Synopsis

Aim: Graphical representation of Interrupt handling in a processor using stack

Objectives:

\* Interrupt handling with specified Interrupt handlers in a processor using stack.

\* Implementing the Interrupts handling graphically.

Introduction:

Interrupt and handler

An interrupt is a request of the processor to suspend its current program and transfer control to a new program called the Interrupt Service Routine (ISR). Special hardware mechanisms that are designed for maximum speed force the transfer. The ISR determines the cause of the interrupt, takes the appropriate action, and then returns control to the original process that was suspended.[1]

\* In 1954, the NBS DYSEAC was the first one to have an I/O interrupt.[2]

\* An interrupt is a change in a execution caused by an external event. [3]

Types of Interrupts [4]:

1. Software Interrupts

2. Hardware Interrupts

3. Exceptions

* Interrupt handling in a PC:

On a PC there are 24 separate interrupt lines that can be asserted. The appropriate interrupt handler is invoked based on the interrupt number[5].

Interrupt handling – OS ISSUES

* When an interrupt is serviced the processor must be able to execute without being interrupted. It must have the capability of temporarily disabling the interrupt atomically[6]
* If an interrupt occurs while an interrupt service is ongoing, it is simply deferred and considered a pending interrupt. It is serviced after the current request terminates[7].

Autodetecting the IRQ Number:[8]

How to get the number? Have the device tells you

MostmoderndevicesannouncetheIRQ#ituses Including PCI devices

The drivers get the IRQ# by reading a status byte from one of the device’s I/O ports or PCI configuration space.

Mechanism:

When a normal program is running suddenly a interrupt is generated, the value of the program counter is pushed in to the stack then ISR(Interrupt Service routine) will be handling the interrupt in case if there is another interrupt by hardware / software then again the current value of the program counter pushed in to the stack like this there will be a series of interrupts occurs and those stored in the stack.

The first interrupt of the stack will be popped and handled by the processor it continues till it executes the first process which we have pushed into the stack.

This will be shown in a graphical representation.

System & Software requirements:

* Software Requirements:
* Languages : c++
* IDE :Drop Box
* OS :windows 7/higher
* Hardware Requirements:
* Processor :Pentium i3/higher
* RAM : Minimum 1024MB
* Hard disk : Minimum 500GB

Design Specification

Algorithm to graphical implement Interrupt handling:

Step 1: start

Step 2:[Running a program like printing 1 to 1000]

For(\_i=0;i<10000;i++) do else go to Step 5

Step 3:interrupt()

Step 4:print i;

Step 5:stop

Algorithm for interrupt()

Step 1:Start

Step 2:Signal=[using functions like SIGABRT SIGALRM

SIGHUP csignal header files ]

Step 3:if(Signal!=NULL) else go to Step 6

Step 4:pushstack()

Step 5:callISR();

Step 6:stop

Algorithm for pushstack()

Step 1:start

Step 2:item=[getting program counter value]

Step 3:interruptStack[i++]

Step 4:programcounter=[putting values of present interrupt ISR]

Step 5:stop

Algorithm for call\_ISR()

Step 1: Start

Step 2:Switch(Signal)

{

case SIGABRT: Abnormal termination with interrupt;

case SIGALRM: ???..

case SIGHUP:?..

.

.

.

}

step3:stop

Main Program

i

STOP

Print i

Interrupt Function call

for i to 1000

START

Flow chart for Interrupt

START

Signal!=NULL

Call push stack function

Call IRS function

STOP

Flow chart for Pushstack

START

Interrupt stack[i++]=PC

Print interrupt

STOP

Flow chart for ISR

SIGNAL

START

case:SIGABRT

case:SIGALRM

.

.

.

.

.

.

.

default

STOP

Source code:

#include<iostream.h> //for cin, cout

#include<conio.h> //for getche()

#include<ctype.h> //for tolower() function

#include<stdlib.h> //for itoa()

#include<GRAPHICS.H> //to load graphics driver

#include<signal.h>

#include<fstream.h>

//#include"INTERRUPT.CPP"

#include<dos.h> //for delay() function

int col=60 , row = 417 ,n ,x=60 , y=417 ,size=10 ,arr[10]={0,0,0,0,0,0,0,0,0,0} , count = 0, top = 0; //Global Variables

char string[45];

char signo[10][10]={"SIGABRT","SIGFPE","SIGILL","SIGINT","SIGSEGV","SIGTERM"};

int cou=0,first=0;

char ch[2];

void push( );

void pop( );

void stack();

void loadgraph();

void signalHandler( int );

void rise1();

void popfu();

int per[10]={0,0,0,0,0,0,0,0,0,0};

void work1();

void info();

void manu()

