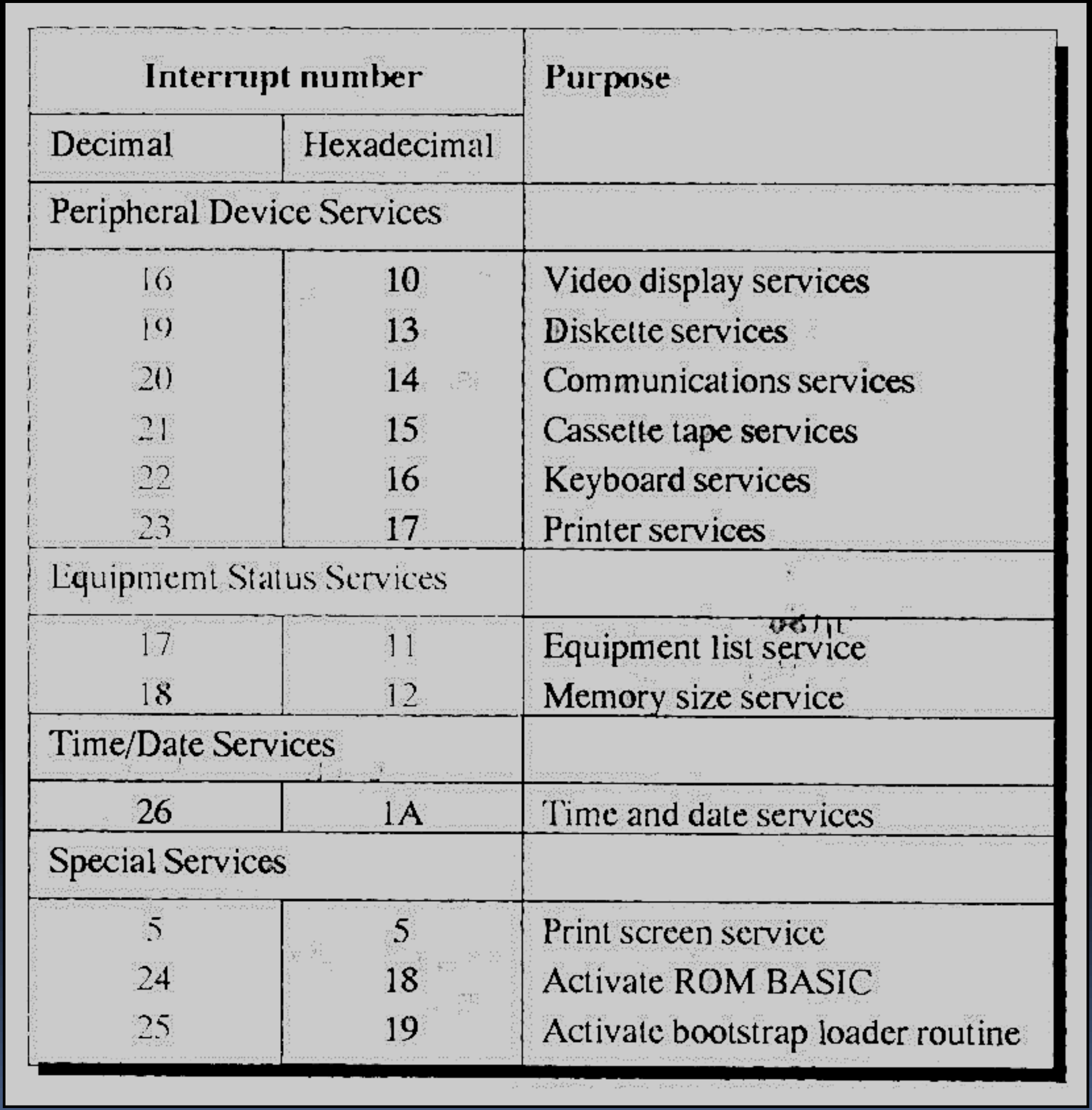
Paper 102: Programming & Problem solving through C

Lecture-26:ROM BIOS Services

Aiusha V.Hujon, Dept. of Computer Science, SAC

ROM-BIOS Services under each interrupt



Interrupt number		Purpose
Decimal	Hexadecimal	
Peripheral Device Services		
16	10	Video display services
19	13	Diskette services
20	14	Communications services
21	15	Cassette tape services
22	16	Keyboard services
23	17	Printer services
Equipmemt Status Services		
17	11	Equipment list service
18	12	Memory size service
Time/Date Services		
26	1A	Time and date services
Special Services		
5	5	Print screen service
24	18	Activate ROM BASIC
25	19	Activate bootstrap loader routine

