

No: 22)

SET VIDEO MODE

26-09-2006

AIM

Write a program to check the printer status.

DESCRIPTION

For setting the video mode ah and al register used. Service number used is 0x00.al register used for holding the mode number.int86() used as interrupt calling function and interrupt number is 0x10.

ALGORITHM

Step 1: Start.

Step 2: Set ah value 0x00.

Step 3: Read mod number from programmer.

Step 4: Set al value is mode value.

Step 5: Call the interrupt function.

Step 6: Print the string on the screen.

Step 7: Stop.

COMPUTER HARDWARE AND NETWORKING LAB

PROGRAM

```
/* Set video mode */
#include<stdio.h>
#include<conio.h>
#include<dos.h>
#include<process.h>
void main()
{
    clrscr();
    union REGS i,o;
    int mod;
    i.h.ah=0x00;
    printf("Read mode value\n");
    scanf("%d",&mod);
    i.h.al=mod;
    int86(0x10,&i,&o);
    printf("Video\n");
    getch();
}
```

OUTPUT

Read mode value

20

Video

RESULT

Program run successfully and required output obtained.

Pamodh

NO:23)

26-09-2006

VIDEO PAGE UPWARD SCROLL

AIM

To write a program to define a window in the screen with corresponding lines in the screen to be scrolled upwards.

DESCRIPTION

In this program, uses the service function number 0x06 to scroll the lines for the described window to be displayed there. This value is passed to the AH register. The interrupt function int86 with the interrupt 0x10 (video interrupt) is called to get the service done. The top-left row and column specifications are get into ch and cl registers respectively. Also lower-right row and column specifications are get into dh and dl registers respectively. As per the row and column specifications a window is displayed on the screen and the lines occupying that space are get scrolled upwards.

ALGORITHM

1. Start
2. Declare the union variables i, o.
3. For scrolling the window the service function used is 0x06 and it is stored in the ah register.
4. Enter the top-left row and column specifications into ch and cl registers respectively.
5. Enter the lower-right row and column specifications into dh and dl registers respectively.
6. Enter the number of lines to be scrolled into the al register.
7. Enter the color of the window into the bh register. 8. Call the interrupt function with interrupt number 0x10.
9. Stop.

PROGRAM

/*VIDEO PAGE UPWARD SCROLL*/

#include<stdio.h>

#include<conio.h>

#include<dos.h>

void main()

{

union REGS i,o;

int m,n,p,q,l,s;

clrscr();

i.h.ah=0x06;

printf("Enter the row specifications: ");

scanf("%d %d",&m,&n);

i.h.ch=m;

i.h.dl=n;

printf("\nEnter the column specifications:");

scanf("%d %d",&p,&q);

i.h.cl=p;

i.h.dh=q;

printf("\nEnter the lines to be scrolled:");

scanf("%d",&s);

i.h.al=s;

printf("\nEnter the color of the window:");

scanf("%d",&l);

i.h.bh=l;

COMPUTER HARDWARE AND NETWORKING LAB

```
int86(0x10,&i,&o);
```

```
getch();
```

```
}
```

RESULT

Video page has been scrolled upwards. Program runs successfully..

✓
Pamali P

No: 30) **RESTRICT THE MOUSE MOVEMENTS**

10-10-2006

AIM

Write a program to restrict the mouse movement.

DESCRIPTION

To restrict the mouse movement with in a specific area service number used is 0x07 for column restriction and 0x08 for row specific restriction.0x01 for show the mouse pointer. Used registers are cx and dx. Interrupt used is int86() with interrupt number 0x33.

ALGORITHM

Step 1: Start.

Step 2: Check the availability of driver.

Step 3: Show the mouse pointer.

Step 4 : Set the ax register as 0x07 and read the column values to cx and dx registers.

Step 5 : Call the interrupt.

Step 6 : Set the ax register as 0x08 and read the row values to cx and dx registers.

Step 7: Call the interrupt.

Step 8: Show the mouse pointer.

Step 9: Exit through right button click on mouse.

Step 10: Stop.

