CSC 321: Microprocessor & Assembly Language

Interrupt Driven IO

Outline

- Interrupts
- Input Output Instructions
- Sample Programs

References

- ***Chapter 3, 4,*** Ytha Yu and Charles Marut, "Assembly Language Programming and Organization of IBM PC"
- ***Chapter 3***, Assembly Language for Intel Based-Computers

Interrupts

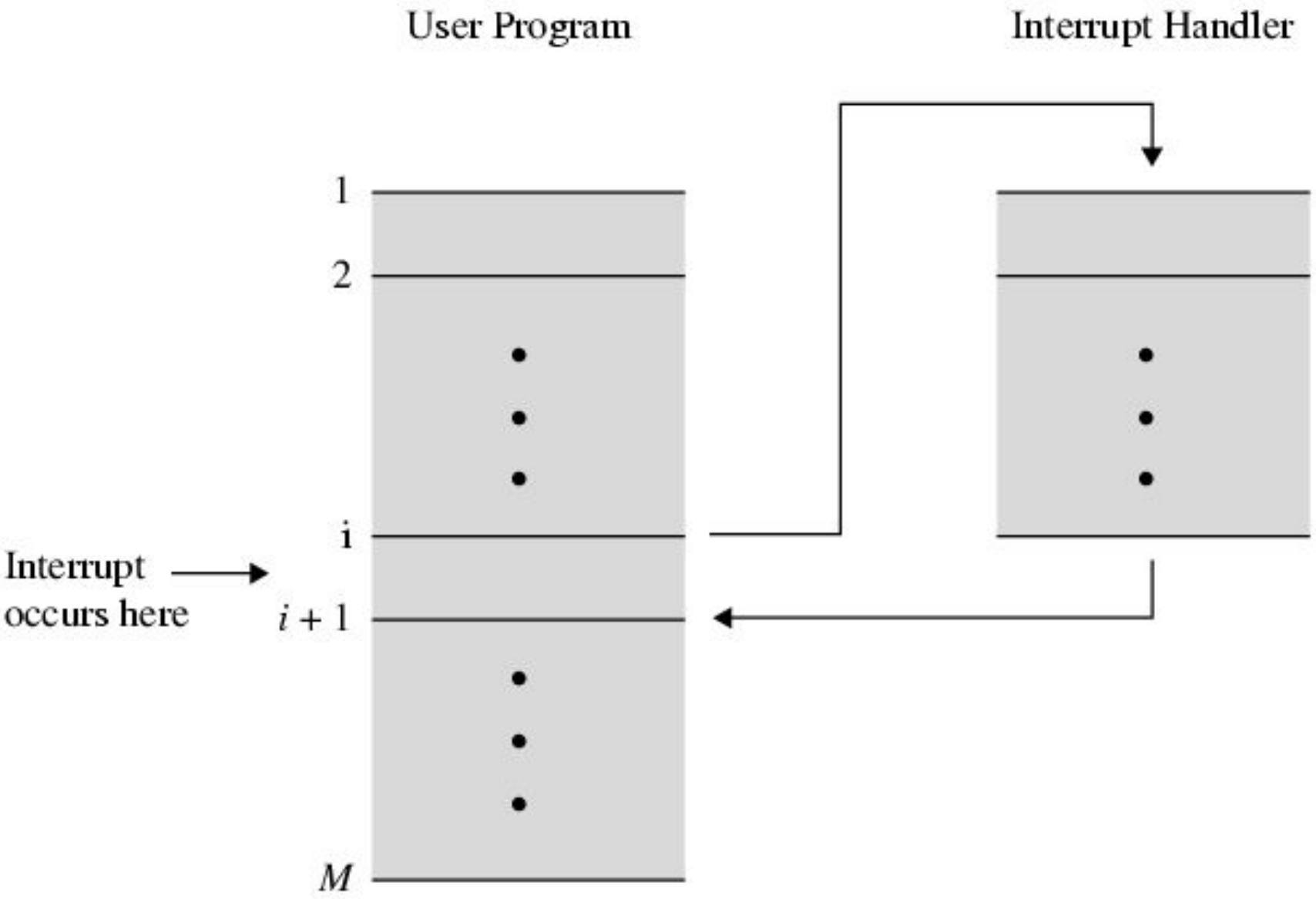
Interrupts – Changing Program Flow

- Mechanism by which other modules (e.g. I/O) may interrupt normal sequence of processing
- Program
  - e.g. overflow, division by zero
- I/O
  - from I/O controller
- Hardware failure

Interrupt Cycle

- Processor checks for interrupt
  - Indicated by an interrupt signal
- If no interrupt, fetch next instruction
- If interrupt pending:
  - Suspend execution of current program
  - Save context
  - Set PC to start address of interrupt handler routine
  - Process interrupt
  - Restore context and continue interrupted program

Transfer of Control via Interrupts



Input and Output Instructions

The INT (Interrupt) instruction

- Syntax:

  INT interrupt_number

  - Where interrupt_number specifies a routine.

- Examples

  INT 10h

  - To enable video mode.

  INT 21h

  - Invoke DOS functions depending on function number present in AH register.

Function No.	Routine
1	Single-key input
2	Single-character output
9	Character string output

Single-Key Input

- AH = 1
- AL = ASCII code if character key is pressed
  = 0 if non-character key is pressed

```
MOV AH,1
INT 21h
```

Single-character output

- AH = 2
- DL = ASCII code of the display character or control character
- AL = ASCII code of the display character or control character

```
MOV AH,2
MOV DL, '?'
INT 21h
```

Control Characters

ASCII Code (Hex)	Symbol	Function
7	BEL	Beep (sound a to e)
8	BS	Backspace
9	HT	Tab
A	LF	Line feed (new line)
D	CR	Carriage return (start of current line)

Sample Programs

Character Input

To read a character from the keyboard:

```
MOV AH, 1
INT 21h
; character is stored in AL
```

Character Output

To display the character ’a’ on the screen:

```
MOV DL, ‘a‘
MOV AH, 2
INT 21h
```

Reading and displaying a character:

```
MOV AH, 1
INT 21h
MOV DL, AL
MOV AH, 2
INT 21h
```

Program 1: Hello World!

```
title Hello World Program          (hello.asm)
; This program displays "Hello, world!"
.model small
.stack 100h
.data
message db "Hello, world!",0dh,0ah,'$'
.code
main proc
    mov  ah,9
    mov  dx,offset message
    int  21h

   main endp
end main
```