Setting Cursor Size

ROM-BIOS Routine	Set Cursor Size
Interrupt 10 hex	Video
Input registers	AH=01
	CH=starting scan line (0-13)
	CL=ending scan line (0-13)
Output registers	None

Setting Cursor Size

```
#include<stdio.h>
#include<stdlib.h>
#include<dos.h>
#define CURSIZE 1            // "Set Cursor Size" Service
#define VIDEO 0x10    //video BIOS interrupt number

void main(int argc, char *argv[])
{
   union REGS regs;
   int start, end;
   if (argc !=3){
       printf("Usage example: C>setcur 12 13");
       return;
   }
   start=atoi(argv[1]); //string to integer
   end=atoi(argv[2]);
   regs.h.ch=start;   //starting scan line number
   regs.h.cl=end;     //ending scan line number
   regs.h.ah=CURSIZE; //service number
   int86(VIDEO, &regs, &regs);
}
```

Positioning Cursor on Screen

ROM-BIOS Routine	Set Cursor Position
Interrupt 10 hex	Video

Input registers	AH=02
	DH=row number
	DL=column number

Output registers	None

Positioning Cursor on Screen

```
#include<stdio.h>
#include<stdlib.h>
#include<dos.h>
#define CURPOSN 2             // "Set Cursor Size" Service
#define VIDEO 0x10     //video BIOS interrupt number

void main(int argc, char *argv[])
{
  union REGS regs;
  int start, end;
  if (argc !=3){
       printf("Usage example: C:/>poscur 10 2");
       return;
  }
  start=atoi(argv[1]); //string to integer
  end=atoi(argv[2]);
  regs.h.dh=start;     //row number
  regs.h.dl=end;       //column number
  regs.h.ah=CURPOSN;   //service number
  int86(VIDEO, &regs, &regs);
  printf("Hello, this text starts at row %d and col
  %d",start, end);
}
```