PROGRAM

```
/*Restrict mouse movement*/
#include<stdio.h>
#include<conio.h>
#include<dos.h>
#include<process.h>
void main()
{
union REGS i,o;
clrscr();
int r1,r2,c1,c2,button;
clrscr();
i.x.ax=0;
int86(0x33,&i,&o);
if(o.x.ax==0)
{
printf("No mouse.");
exit(0);
}
i.x.ax=1;
int86(0x33,&i,&o);
i.x.ax=7;
printf("Enter the start and end column");
scanf("%d %d",&c1,&c2);
i.x.cx=c1;
i.x.dx=c2;
int86(0x33,&i,&o);
i.x.ax=8;
printf("Enter the start and end row");
scanf("%d %d",&r1,&r2);
i.x.cx=r1;
i.x.dx=r2;
int86(0x33,&i,&o);
```

COMPUTER HARDWARE AND NETWORKING LAB

```
do
{
i.x.ax=3;
int86(0x33,&i,&o);
button=o.x.bx&3;
}while(button!=3);
getch();
}
```

OUTPUT

Enter the start and end column 350 350

Enter the start and end row 350 350

RESULT

Program run successfully and required output obtained.

Perveen K

No: 31)

MEMORY SIZE DETERMINATION

10-10-2006

AIM

Write a program to print the memory size.

DESCRIPTION

Used registers for the memory size determination is ax. No specific Service number used here. Interrupt used is int86() with interrupt number 0x12. Memory size value stored in ax register.

ALGORITHM

Step 1: Start.

Step 2: Call the interrupt with interrupt number 0x12.

Step 3: Print the size of ax register.

Step 4: Stop.

COMPUTER HARDWARE AND NETWORKING LAB

PROGRAM

```
/*Memory size*/
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
union REGS i,o;
clrscr();
int86(0x12,&i,&o);
printf("Ram size=%d",o.x.ax);
getch();
}
```

OUTPUT

Ram size=640

RESULT

Program run successfully and required output obtained.

Pooja

No: 32)

CHECK THE PRINTER STATUS

10-10-2006

AIM

Write a program to check the printer status.

DESCRIPTION

For checking the printer status the service number used is 0x02. Used registers are ah, dx. Interrupt used is int86() and the interrupt number is 0x17H

ALGORITHM

Step 1: Start.

Step 2: Set ah with value 0x02.

Step 3: Initialize dx with 0x00.

Step 4: declare a variable mask with value 01.

Step 5: Call the interrupt.

Step 6: do the following steps till loop boundary reach 128

Step 6.1: Perform the AND operation between the content of ah and mask save result in temp.

Step 6.2: Double the mask value.

Step 6.3: Switch(temp)

Step 6.4: Case 1 print Time Out.

Step 6.5: Case 8 print ID Error

Step 6.6: Case 32 Print end of the paper

Step 6.7: Case 64 Print Ack: the signal.

Step 6.8: Case 128 Print printer ideal

Step 6.9: End of the switch case.

Step 7: Stop.

PROGRAM

```
/* Printer Status*/
#include<stdio.h>
#include<conio.h>
#include<dos.h>
#include<process.h>
void main()
{
clrscr();
union REGS i,o;
i.h.ah=0x02;
i.x.dx=0x00;
int mask=01,temp,j;
int86(0x17,&i,&o);
for(j=0;j<=128;j++)
{
temp=mask&o.h.ah;
mask=mask<<1;
switch(temp)
{
case 1:printf("Time out\n");
break;
case 8:printf("ID error\n");
break;
case 32: printf("Out of the paper\n");
break;
case 64:printf("Ack: signal send\n");
break;
case 128:printf("Printer ideal\n");
break;
default: break;
}
}
}
```

COMPUTER HARDWARE AND NETWORKING LAB

```
getch();  
}
```

OUTPUT

Printer ideal

RESULT

Program run successfully and required output obtained.

Pamodh

No: 35)

17-10-2006

BOOT SECTOR OF FLOPPY**AIM**

Write a program to show the details of boot sector of a floppy.

DESCRIPTION

The boot sector of a floppy disk contains boot information and the details of disk boot startup program. `absread()` function used to read the specific disk sector. The syntax of `absread()` is given below:

```
int absread(int drive,int nsect,long lsect,void *buffer);
```

`absread()` uses dos interrupt to read specific disk sector

drive: Drive no to read 0=A&1=B

nsect: No of sector to read.

lsect: Beginning Logical sector No:

buffer : Address value where the data is to be read.

