We have some service which works long time (several months or even more). It seems nice to have some watchdog application which would support the main service and help to work it smoothly...

Let me introduce 'Watch Dog Service' (WDS). My current version of WDS implementation allows to control activity of one system process. For current example I wrote 'somebot.py' application which just prints current time with interval 2 seconds, it was written just to show operation of WDS. Suppose our 'somebot.py' application is a service which should work 24/7/12 and WDS will look after 'somebot' starts and restarts it if something happen with it.

WDS was implemented in 'watchdog.py' and has some local lib to serve some functionality.
Basically the lib consist from 2 parts: 1st is 'Scheduler' allows to call some functions periodically, 2nd is 'AppController' allows us to describe some particular application, or service which should be under control of our WDS. This 2 parts make a core of our WDS service.

In order to check work of WDS your lab should have Ubuntu with Python 3.3.1 not less than 3.2.*
I know you would like me to implement the task with Python 2.x and if change 'subprocess' part to classic os.fork() os.setsid() and os.exec*() it can operates under Python 2.x but I've never worked with 'subprocess' lib before and couldn't avoid temptation to meet it up with a project. I do hope it is not so critical just to check it for your lab.

How to check WDS:
1. Unpack archive to your test folder.
2. You should have appropriate version of Python 3x , ps, egrep (GNU grep 2.14)
3. decide how to start python scripts: 1st is `'python3 watchdog.py`' or 2nd `'./watchdog.py`'
for last way use `chmod +x watchdog.py`
Note: you can start somebot.py in a same way and/or separately, or wait for WDS to start it itself

Usage of 'watchdog.py':

```
$ ./watchdog.py -h
usage: watchdog.py [-h] [-c] [-f file=Namefile]

optional arguments:
  -h, --help           show this help message and exit
  -c, --cw             use this option if you have CW (color wrapper)
  -f file=Namefile, --cfg file=Namefile
                       use this option if you want to use some cfgFile
```

My system has application Color Wrapper (CW) which allow to color output of standard Linux applications in terminals if you have same application just use option '-c'. CW adds its own data to standard output of ps, egrep application and as a result have to parse output in a different way.
By default WDS control (starts, restarts) 'somebot.py' but if you would like you can check with another application or service. CfgFile may have only 2 strings begin from keys: 'prgName, flagCW' prgName is a name of program-service or application for control, flagCW can be yes/no if you have CW or no respectively. Key and value can be separated with space or tabs. CfgFile can't keep commentary strings. NOTE: I had no much time and checked WDS only with somebot.py. Default use of WDS is preferable

4. start WDS with command : `$ ./watchdog.py`
5. use: `ps ax |grep python`

```
32577 pts/7    S+     0:00 python3 ./watchdog.py
32582 ?        Ss     0:00 python3 ./somebot.py
```

6. kill somebot.py with command `kill -9 itsID`
7. use: `ps ax |grep python` then you see new pidID for somebot.py

8. to stop use of watchdog.py and somebot.py kill both process in the same way.

WDS will check every 5 seconds a status of 'somebot' if there is no such process it will be started, if somebot has (T,Z,X) status it will be restarted with WDS service. As an additional functionality there can be added analyzing of all parameters from /proc/**id and rules of behavior in WDS. Improvement process is endless, all we need is just to know what we want to …..

With the best regards,
Konstantin