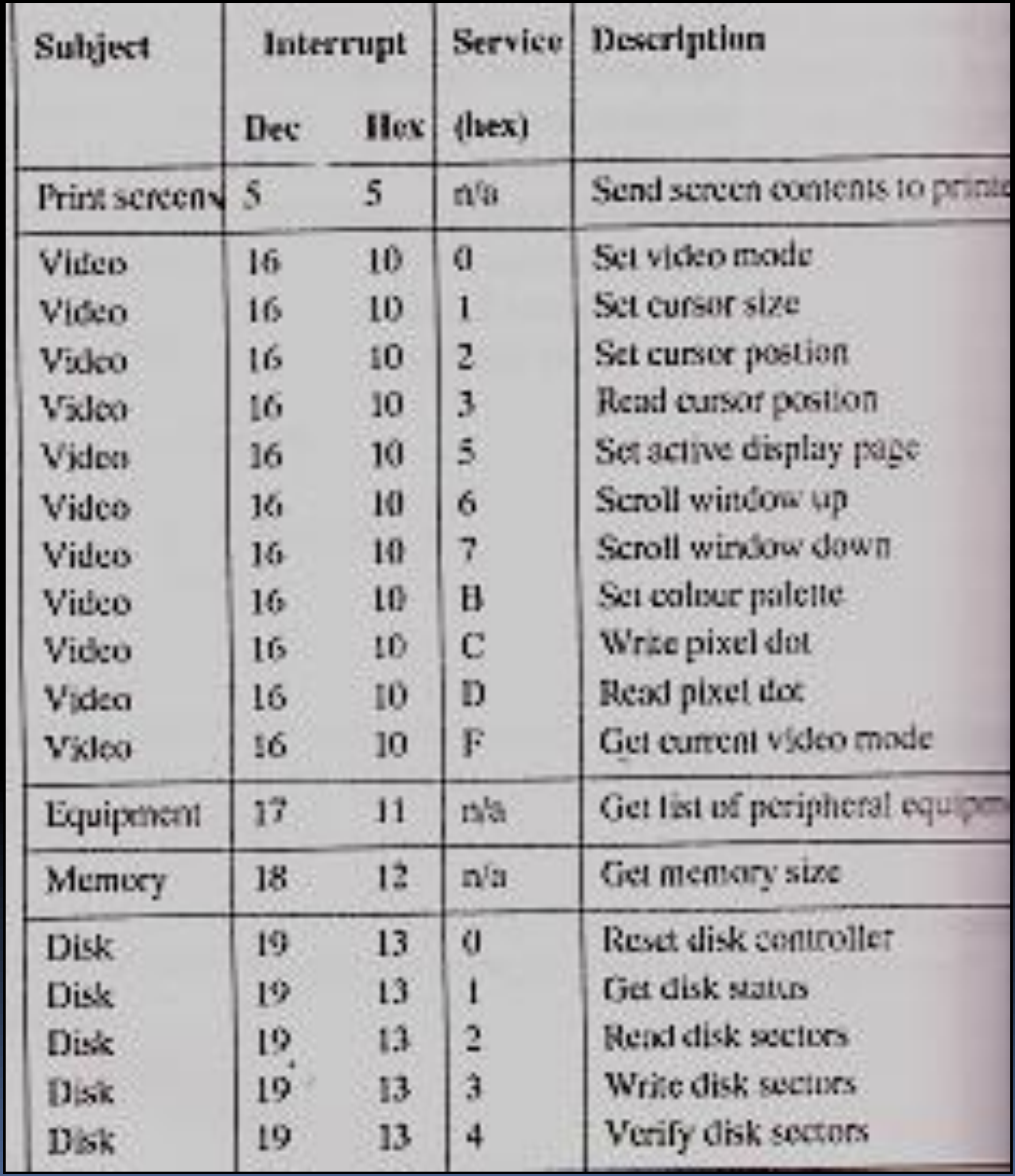
Making Cursor Disappear

- To make cursor disappear, you need to set the 5th bit (counting from 0) in the byte stored in CH register to 1.
- This can be done by placing the hex number 20 in CH register
- Hex 20 is binary 00100000
- ROM-BIOS Routine		Set Cursor Size
- Interrupt 10 hex		Video

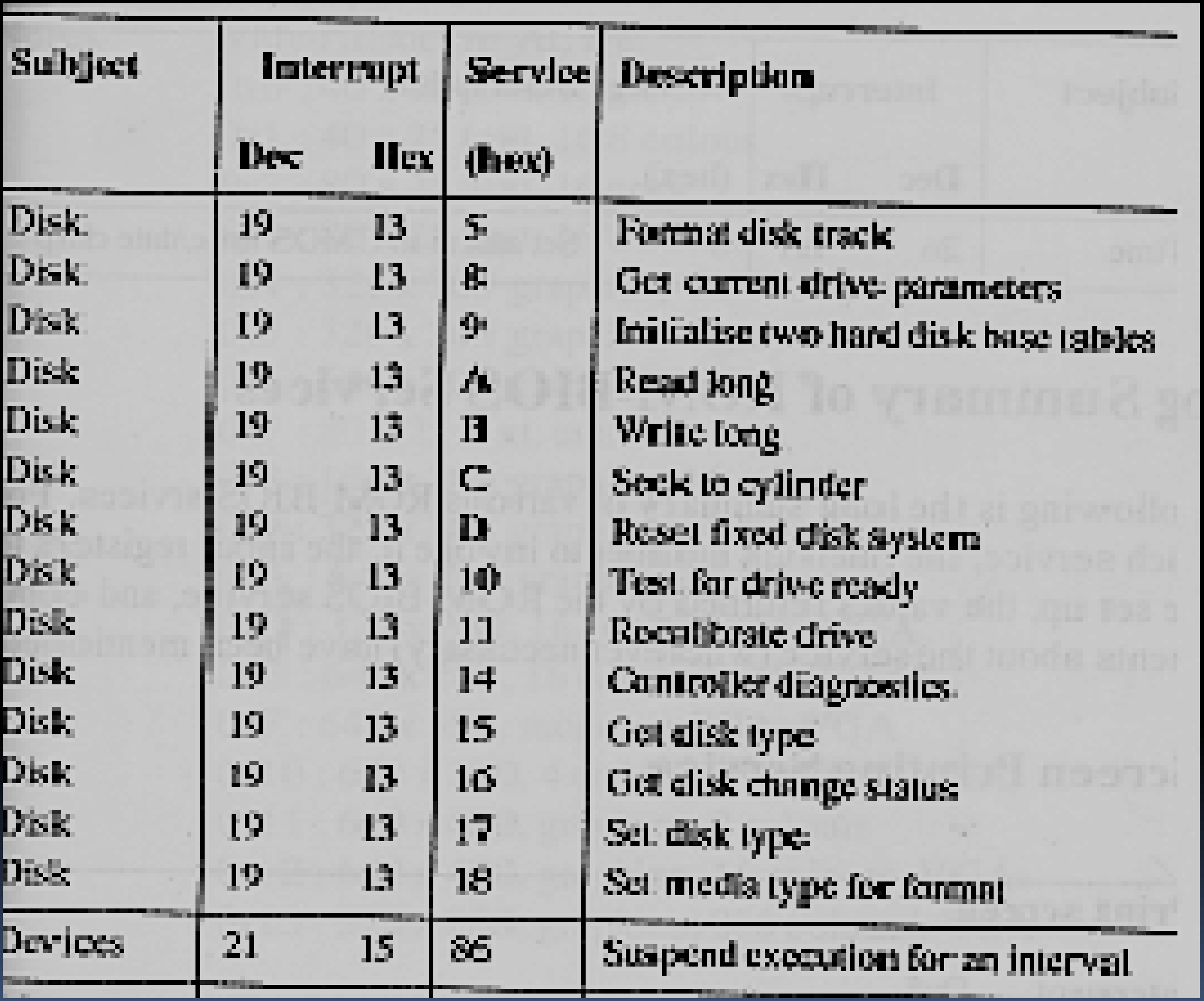
Making Cursor Disappear

```
#include<dos.h>
#define CURSIZE 1   // "Set Cursor Size" Service
#define VIDEO 0x10  //video BIOS interrupt number
#define STOPBIT 0x20//setting 5th bit by placing hex 20
void main(void)
{
   union REGS regs;
   regs.h.ah=CURSIZE;      //service number
   regs.h.ch=STOPBIT;      //turns cursor off
   int86(VIDEO, &regs, &regs);
}
```

