
Subject: Re: running IDL as a 'sleeping' process
Posted by [Bringfried Stecklum](#) on Tue, 03 Jan 2012 20:59:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

If the code shall run at regular times I recommend to use a shell script as a wrapper in combination with cron for the scheduling. At least for Linux this is fairly easy to do. There is no need to keep the IDL process forever in memory.

Regards, Bringfried

desertdryad wrote:

> Hi folks -
>
> This is a bit of an ignorant newbie question. I am developing a
> fairly complex bit of code, that in the long run will ideally entail
> daily data retrieval from the web, parsing that data and making
> several calculations, spitting out results, and then going to 'sleep'
> until the following day, when it all repeats. I'm wondering if this
> is even something IDL can *do*. I'm sure there are better platforms
> to develop such a sort of program, but I'm hoping not to reinvent the
> wheel on the development I've done so far (I've built several modules
> already in IDL for this project). I'm currently worknig under
> Windows, but it's highly likely this project will be migrating to a
> Linux OS.
>
> Is this sort of daily repeating process in the realm of possibility
> for IDL to do? Should I be thinking of some sort of 'wrapper' program
> that calls IDL to do this? I know just enough C/C++ to get in
> trouble :) but not much of anything like Java and the like. Comments
> and suggestions welcome. Thanks!

Subject: Re: running IDL as a 'sleeping' process
Posted by [natha](#) on Tue, 03 Jan 2012 21:04:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Under Linux you can use crontab to call a script that runs your code.
Your script could be something like:

```
#!/bin/bash  
cd /idl_routine_path  
idl -e your_routine
```

Under Windows you could use scheduled tasks to do something similar
but you have to create a SAV file.
Your 3th option is run the code in background. Using the command WAIT
you can do what you want. I combine this technique with the first one
to run some of my "invisible" IDL scripts to download and process data

every X time.

nata

Subject: Re: running IDL as a 'sleeping' process

Posted by [Vincent Sarago](#) on Tue, 03 Jan 2012 22:20:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

As bstecklu said, the best way to do a scheduling task is to combine a cron (unix) + a script calling the IDL procedure

here is an example of script :

```
#!/bin/sh
```

```
#set -x
```

```
#Working Directory
```

```
DIR=~/
```

```
# IDL directory
```

```
IDL="/opt/idl8/idl/bin/idl80"
```

```
LOCK=$DIR/.tasklock
```

```
# Function to delete the lock file
```

```
clean() {
```

```
    rm -f ${lock}
```

```
}
```

```
# cette fonction verifie si les 5 fichiers sont arrivé et lance IDL
```

```
dowork () {
```

```
FILE=$1
```

```
#Start IDL and run program
```

```
$IDL -32 -quiet -queue 2>> $DIR/log.txt <<here
```

```
restore, '../..../directory/project.sav'
```

```
spawn, 'echo $file', input
```

```
procedure, input
```

```
exit
```

```
here
```

```
}
```

```
#when end of script, call clean()
```

```
trap clean 0
```

```
# if lock found ...
if [ -f ${lock} ]; then
# ... if process is still running ...
  if [ -d /proc/`cat ${lock}` ]; then
    exit 0
  fi
fi

# writing process number in the lock file
echo -n $$ > "$lock"

FIND=$(find $DIR -maxdepth 1 -name 'file.tmp')

echo "$FIND" | while read file; do
  [ -z "$file" ] && continue
  dowork $input
done

# delete lock file
clean
```

vincent

Subject: Re: running IDL as a 'sleeping' process
Posted by [penteado](#) on Wed, 04 Jan 2012 15:21:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jan 3, 5:48 pm, desertdryad <dry...@gmail.com> wrote:
> Is this sort of daily repeating process in the realm of possibility
> for IDL to do? Should I be thinking of some sort of 'wrapper' program
> that calls IDL to do this? I know just enough C/C++ to get in
> trouble :) but not much of anything like Java and the like. Comments
> and suggestions welcome. Thanks!

One simple and IDL-only (thus platform-independent) way to do it is
use IDL's wait procedure. The program can do its thing, then call
sysyme() to find out how long to sleep, then call wait to sleep for
the desired time.

Subject: Re: running IDL as a 'sleeping' process
Posted by [Russell\[1\]](#) on Wed, 04 Jan 2012 21:14:49 GMT

On Jan 4, 10:21 am, Paulo Penteado <pp.pente...@gmail.com> wrote:

> On Jan 3, 5:48 pm, desertdryad <dry...@gmail.com> wrote:

>

>> Is this sort of daily repeating process in the realm of possibility
>> for IDL to do? Should I be thinking of some sort of 'wrapper' program
>> that calls IDL to do this? I know just enough C/C++ to get in
>> trouble :) but not much of anything like Java and the like. Comments
>> and suggestions welcome. Thanks!

>

> One simple and IDL-only (thus platform-independent) way to do it is
> use IDL's wait procedure. The program can do its thing, then call
> systime() to find out how long to sleep, then call wait to sleep for
> the desired time.

This is true, but this will mean you have an IDL prompt sitting there with no access to it. It will essentially be "stuck" on the wait statement for a very long time, and you may not want it to start at some preset time. Instead, a (perhaps more elegant) way along these lines, would be to use the widget_timer functionality. What amounts to the wait command with a long wait time derived from systime() would then be enveloped in an event handler that would trigger at some defined time. That could be simple (such as every 24 hours trigger), or you could test the existence of some auxiliary file (say every 30s or so) and when the file exists, then trigger the main code. This is probably a lot more work than you signed up for, but really not that bad once you get the hang of it. The virtue of this is that you get the IDL prompt back, and this thing essentially runs "in the background".

Russell