{

line(639,0,639,639);

line(350,200,639,200);

line(0,0,0,650);

line(350,20,350,650);

moveto(250,10);

outtext("INTERRUPT HANDLING");

line(245,19,322,19);

line(328,19,398,19);

moveto(360,30);

outtext("SIGABRT:Abnormal termination [1]");

moveto(360,50);

outtext("SIGFPE:erroneous arithmetic[2]");

moveto(360,70);

outtext("SIGILL:illegal instruction[3]");

moveto(360,90);

outtext("SIGINT:interactive attention[4]");

moveto(360,110);

outtext("SIGTERM:termination request[5]");

moveto(360,130);

outtext("SIGSEGV:invalid access storage[6]");

moveto(360,150);

outtext("signal informantion[7]");

}

void main()

{

clrscr();

char ch;

loadgraph();

setbkcolor(0);

setcolor(2);

do

{

gotoxy(37,5);

cout<<"Menu";

gotoxy(30,6);

cout<<"Stack [s]";

gotoxy(30,8);

cout<<"Exit [x]";

gotoxy(25,9);

cout<<"\t\t\t\t";

gotoxy(30,10);

cout<<"Enter Option [ ]\b\b";

ch=getche();

ch = tolower(ch);

gotoxy(30,15);

if(ch=='s')

stack();

cleardevice();

}while(ch!='x');

cleardevice();

closegraph();

}

void loadgraph()

{

clrscr();

int driver = DETECT, mode, errorcode;

initgraph(&driver, &mode, "c:\\tc\\bgi");

errorcode = graphresult();

if (errorcode != grOk)

{

cout<<"Graphics error: \n"<<(grapherrormsg(errorcode));

cout<<"Press any key to halt:";

getch();

exit(1);

}

}

void stack()

{

cleardevice();

setcolor(14);

moveto(300,425);

outtext("Top");

itoa(top,string,10);

outtextxy(308,450,string);

setbkcolor(0);

setcolor(14);

outtextxy(55,26,"STACK");

line(30,40,30,440);

int r=40;

for(int i = 1 ;i<12 ;i++)

{

line(30,r,110,r);

r+=40;

}

line(110,40,110,440);

r = 55;

for( i = 10 ;i>=1 ;i--)

{

moveto(10,r);

itoa(i,string,10);

outtext(string) ;

r+=40;

}

line(165,440,210,440);

line(165,440,165,460);

line(210,440,210,460);

line(165,460,210,460);

setfillstyle(0, getmaxcolor());

do

{

moveto(400,170);

outtext("ENTER OPTION ");

if(kbhit())

{

ch[0]=getche();

}

else

ch[0]='z';

//convert entered option in lower character

ch[0] = tolower(ch[0]);

//Displays stacks Menu

manu();

rise1();

signal(SIGINT, signalHandler);

}while(ch[0]!='x');

}

void push()

{

int a=178 , b=425 ;

/\*---------------------------------------------------------------\*/

//displays the value that is to be pushed in stack

moveto(178,447);

itoa(arr[count],string,10);

outtext(string);

/\*--------------------------------------------------------------\*/

/\*after a delay remove the value , and display the

value in box that is to pushed next \*/

delay(10);

bar(170,446,208,455);

itoa(arr[count+1],string,10);

moveto(178,447);

outtext(string);

/\*--------------------------------------------------------------\*/

moveto(a,b);

/\*this loop displays and moves the value untill top row position

is not found \*/

while(gety()!=y)

{

setfillstyle(0, getmaxcolor());

circle(a+6,b+3,12);

itoa(arr[count],string,10);

outtext(string);

delay(10);

setfillstyle(0, getmaxcolor());

bar(170 ,gety()-20 ,200 ,gety()+14);

moveto(a,b);

b--;

}

setfillstyle(0, getmaxcolor());

circle(a+6,b+3,9);

outtext(string);

moveto(125,b);

/\*--------------------------------------------------------------\*/

//This loop displays the animated arrow toward top box os stack

for(int j =0 ;j<6 ; j++)

{

outtext("®");

delay(100);

setfillstyle(0, getmaxcolor());

bar(getx()-3 ,gety() ,getx() ,gety()+8);

delay(100);

}

/\*--------------------------------------------------------------\*/ delay(100);

//Remove last displayed value

setfillstyle(0, getmaxcolor());

bar(getx()-48 ,gety() ,getx() ,gety()+8);