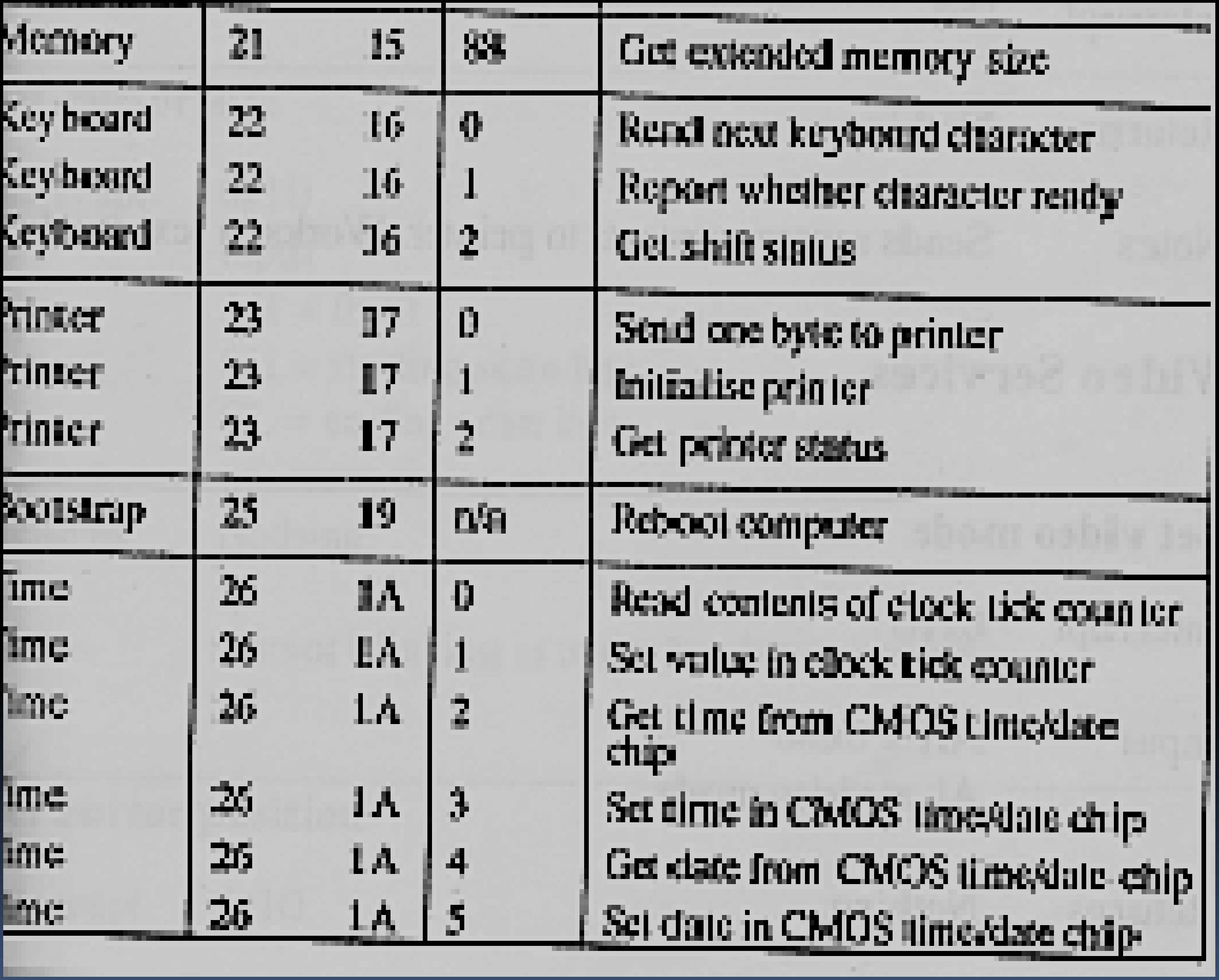
List of Interrupt services



Subject	Interrupt		Service	Description
	Dec	Hex	(hex)	
Print screen	5	5	n/a	Send screen contents to printe
Video	16	10	0	Set video mode
Video	16	10	1	Set cursor size
Video	16	10	2	Set cursor postion
Video	16	10	3	Read cursor position
Video	16	10	5	Set active display page
Video	16	10	6	Scroll window up
Video	16	10	7	Scroll window down
Video	16	10	B	Set colour palette
Video	16	10	C	Write pixel dot
Video	16	10	D	Read pixel dot
Video	16	10	F	Get current video mode
Equipment	17	11	n/a	Get list of peripheral equipm
Memory	18	12	n/a	Get memory size
Disk	19	13	0	Reset disk controller
Disk	19	13	1	Get disk status
Disk	19	13	2	Read disk sectors
Disk	19	13	3	Write disk sectors
Disk	19	13	4	Verify disk sectors



Subject	Interrupt		Service	Description
	Dec	Hex	(hex)	
Disk	19	13	5	Format disk track
Disk	19	13	8	Get current drive parameters
Disk	19	13	9	Initialise two hard disk base tables
Disk	19	13	A	Read long
Disk	19	13	B	Write long
Disk	19	13	C	Seek to cylinder
Disk	19	13	D	Reset fixed disk system
Disk	19	13	10	Test for drive ready
Disk	19	13	11	Recalibrate drive
Disk	19	13	14	Controller diagnostics
Disk	19	13	15	Get disk type
Disk	19	13	16	Get disk change status
Disk	19	13	17	Set disk type
Disk	19	13	18	Set media type for format
Devices	21	15	86	Suspend execution for an interval



Memory	21	15	88	Get extended memory size
Keyboard	22	16	0	Read next keyboard character
Keyboard	22	16	1	Report whether character ready
Keyboard	22	16	2	Get shift status
Printer	23	17	0	Send one byte to printer
Printer	23	17	1	Initialise printer
Printer	23	17	2	Get printer status
Bootstrap	25	19	n/a	Reboot computer
Time	26	1A	0	Read contents of clock tick counter
Time	26	1A	1	Set value in clock tick counter
Time	26	1A	2	Get time from CMOS time/date chip
Time	26	1A	3	Set time in CMOS time/date chip
Time	26	1A	4	Get date from CMOS time/date chip
Time	26	1A	5	Set date in CMOS time/date chip

Read next character from keyboard

- Interrupt 0x16
- Input registers AH=0x00
- Output registers AH=scan code
  AL=ASCII value

Get Shift status

- Interrupt 0x16
- Input registers AH=0x02
- Output registers AL=Shift status bits
- Shift status bits are:

  bit 0=1: right shift depressed
  bit 1=1: left shift depressed
  bit 2=1: ctrl depressed
  bit 3=1: alt depressed
  bit 4=1: scroll lock depressed
  bit 5=1: num lock depressed
  bit 6=1: caps lock depressed
  bit 7=1: insert on

To read key that is pressed from keyboard

```
#include<dos.h>
void main()
{
        union REGS inregs,outregs;
        int scancode, asciicode;

        inregs.h.ah=0x00;
        int86(0x16,&inregs,&outregs);
        scancode=outregs.h.ah;
        asciicode=outregs.h.al;
        printf("\n the key %c having scan code %d is
        pressed",asciicode,scancode);

}
```