ECE291

Lecture 10

Interrupt examples and exam review

Lecture outline

- INT 21h
- INT 16h
- Instruction review
- Exam review

INT 21h

- DOS provides **INT 21h**, which is called the DOS function dispatcher and supports functions such as: read from the keyboard, write to the screen, write to the printer, read and write to disk files, etc.
- INT 21h must be told which function is being requested
  - this information is passed by placing the function number in the **AH register**
  - depending on the function being used, other information may be needed

INT 21h, AH = 02h

- DOS 1+ - WRITE CHARACTER TO STANDARD OUTPUT

- AH = 02h
- DL = character to write

- Return: AL = last character output (despite the official docs which state nothing is returned)

- Notes: ^C/^Break are checked, and INT 23 executed if pressed. Standard output is always the screen under DOS 1.x, but may be redirected under DOS 2+. The last character output will be the character in DL unless DL=09h on entry, in which case AL=20h as tabs are expanded to blanks. If standard output is redirected to a file, no error checks (write- protected, full media, etc.) are performed

INT 21h, AH = 6

- DOS 1+ - DIRECT CONSOLE OUTPUT

- AH = 06h
- DL = character to output (except FFh)

- Return: AL = character output (despite official docs which state nothing is returned) (at least DOS 2.1-7.0)

- Notes: Does not check ^C/^Break. Writes to standard output, which is always the screen under DOS 1.x, but may be redirected under DOS 2+

Displaying A Single Character Using INT 21h

Example:

- Suppose that the letter ‘A’ is to be printed to the screen at the current cursor position
- Can use function 02h or 06h
- INT 21h must also be told which letter to print
  - the ASCII code must be placed into DL register

```
MOV     DL, ‘A’
MOV     AH, 06h
INT     21h
```

INT 21h, AH = 09h

- DOS 1+ - WRITE STRING TO STANDARD OUTPUT

- AH = 09h
- DS:DX -> '$'-terminated string

- Return: AL = 24h (the '$' terminating the string, despite official docs which state that nothing is returned)

- Notes: ^C/^Break are checked, and INT 23 is called if either pressed. Standard output is always the screen under DOS 1.x, but may be redirected under DOS 2+. Under the FlashTek X-32 DOS extender, the pointer is in DS:EDX

Displaying A Character String Using INT 21h

- DOS function 09h displays a character string that ends with ‘$’

```
MSG   DB      ‘This is a test line’,’$’
…
MOV   DX, MSG
MOV   AH, 09h
INT   21h
```

- The string will be printed beginning at the current cursor position

INT 21h, AH = 01h

- DOS 1+ - READ CHARACTER FROM STD INPUT, WITH ECHO
- AH = 01h
- Return: AL = character read
- Notes: ^C/^Break are checked, and INT 23 executed if read. ^P toggles the DOS-internal echo-to-printer flag. ^Z is not interpreted, thus not causing an EOF if input is redirected. Character is echoed to standard output. Standard input is always the keyboard and standard output the screen under DOS 1.x, but they may be redirected under DOS 2+

Reading A Single Character Using INT 21h

- DOS function 01h reads a single character typed from the keyboard and echoes that character to the screen

```
MOV   AH, 01h
INT   21h
```

- The computer will wait until the user presses a key, then input character will be in AL

INT 10h

- The DOS function calls allow a key to be read and a character to be displayed but the cursor is difficult to position at a specific location on the screen.
- The BIOS function calls allow more control over the video display and require less time to execute than the DOS function calls
- BIOS provides interrupt **INT 10h**, known as the video interrupt, which gives access to various functions to control video screen
- Before we place information on the screen we should get the position of the cursor:

  function 03h reads cursor position (DH=Row, DL=Column, BH=Page #)

  function 02h sets cursor position (DH=Row, DL=Column, BH=Page #)

INT 10h

- Function 0Fh finds the number of the active page
  - the page number is returned in the BH register
- The cursor position assumes that
  - the left hand page column is column 0 progressing across a line to column 79
  - the row number corresponds to the character line number on the screen, the top line being line 0

Monolithic example

http://www.ece.uiuc.edu/ece291/lecture/lec10ex.asm

http://www.ece.uiuc.edu/ece291/lecture/lec10ex.exe

Instruction review

- Mov
- Arithmetic (add adc sub sbb div idiv mul imul dec inc neg)
- Logic (and or xor not cmp loop)
- Shifting (shl shr sar sal rol ror rcl rcr)
- Procedures (call ret retf int iret)
- Jumping (jmp j– loop loop??)

Exam review

- Registers
  - What they are
  - What they do or are used for
  - How they are accessed
- Memory
  - Its overall structure
  - How it is addressed
  - Segment:offset notation and finding effective addresses
  - What registers can address it
  - What memory access instructions are valid
- Variables/Arrays/Tables
  - How to declare
  - How to access
  - What effect various directives have on memory

Exam review

- Arithmetic and logic
  - How the instructions we looked at work
  - In general what flags they affect
  - Difference between signed and unsigned operands
- Jumping
  - Know how to handle signed vs. unsigned operands—know what flag conditions apply to each
  - Know the different types of jumps (short, near, far, conditional)
  - Be able to evaluate an instruction and determine what a given conditional jump statement will do

Exam review

- Procedures and macros
  - Understand the entire procedure calling mechanism
  - How CALL, FAR CALL, and INT work and how they differ
  - Know the difference between procs and macros
- Interrupts
  - Know what the int vec tab is and where it's located
  - Know how ISR's differ from regular procedures
  - Understand the function of the 8259A programmable interrupt controller

Exam preview

- Structure and organization
  - 120 points evaluation and theory
  - 80 points coding
- Example programming problem
  - Change characters in a string from lower case to upper case
  - Calculate area of a triange