/\*--------------------------------------------------------------\*/ delay(100); delay(200);

/\*while required coloumn of top box is not found , Display

and move value to required colimn \*/

while(getx()!=x+60)

{

setfillstyle(0, getmaxcolor());

circle(a+6,b+3,9);

itoa(arr[count],string,10);

outtext(string);

delay(10);

setfillstyle(0, getmaxcolor());

bar(getx()-20 ,gety()-10 ,getx()+70 ,gety()+20);

moveto(a,b);

a--;

}

/\*--------------------------------------------------------------\*/ delay(100);

//display the value in required box

moveto(60,gety());

outtext(string);

count++;

// to have cursor position in top box of stack

y = y-40;

}

void pop()

{

int a=120, b ;

//Update row , from where next value is to be poped

y = y + 40 ;

b = y ;

moveto(120,y);

/\*-------------------------------------------------------------\*/

/\* This loop displayes the arrows towards the stack's box

from where , value is to be poped \*/

for(int j=0 ;j<6 ; j++)

{

outtext("®");

delay(100);

setfillstyle(0, getmaxcolor());

bar(getx()-3 ,gety() ,getx() ,gety()+8);

delay(100);

}

/\*-------------------------------------------------------------\*/

// Remove last displayed arrow

delay(100);

setfillstyle(0, getmaxcolor());

bar(getx()-48 ,gety() ,getx() ,gety()+8);

delay(200);

moveto(60,gety());

setfillstyle(0, getmaxcolor());

bar(getx()-10 ,gety()-3 ,getx()+20 ,gety()+8);

/\*-------------------------------------------------------------\*/

/\*This loop will display the value untill poped value is not

positioned in values box column \*/

delay(500);

while(getx()!=178)

{

setfillstyle(0, getmaxcolor());

circle(a+6,b+3,12);

moveto(a,gety());

itoa(arr[count-1],string,10);

outtext(string);

delay(10);

setfillstyle(0, getmaxcolor());

bar(getx()-25 ,gety()-10 ,getx()+30 ,gety()+20);

a++;

}

/\*-------------------------------------------------------------\*/

/\*This loop will display the value untill poped value is not

positioned in values box Row \*/

delay(500);

while(gety()!=430)

{

setfillstyle(0, getmaxcolor());

circle(a+21,b+3,7);

moveto(178,b);

itoa(arr[count-1],string,10);

outtext(string);

delay(10);

setfillstyle(0, getmaxcolor());

bar(getx()-30 ,gety()-10 ,getx()+20 ,gety()+9);

b++;

}

/\*-------------------------------------------------------------\*/

// These statements displyes poped values in vlaue box

delay(10);

moveto(178,447);

bar(170,446,208,455);

count--;

moveto(178,447);

itoa(arr[count],string,10);

outtext(string);

size++;

/\*-------------------------------------------------------------\*/

}

void signalHandler( int signum )

{

}

void handle(int signo)

{

outtextxy(10,447,"pushing signal");

// cout << "Interrupt signal (" << signum << ") received.\n";

if(cou>1)

{

// start of outer if

// displays p

// outtext("pushing signal");

// make a delay

delay(500);

// draw a bar to remove p

bar(getx()-30,gety(),getx()+2,gety()+20);

/\*--------------------------------------------------------\*/

// if size of stack's remaining boxese less than one

// than stack is overflow

if(size<1)

{ //start of nested if

for(int i = 0 ;i<4 ; i++){

moveto(420,170);

outtext("Stack Overflow ");

delay(1000);

setfillstyle(0, getmaxcolor());

bar(420 ,165 ,630 ,180);

delay(500);}

}//end of nested if

/\*--------------------------------------------------------\*/

// if there are stack's boxes remains (if size>1)

else

{

//start of nested else part

// decrease stack remaining boxes

size--;

// increase top position , where last item is Inserted

top++;

// call to push function

push();

// bar removes last displayes value

bar(290,440,350,479);

// convert integer value to assci value

/\* becuse outtext/outtextxy function require string

as third argument \*/

itoa(top,string,10);

outtextxy(308,447,string);

}//End of Nested else Part

}

cou++;

}

void rise1()

{

switch(ch[0])

{

case '1'://raise(SIGABRT);break;

case '2'://raise(SIGFPE);break;

case '3'://raise(SIGILL) ;break;

case '4'://raise(SIGINT);break;

case '5'://raise(SIGSEGV);break;

case '6'://raise(SIGTERM);break;

handle(2);per[top]=0;break ;

case '7':info();break;

default:work1();//per[top]++;//bar(360,170,500,180);