ALGORITHM

Step 1: Start.

Step 2: Declare necessary variables.

Step 3: Check return value of `absread()`. If not a zero then

Step 4: print disk problem.

Step 5: print all the needed details.

Step 6: Stop.

PROGRAM

```
//boot sector of a floppy
#include<stdio.h>
#include<process.h>
#include<dos.h>
#include<conio.h>
void main()
{
int i,start,ch_out,sector;
struct details
{
char c;
unsigned int addr;
char name[8];
char no_of_bps;
char set_p_alloc;
int reverse_sect;
char no_fat;
int max_root;
int no_of_sect;
unsigned char media_disk;
int no_of_set_in_lfact;
int no_of_fps;
int no_of_sum;
int hiddensect;
}buff;
clrscr();
printf("\n Insert a floppy disk and press any key");
getch();
sector=0;
if(absread(0,1,sector,&buff)!=0)
{
```

COMPUTER HARDWARE AND NETWORKING LAB

```
perror("Disk Problem");
exit(1);
}
printf("\n Start of boot code::%xh",buff.addr);
printf("\n Version::%s,buff.name");
printf("\n Sector Size::%d",buff.no_of_bps);
printf("\n No. of Reserved Sector::%d",buff.reverse_sect);
printf("\n No. of Fats::%d",buff.no_fat);
printf("\n No. of Root::%d",buff.max_root);
printf("\n No. of Sector::%d",buff.no_of_sect);
printf("\n No. of Descriptor::%x",buff.media_disk);
printf("\n No. of sectors in one FAT::%d",buff.no_of_set_in_lfact);
printf("\n No. of sectors per Track::%d",buff.no_of_fps);
printf("\n No. of Surfaces::%d",buff.no_of_sum);
printf("\n No. of Hidden Sector::%d",buff.hiddensect);
getch();
exit(0);
return;
}
```

RESULT

Program run successfully and required output obtained.

Perinika

No: 36)

17-10-2006

STATUS OF FLOPPY

AIM

Write a program for checking status of floppy.

DESCRIPTION

BIOS disk service includes the function biosdisk() and _bios_disk(). Both use interrupt 13h to issue disk operations directly to the BIOS. The function used have the following syntax:

int absread(int,int,long,void *)

int abswrite(int,int,long,void *)

int biosdisk(int,int,int,int,int,int,void *)

BIOS uses interrupt 13h to issue disk operations directly to the BIOS

ALGORITHM

Step 1: Start.

Step 2: Declare necessary variables.

Step 3: Check return value of absread() if not a zero then.

Step 3.1: if errno=-32766 print Drive not ready

Step 3.2: else check biosdisk() return value.

Step 3.2.1 : if value=0x10 then print bad floppy

Step 4: else print Good Floppy.

Step 5: check abswrite() return value if not a zero then.

Step 5.1: if errno=768 then print Floppy write protected.

Step 5.2: else print Bad floppy.

Step 6: Stop.

COMPUTER HARDWARE AND NETWORKING LAB

PROGRAM

```
//Status of Floppy Disk
#include<stdio.h>
#include<conio.h>
#include<dos.h>
#include<process.h>
#include<errno.h>
#include<bios.h>
int main(void)
{
    int i,ch_out,sector,result;
    char buff[512];
    clrscr();
    printf("Insert a floppy and press a key\n");
    getch();
    sector=0;
    if(absread(0,1,sector,&buff)!=0)
    {
        f(errno==-32766)
        {
            printf("\n Drive not ready");
            getch();
            exit(0);
        }
        else
        {
            result=biosdisk(4,0,0,0,0,1,buff);
            if(result==0x10)
            {
                perror("\n bad floppy");
                exit(1);
            }
        }
    }
}
```

COMPUTER HARDWARE AND NETWORKING LAB

```
}  
else  
    printf("read OK in drive Reader good floppy");  
if(abswrite(0,1,sector,&buff)!=0)  
    if(errno==768)  
        printf("floppy write protected");  
    else  
        printf("bad floppy");  
getch();  
printf("\n");  
return 0;  
}
```

RESULT

Program run successfully and required output obtained.

Pooja