecture

- **Link Library Overview**
- **Calling a Library Procedure.**
- **Linking to a Library**
- **Library Procedures – Overview**
- **Examples**
- **Programming Examples**

Link Library Overview

- A **file** containing **procedures** that have been compiled into machine code.
  - constructed from one or more **OBJ** files
- To build a library, . . .
  - start with one or more ASM source files
  - assemble each into an OBJ file
  - create an empty library file (extension .LIB)
  - add the OBJ file(s) to the library file, using the **Microsoft LIB utility**

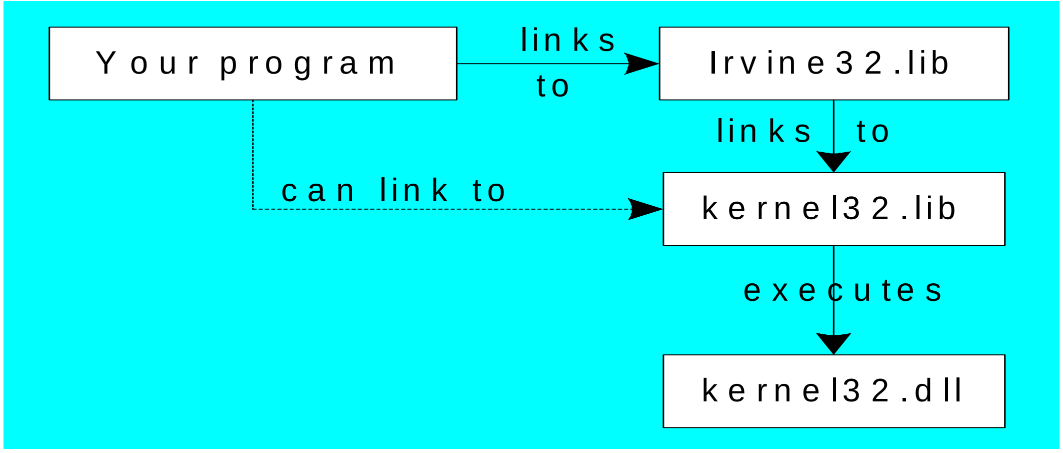
Calling a Library Procedure

- Call a library procedure using the **CALL** instruction. Some procedures require **input arguments**. The **INCLUDE** directive copies in the procedure prototypes (declarations).
- The following example displays **"1234"** on the console:

```
INCLUDE Irvine32.inc
.code
     mov eax,1234h  ; input argument
     call WriteHex  ; show hex number
     call Crlf      ; end of line
```

Linking to a Library

- Your programs link to **Irvine32.lib** using the linker command inside a batch file named **make32.bat**.
- Notice the two LIB files: **Irvine32.lib, and kernel32.lib**
  - the **kernel32.lib** is part of the **Microsoft *Win32 Software Development Kit (SDK)***



Library Procedures – Overview

- **Clrscr** *- Clears the console and locates the cursor at the upper left corner.
- **Crlf** - Writes an end of line sequence to standard output.
- **DumpMem** - Writes a block of memory to standard output in hexadecimal.
- **DumpRegs** - Displays the EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP, EFLAGS, and EIP registers in hexadecimal. Also displays the Carry, Sign, Zero, and Overflow flags.
- **ReadChar** - Reads a single character from standard input.
- **ReadHex** - Reads a 32-bit hexadecimal integer from standard input, terminated by the Enter key.
- **ReadInt** - Reads a 32-bit signed decimal integer from standard input, terminated by the Enter key.
- **ReadString** - Reads a string from standard input, terminated by the Enter key.
- **WriteBin** - Writes an unsigned 32-bit integer to standard output in ASCII binary format.
- **WriteChar** - Writes a single character to standard output.
- **WriteDec** - Writes an unsigned 32-bit integer to standard output in decimal format.
- **WriteHex** - Writes an unsigned 32-bit integer to standard output in hexadecimal format.
- **WriteInt** - Writes a signed 32-bit integer to standard output in decimal format.
- **WriteString** - Writes a null-terminated string to standard output.
- **Delay** - Pauses the program execution for a specified *n* millisecond interval.
- **GetCommandtail** - Copies the program's command-line arguments (called the *command tail*) into an array of bytes.
- **GetMseconds** - Returns the number of milliseconds that have elapsed since midnight.
- **Gotoxy** - Locates cursor at row and column on the console.
- **Random32** - Generates a 32-bit pseudorandom integer in the range 0 to FFFFFFFFh.
- **Randomize** - Seeds the random number generator.
- **RandomRange** - Generates a pseudorandom integer within a specified range.
- **SetTextColor** - Sets the foreground and background colors of all subsequent text output to the console.
- **WaitMsg** - Displays message, waits for Enter key to be pressed.