//moveto(420,170);

// outtext("Error");

break;

}

}

void popfu()

{ bar(10,440,40,550);

outtextxy(10,447,"poping signal");

// start of outer if's elseif part

// Displays o , and remove it after a delay

// outtext("o");

delay(500);

bar(getx()-30,gety(),getx()+10,gety()+20);

/\*--------------------------------------------------------\*/

/\* if all boxes of stack are empty than display message

of overflow \*/

if(size==10)

{

for(int i = 0 ;i<4 ; i++){

moveto(410,170);

outtext("Stack, Underflow ");

delay(1000);

setfillstyle(0, getmaxcolor());

bar(330 ,165 ,630 ,180); // (left, top) ³ upper left corner

delay(500); } // (right, bottom) ³ lower right corner\*/

}

/\*--------------------------------------------------------\*/

/\*if there are some boxes of stack filled than call

pop function \*/

else

{

// update top's value

top--;

pop();

//display value of updated top

bar(290,440,350,479);

itoa(top,string,10);

outtextxy(308,447,string);}

}

void work1()

{ // if(top!=0)

// {

moveto(360,210);

outtext("Handling is going on");

setfillstyle(INTERLEAVE\_FILL,BLACK);

bar(360,220,620,233);

setfillstyle(INTERLEAVE\_FILL,RED);

bar(360,220,360+(per[top]\*2.6),230);

setfillstyle(INTERLEAVE\_FILL,BLACK);

bar(480,221,500,226);

itoa(per[top],string,10);

outtextxy(480,222,string);

char al[2],line[80];

int hh1=0,d2=0;

ifstream fou ("b.txt");

if(!fou.good())

{

moveto(480,242);

outtext("can't open");

}

while(fou.eof())

{

if(hh1%80)

{hh1=0;

d2=d2+20;

}

fou.get(al[0]);

al[1]='\0';

moveto(480+hh1,230+d2);

outtext(al);

hh1+2;

if(d2==300)

{d2=0;

setfillstyle(INTERLEAVE\_FILL,BLACK);

bar(480,221,500,226);

}

}

fou.close();

if(per[top]==100)

{popfu();

per[top]=0;

}

per[top]++;

arr[top]=per[top];

for(long int ee=0;ee<1000000;ee++);

// }

}

void info()

{ int rr;

char ch[2];

do{

setfillstyle(INTERLEAVE\_FILL,BLACK);

bar3d(200,150,520,450,9,1);

// bar(181,51,449,249);

gotoxy(30,15);

cout<<"Enter the Top no for infornamtion";

ch[0]=getche();

int we=int(ch[0]);

we=we-48;

gotoxy(30,20);

cout<<"top no"<<we;

gotoxy(30,22);

cout<<"Name of Signal: "<<signo[we];

int \*ee=&per[we];

gotoxy(30,24);

cout<<"addres: "<<ee;

gotoxy(30,26);

// cout<<per[ch]<<"%";

for(long int ff=0;ff<10000000;ff++);

}while(ch[0]!='x');

stack();

}

Testing



Figure 1: Graphics of Interrupt occurance.



Figure 2:Completion of Interrupt Handling



Figure 3:Information of about Interrupt in the stack.

References:

[1]. By Frederico Jer¢nimo, Copyright 1999-2000 , [www.delorie.com](http://www.delorie.com/)/djgpp/ doc/ug/I nterrupts/inthandlers1.html

[2]. Interrupt handling a ppt by Hebah Kohar,Ratnaker Reddy Bodhireddy and sheril Dhabriya.

[3]. People.cs.clemson.edu/`mark/interrupts.html.

[4]. S. Dandamudi, ?Fundamentals of Computer Organization and Design,? Springer, 2003

[5]. Interrupt handling a ppt by Hebah Kohar,Ratnaker Reddy Bodhireddy and sheril Dhabriya.

[6]. Interrupt handling a ppt by Hebah Kohar,Ratnaker Reddy Bodhireddy and sheril Dhabriya.

[7]. Interrupt handling a ppt by Hebah Kohar,Ratnaker Reddy Bodhireddy and sheril Dhabriya.

[8]. The information on the slides are from  
“*Linux Device Drivers*, Third Edition, by Jonathan Corbet, Alessandro Rubini, and Greg Kroah- Hartman. Copyright 2005 O’Reilly Media, Inc., 0-596-00590-3.”