Subject: SetTimeCallback failure on arch=armv5l POSIX_PLATFORM
Posted by jlfranks on Tue, 24 Jun 2008 15:10:15 GMT
View Forum Message <> Reply to Message

We have discovered a flaw with SetTimeCallback on an armv5l architecture with the Linux 2007.1 distribution compiled w/ GNU 4.2 eabi.

We are not sure if it is a Upp problem or not because it works fine on i686 Linux.

The Upp Clock program is our simplified test example.

We set the target system clock raw ticks to (decimal) 1213999990 and watch it count up to 1213999999 and then roll to 1214000000, the Clock program timer callback stops being called.

We've also tested 1214999990 and see the same behavior when it rolls over from 1214999999 to 1215000000, the clock appears to stop, and the timer callback is not called.

We ran 'top' and it shows that there is basically no CPU usage by the Clock program -- so, it is not going cpu-bound.

If we restart the Clock program just past the roll-over, everything works great (until the next roll-over).

We are perplexed in that we don't really know where to look for the problem. Is there a mechanism in Upp that could cause this?

Where else should we look for the problem. Is X involved somehow?

Any suggestion on how to proceed would be greatly appreciated.

NOTE: we have used Upp extensivly on the arv5l architecture and have just discovered the problem. Everything else in Upp works as it should.

--jlf

Subject: Resolved: SetTimeCallback failure on arch=armv5l POSIX_PLATFORM
Posted by jlfranks on Wed, 25 Jun 2008 15:44:20 GMT
View Forum Message <> Reply to Message

We were able to identify the problem to line 88 of CtrlTime.cpp
and applied a correction as follows:

```
void Ctrl::TimerProc(dword time)
{
 sTimerLock.Enter();
 TimeEvent *list = tevents();
```

```
 if(sTClick > time)
  for(TimeEvent *e = list->GetNext(); e != list; e = e->GetNext())
//    if(e->time > 0x80000000)
            if(e->time >= 1000000000)  // --jlf fixes freeze on 06/20/2008 22:13:20
     e->time = 0;
 sTClick = time;
 sTimerLock.Leave();
 Ctrl::CheckMouseCtrl();
 Ctrl::SyncCaret();
 sTimerLock.Enter();

 while(list->GetNext() != list && list->GetNext()->time < time) {
  TimeEvent *e = list->GetNext();
  e->Unlink();
  if(e->delay < 0)
   sTimeCallback(time - e->delay, e->delay, e->cb, e->id);
  sTimerLock.Leave();
  e->cb();
  sTimerLock.Enter();
  delete e;
 }
 sTimerLock.Leave();
}
```

---

Subject: Re: Resolved: SetTimeCallback failure on arch=armv5l POSIX_PLATFORM
Posted by mirek on Wed, 25 Jun 2008 21:42:20 GMT
View Forum Message <> Reply to Message

I do not think this is correct fix. This is supposed to with situation when time "wraps around" the dword.

The idea here is that most recent entries are in the second half of dword range when it wraps. Anyway, at the moment time already passed them. So they are reset to 0.

Well, anyway, I guess I just do not see the trouble. What do you achieve changing 0x80000000 to 1000000000 ?

Mirek

---

Subject: Re: SetTimeCallback failure on arch=armv5l POSIX_PLATFORM
Posted by mr_ped on Thu, 26 Jun 2008 07:16:21 GMT
View Forum Message <> Reply to Message

Mirek: I would expect ">= 0x80000000", not sharp ">", if it is supposed to handle wrap of dword.

Edit: wait a second, too early in morning for me . This does not make sense too... ">" vs ">=" would make sense only if there were also some signed/unsigned problem elsewhere.

1000000000 = 0x3B9ACA00
so it means it thinks about "wrap" sooner, but I don't get the rest of the code, so I have no overall idea.

---

Subject: Re: SetTimeCallback failure on arch=armv5l POSIX_PLATFORM
Posted by mirek on Thu, 26 Jun 2008 08:41:49 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Thu, 26 June 2008 03:16Mirek: I would expect ">= 0x80000000", not sharp ">", if it is supposed to handle wrap of dword.

Edit: wait a second, too early in morning for me . This does not make sense too... ">" vs ">=" would make sense only if there were also some signed/unsigned problem elsewhere.

1000000000 = 0x3B9ACA00
so it means it thinks about "wrap" sooner, but I don't get the rest of the code, so I have no overall idea.

Yes, I was thinking about signed/unsinged too... But IMO it should be OK.

Mirek

---

Subject: Re: SetTimeCallback failure on arch=armv5l POSIX_PLATFORM
Posted by jlfranks on Thu, 26 Jun 2008 14:25:09 GMT
View Forum Message <> Reply to Message

We put cout statements in the code during debug and found
that overflow did not occur because of the following code
in Util.cpp

The reset to a correct value must occur earlier than the
range of the int because of the modulo math in GetTickCount().

```
#ifdef PLATFORM_POSIX
int GetTickCount() {
 struct timeval tv[1];
 struct timezone tz[1];
 memset(tz, 0, sizeof(tz));
```

```
 gettimeofday(tv, tz);
 return tv->tv_sec % 1000000 * 1000 + tv->tv_usec / 1000;
}
#endif
```

--jlf

---

Subject: Re: SetTimeCallback failure on arch=armv5l POSIX_PLATFORM
Posted by mirek on Thu, 26 Jun 2008 16:16:13 GMT
View Forum Message <> Reply to Message

jlfranks wrote on Thu, 26 June 2008 10:25
We put cout statements in the code during debug and found
that overflow did not occur because of the following code
in Util.cpp

The reset to a correct value must occur earlier than the
range of the int because of the modulo math in GetTickCount().

```
#ifdef PLATFORM_POSIX
int GetTickCount() {
 struct timeval tv[1];
 struct timezone tz[1];
 memset(tz, 0, sizeof(tz));
 gettimeofday(tv, tz);
 return tv->tv_sec % 1000000 * 1000 + tv->tv_usec / 1000;
}
#endif
```

--jlf

Excellent, thanks! That is it...

Anyway, I guess it is rather worth fixing this function and make it return dword. I believe this should do the trick:

```
dword GetTickCount() {
 struct timeval tv[1];
 struct timezone tz[1];
 memset(tz, 0, sizeof(tz));
 gettimeofday(tv, tz);
 return (dword)tv->tv_sec * 1000 + tv->tv_usec / 1000;
}
```

Can you confirm?

Mirek

---