
Subject: Re: Alarm clock -- ring!
Posted by [steinhh](#) on Fri, 01 Nov 1996 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <55bct7\$e4s\$1@hammer.msfc.nasa.gov>, mallozzi@ssl.msfc.nasa.gov writes:

|>
|> Hi all,
|>
|> I've written a fairly large widget app, developed on a DEC Alpha. Recently,
|> I started to run it on an SGI (IRIX 5.3, IDL 4.0.1). More and more
|> frequently, I am getting the message "Alarm clock" and am kicked totally
|> out of IDL back to the unix prompt (no other messages). We have a
|> 5-user license for the machine. Does anyone know the meaning of this?
|> Thanks,
|>
|> -bob

I seem to remember that this happens when some timer is counted down to zero -- if your program doesn't use CALL_EXTERNAL to set/modify timers through the un*x system call setitimer ("man setitimer" for more info) this could be a problem with IDL not catching its timer signals in the correct way.

I developed some routines to modify/read the virtual timer for profiling purposes a long time ago, I'll include those below (with *no* modifications, use at own risk, it's a long time since I've used those..). Perhaps you could modify them to turn off the timer(s) that are causing the problem. Turning off timers that IDL uses for internal purposes would have unpredictable effects, though...:-)

On the other hand, check whatever constraints your shell is imposing on it's subprocesses -- use the command "limit" to see if e.g., cputime is limited to a finite amount -- maybe this would cause some SIGALRM event to happen..

If you don't get anywhere, I'd say this is a task for the support people at RSI.

Another option is to make a CALL_EXTERNAL routine using the system call "signal" to modify how SIGALRM and its friends are handled -- "man signal" or "man sigvec" or.... for more info.

Regards,

Stein Vidar H. Haugan

initime.pro (initialize virtual timer)

```

pro initime
  if !version.arch eq 'alpha' then $
    dummy=call_external('/mn/leda/u1/steinhh/IDL/C/timer.so','in itutimer')
  end

```

timer.pro (to read virtual timer -- use as e.g., systime(1) for timing purposes).

```

function timer
  if !version.arch eq 'alpha' then $
    return,call_external('/mn/leda/u1/steinhh/IDL/C/timer.so','g etutimer',$
      /D_VALUE) $
  else return,systime(1)
end

```

timer.c (contains routines called by initime/timer)

```

/* Call with:
   STATUS = CALL_EXTERNAL('..','initutimer')
*/

```

```

#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>

```

```

int initutimer(int argc,char * argv[])
{
  struct itimerval value;

  value.it_interval.tv_sec = 0L;
  value.it_interval.tv_usec= 0L;
  value.it_value.tv_sec = 32000L;
  value.it_value.tv_usec= 0L;
  setitimer(ITIMER_VIRTUAL,&value,&value);
  return 0;
}

```

```

/* Call with:
   time = call_external('..','getutimer',/D_VALUE)
*/

```

```
double getutimer(int argc, char *argv[])
{
    struct itimerval value;
    double time;

    getitimer(ITIMER_VIRTUAL,&value);

    time = (32000.0 - (double) (value.it_value.tv_sec)
           - (double) (value.it_value.tv_usec * 0.000001));
    return time;
}
```