*** Green color procedures must know how to use**

Examples

Example 1

Clear the screen, delay the program for 500 milliseconds, and dump the registers and flags.

```
INCLUDE Irvine32.inc
.code
     call Clrscr
     mov  eax,500
     call Delay
     call DumpRegs
```

Sample output:

```
EAX=00000613 EBX=00000000 ECX=000000FF EDX=00000000
ESI=00000000 EDI=00000100 EBP=0000091E ESP=000000F6
EIP=00401026 EFL=00000286 CF=0 SF=1 ZF=0 OF=0
```

Example 2a

Display a null-terminated string and move the cursor to the beginning of the next screen line.

```
INCLUDE Irvine32.inc
.data
str1 BYTE "Assembly language is easy!",0
.code
     mov  edx,OFFSET str1
     call WriteString
     call Crlf
```

Example 2b

Display a null-terminated string and move the cursor to the beginning of the next screen line (use embedded CR/LF)

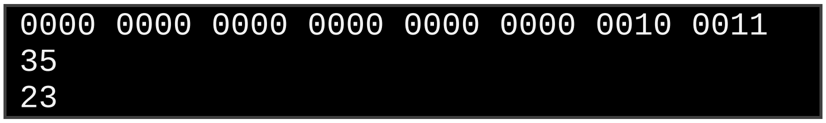
```
INCLUDE Irvine32.inc
.data
str1 BYTE "Assembly language is easy!",0Dh,0Ah,0
.code
     mov  edx,OFFSET str1
     call WriteString
```

Example 3

Display an unsigned integer in binary, decimal, and hexadecimal, each on a separate line.

```
INCLUDE Irvine32.inc
IntVal = 35
.code
     mov  eax,IntVal
     call WriteBin  ; display binary
     call Crlf
     call WriteDec  ; display decimal
     call Crlf
     call WriteHex  ; display hexadecimal
     call Crlf
```

Sample output:



```
0000 0000 0000 0000 0000 0000 0010 0011
35
23
```

Example 4

Input a string from the user. EDX points to the string and ECX specifies the maximum number of characters the user is permitted to enter.

```
INCLUDE Irvine32.inc
.data
fileName BYTE 80 DUP(0)
.code
     mov edx,OFFSET fileName
     mov ecx,SIZEOF fileName – 1
     call ReadString
```

A null byte is automatically appended to the string.

Example 5

Generate and display ten pseudorandom signed integers in the range 0 – 99. Pass each integer to **WriteInt** in EAX and display it on a separate line.

```
INCLUDE Irvine32.inc
.code
     mov ecx,10          ; loop counter
L1:  mov  eax,100        ; ceiling value
     call RandomRange    ; generate random int
     call WriteInt       ; display signed int
     call Crlf           ; goto next display line
     loop L1             ; repeat loop
```

Example 6

Display a null-terminated string with yellow characters on a blue background.

```
INCLUDE Irvine32.inc
.data
str1 BYTE "Color output is easy!",0
.code
     mov  eax,yellow + (blue * 16)
     call SetTextColor
     mov  edx,OFFSET str1
     call WriteString
     call Crlf
```

The background color is multiplied by 16 before being added to the foreground color.

Programming Examples

Programming Example 1

```
TITLE Program Template     (Stack.asm)
; Testing PUSH and POP, 16-bit and 32-bit operands
Include Irvine32.inc
.code
main PROC
     call DumpRegs
     push 1
     call DumpRegs
     push eax
     call DumpRegs
     push eax
     call DumpRegs
     exit
main ENDP
END main
```

Programming Example 2

```
TITLE Library Test #1: Integer I/O   (TestLib1.asm)
; Tests the Clrscr, Crlf, DumpMem, ReadInt,
; SetTextColor, WaitMsg, WriteBin, WriteHex,
; and WriteString procedures.
INCLUDE Irvine32.inc
.data
arrayD     DWORD 1000h,2000h,3000h
prompt1    BYTE "Enter a 32-bit signed integer: ",0
dwordVal   DWORD ?
.code
main PROC
; Set text color to yellow text on blue background:
     mov  eax,yellow + (blue * 16)
     call SetTextColor
     call Clrscr              ; clear the screen
; Display the array using DumpMem.
     mov  esi,OFFSET arrayD   ; starting OFFSET
     mov  ecx,LENGTHOF arrayD ; number of units in dwordVal
     mov  ebx,TYPE arrayD     ; size of a doubleword
     call DumpMem             ; display memory
     call Crlf                ; new line
; Ask the user to input a signed decimal integer.
     mov  edx,OFFSET prompt1
     call WriteString
     call ReadInt             ; input the integer
     mov  dwordVal,eax        ; save in a variable
; Display the integer in decimal, hexadecimal, and binary.
     call Crlf                ; new line
     call WriteInt            ; display in signed decimal
     call Crlf
     call WriteHex            ; display in hexadecimal
     call Crlf
     call WriteBin            ; display in binary
     call Crlf
     call WaitMsg             ; "Press any key..."
; Return console window to default colors.
     mov  eax,lightGray + (black * 16)
     call SetTextColor
     call Clrscr
     exit
main ENDP
END main
```

Programming Example 3

```
TITLE Link Library Test #2    (TestLib2.asm)
; Testing the Irvine32 Library procedures.
INCLUDE Irvine32.inc
TAB = 9        ; ASCII code for Tab
.code
main PROC
     call Randomize      ; init random generator
     call Rand1
     call Rand2
     exit
main ENDP
Rand1 PROC
; Generate ten pseudo-random integers.
     mov  ecx,10         ; loop 10 times
L1:  call Random32       ; generate random int
     call WriteDec       ; write in unsigned decimal
     mov  al,TAB         ; horizontal tab
     call WriteChar      ; write the tab
     loop L1
     call Crlf
     ret
Rand1 ENDP
Rand2 PROC
; Generate ten pseudo-random integers between -50 and +49
     mov  ecx,10         ; loop 10 times
L1:  mov  eax,100        ; values 0-99
     call RandomRange    ; generate random int
     sub  eax,50         ; vaues -50 to +49
     call WriteInt       ; write signed decimal
     mov  al,TAB         ; horizontal tab
     call WriteChar      ; write the tab
     loop L1
     call Crlf
     ret
Rand2 ENDP
END main
```