Thread: High Speed IRQ Timer/Clock in C

12-27-2007 #1

andrewwan1980
Registered User
Join Date: Jun 2007
Posts: 9

High Speed IRQ Timer/Clock in C

I found this excellent High Speed Timer (in Pascal). I compiled it (using Turbo Pascal 7 and it runs fine):

http://www.sorucevap.com/bilisimtekn...ers.asp?207995
and same High Speed Timer here too:
http://groups.google.com/group/comp....7ff3cf587648ef

I converted it to C (using p2c), compiled it using Borland C++ 4.5 and it runs. But it crashes when it gets to setvect(...) in TimerOn.

Does anyone know how IRQ programming works in C? If you know IRQ timer/clock please contact me. I need urgent help in understanding why it's not working.

12-27-2007 #2

andrewwan1980
Registered User
Join Date: Jun 2007
Posts: 9

Here is the converted code to C. It's suppose to cycle through NewInt08, NewInt1C repeatedly. Yes, behaviour of the program is erratic, sometimes it prints out about 20 NewInt08, NewInt1C before it crashes, other times it crashes straight away. I just want someone who has done IRQ programming before to see if they can spot any mistakes am making.

Code:

```
1  #define MaxRate         1193180L
2
3
4  void interrupt(  far *OldInt08)();//Static  PROCE
```

```
Static unsigned short IntCount08, Trigger;
Static boolean TimerAlreadySet;
Static unsigned short Frequency;

void GetIntVec(int a, void interrupt(__far *b)())
{
    b = getvect(a);
}

void SetIntVec(int a, void interrupt(__far *b)())
{
    setvect(a, b);
}


Static Void IrqOn()
{
  asm{sti;}//asm(" inline $FB");
}


Static Void IrqOff()
{
  asm{cli;}//asm(" inline $FA");
}

/*$F+*/
void interrupt NewInt1C()//Static Void NewInt1C()
{printf("\nNewInt1C()");
  ClockTicks++;
}

/*$F-*/

/*$F+*/
void interrupt NewInt08()//Static Void NewInt08()
{printf("\nNewInt08()");
  IrqOff();
  asm{int 1Ch;}//asm(" inline $CD");
  //asm(" inline $1C");   /*Generate INT 1Ch inst
  if (IntCount08 == Trigger) {
     IntCount08 = 0;
     asm{pushf;}//asm(" inline $9C");
     OldInt08();/*if (OldInt08.link != NULL)
        (*(Void(*) PP((Anyptr _link)))OldInt08.pr
     else
        (*(Void(*) PV())OldInt08.proc)();*/
  } else
    IntCount08++;
  outportb( 0x20, 0x20 );//PORT(0x20) = 0x20;   /
  IrqOn();
}


/*$F-*/

Void TimerOn(Freq)
long Freq;
{
  LONGINT Temp = MaxRate;
  unsigned short Count;
  _PROCEDURE TEMP1;
printf("\nTimerOn()");
  if (TimerAlreadySet)
    return;
  ClockTicks = 0;
  IntCount08 = 0;
  Frequency = Freq;
  Trigger = (long)(Freq / 18.2);
  Temp = (long)((double)Temp / Freq);
  Count = Temp;
```

```
  TEMP1.proc = (Anyptr)NewInt08;
  TEMP1.link = (Anyptr)NULL;
  SetIntVec(0x8, NewInt08);//SetIntVec(0x8, TEMP1);
  GetIntVec(0x1c, OldInt1C);
  TEMP1.proc = (Anyptr)NewInt1C;
  TEMP1.link = (Anyptr)NULL;
  SetIntVec(0x1c, NewInt1C);//SetIntVec(0x1c, TEMP1
  outportb( 0x43, 0xb6);
  outportb( 0x40, Count & 255);
  outportb( 0x40, Count >> 8);
  TimerAlreadySet = true;
}


Void TimerOff()
{printf("\nTimerOff()");
  if (!TimerAlreadySet)
    return;
  outportb( 0x43, 0xb6);
  outportb( 0x40, 0xff);
  outportb( 0x40, 0xff);
  SetIntVec(0x8, OldInt08);
  SetIntVec(0x1c, OldInt1C);
  TimerAlreadySet = false;
}


Void ResetTimer()
{
  ClockTicks = 0;
}


double TimeElapsed()
{
  return ((double)ClockTicks / Frequency);
}


void _Timer_init()
{
  static int _was_initialized = 0;
  if (_was_initialized++)
    return;
  TimerAlreadySet = false;
}

void main()
{
  int i;
  _Timer_init();

  TimerOn(546);
  for(i=0; i<100000; i++) {
     if (i&#37;10000==0)
        printf("\n... %d", i);
  }
  TimerOff();

}
```

Last edited by andrewwan1980; 12-28-2007 at 04:25 AM. ***Reason:*** .

andrewwan1980

Registered User

Join Date: Jun 2007
Posts: 9

The thing is, it worked perfectly from a Pascal compiled source. But from C, I get this error:

16 bit MS-DOS Subsystem (DIALOG)
C:\TIMER\timer.exe
The NTVDM CPU has encountered an illegal instruction.
CS:0008 IP:08f1 OP:0f 00 74 01 c3 Choose 'Close' to terminate the application.

Surely if one is allowed to do IRQ programming in Pascal and XP allows it executed 100%, then it should also work under C too.

Am just looking for oldskool experienced C IRQ programmers who is familiar with this timer code.

12-27-2007 #4

Salem

and the hat of int overfl

Join Date: Aug 2001
Location: The edge of the known universe
Posts: 36,491

Perhaps the NT/XP emulation of 16-bit DOS isn't as perfect as you would like it to be.

How about stating the actual problem, rather than this round-about method of converting 16-bit Pascal to 16-bit C, running on a 32-bit OS.

If you dance barefoot on the broken glass of undefined behaviour, you've got to expect the occasional cut.
If at first you don't succeed, try writing your phone number on the exam paper.

12-27-2007 #5

Adak

Registered User

Join Date: Sep 2006
Posts: 8,868

You have shown nothing about C, Andrew. What you've shown is that your Pascal to C converter, just didn't work exactly right.

If you want to REALLY know the reason, you'll need to pull out your ASM output from the compiler, for both program,s, and compare; see what's different.

My opinion is that software timers on Window's systems, are pretty good if you give them 1/2 second, or more. With less than 1/2 second, they become more erratic, and their margin of error grows exponentially as you decrease the time being measured.

Hardware dedicated to the purpose, and to that timing purpose alone, is the only way to get accurate measurements of very small amounts of time. Windows is not even close to being a real-time OS.

01-16-2008 #6

andrewwan1980

Registered User

Join Date: Jun 2007
Posts: 9

I found the closest Turbo C DOS timer source code written by David Oshinsky at:

http://www.bookcase.com/library/soft...s.turbo-c.html (TIMERTST)

Code:

```
To get my own timer (original Pascal port) I had to c
```

What am wondering is:

1. In Pascal version it uses inline($CD / $1C); & inline($9C); before the OldInt08() call. I though I could call the equivalent asm{ int 1Ch;} & asm{pushf;} but I guess this was wrong since it crashed my program. Is there an inline assembly call in C?

2. David's initializing the timer code uses

Code:

```
/* Set up 8259 PIC chip to allow INT0 interrupt. */
outportb(0x21, inportb(0x21) & 0xfe);

/* issue command to 8253:  counter 0, binary counter
/* (mode 2), load least significant byte of counter
/* most significant byte                          */
outportb(0x43, 0x34);

/* Timer is set for 0x4cd * 813.8 ns = 1 ms (LSB fol
outportb(0x40, 0xcd); /* least significant byte of t
outportb(0x40, 0x04); /* most significant byte of tir
```

But mine uses

Code:

```
outportb( 0x43, 0xb6);
outportb( 0x40, Count & 255);
outportb( 0x40, Count >> 8);
```

What is the difference? Even though both works.

3. Same goes for the clean up code:
His is:

Code:

```
/* restore 8253 to original state set during PC boot
/* NOTE:  this program leaves 8259 mask register with
/* least significant bit clear (i.e., INT0 enabled).
outportb(0x43, 0x34);
outportb(0x40, 0);
outportb(0x40, 0);
```

Mine is:

Code:

```
outportb( 0x43, 0xb6);
```

```
3   outportb( 0x40, 0xff);
```

Again both works but I don't understand why different?

01-16-2008 #7

matsp

Kernel hacker

Join Date: Jul 2007
Location: Farncombe, Surrey, England
Posts: 15,677

1. With my "recognize instructions without a book", your int 0x1c; pushf sounds about right. Perhaps something else is wrong?

2. Not sure what mode 0xb6 means in the timer - haven't got my book around either.

3. Well, one counts 65535 and the other counts 65536 clock-ticks. Not much of a difference really - although I always thought that the DOS timer was actually initialized to a slightly different value than that.

--
Mats

Compilers can produce warnings - make the compiler programmers happy: Use them!
Please don't PM me for help - and no, I don't do help over instant messengers.

01-16-2008 #8

matsp

Kernel hacker

Join Date: Jul 2007
Location: Farncombe, Surrey, England
Posts: 15,677

Also note that the 8253/54 is not a "high speed" timer - there are much better timers available in modern hardware.

--
Mats

Compilers can produce warnings - make the compiler programmers happy: Use them!
Please don't PM me for help - and no, I don't do help over instant messengers.

Quick Navigation C Programming Top

« Previous Thread | Next Thread »

Popular pages

- Exactly how to get started with C++ (or C) today
- C Tutorial
- C++ Tutorial
- 5 ways you can learn to program faster
- The 5 Most Common Problems New Programmers Face
- How to set up a compiler
- 8 Common programming Mistakes
- What is C++11?

- Creating a game, from start to finish

Recent additions

- How to create a shared library on Linux with GCC - December 30, 2011
- Enum classes and nullptr in C++11 - November 27, 2011
- Learn about The Hash Table - November 20, 2011
- Rvalue References and Move Semantics in C++11 - November 13, 2011
- C and C++ for Java Programmers - November 5, 2011
- A Gentle Introduction to C++ IO Streams - October 10, 2011

Similar Threads

Flight Simulator Wind Speed!! By Dilmerv in forum C++ Programming	Replies: 6 Last Post: 03-20-2006, 12:40 AM
I've got high speed internet!!!!! By PJYelton in forum A Brief History of Cprogramming.com	Replies: 20 Last Post: 03-27-2005, 07:57 AM
So long AOL, Hellooooo High SPEED By RoD in forum A Brief History of Cprogramming.com	Replies: 31 Last Post: 03-07-2004, 10:23 PM
high speed? Wow By Fountain in forum A Brief History of Cprogramming.com	Replies: 10 Last Post: 07-24-2002, 02:51 AM
Rogers High Speed Internet Router Setup By Unregistered in forum A Brief History of Cprogramming.com	Replies: 0 Last Post: 06-25-2002, 08:39 PM