PARANAL ERROR DATABASE
ESOpy2.0 - R. THOMAS

In Paranal we often have errors on the telescopes and instruments. The error messages are in general obscure for sciops department and the team staff might take some time to retrieve the good PPRS or the troubleshooting operation that allows to go back to the sky. The idea here is to create an error database that would link error message and troubleshooting and would increase the efficiency of the team crew.The principle would be to be able to fetch the error message from the log monitor into a program that display information about that error

-# of occurences,
-last occurrence,
-associated PPRSs
-Troubleshooting

If the error has never happened before, a new entry in the database would be created. The ideal would be to make a GUI (optional for the project) and one particular DB for each instrument and a common for the UTs.

What you will need:
-python 3.5 +
-sqlite3
-argparse or pyqt5 (CLI or GUI)

<u>Material:</u>

- This document
- fake logs from instruments /logs/*log
- sqlite3_in_nutshell.py this will give you in less than 10 commands, the basic of sqlite3 database management in python.

A-the logs

I created for this project 3 fake log for the UT2 instruments. Each log can be found in the log directory and contain 2000 lines of log. Each line is composed of :

- number of the line,
- date
- hour
- instrument
- log message

The log message is 'No problem! ALL is fine' if there is no problem. If a problem occures, an error message is displayed. For instance:

Error z: something ununderstandable

B-managing database in python

In the piece of code sqlite3_in_nutshell.py I summarized the basic command of the sqlite3 module that will help you using it.

In practice, what can you do?

First of all you have to decide what how to proceed. There is, as usual, not only one way to write a program/code/software.
Here are the things you have to decide. They can be splitted into **3 parts**:

1- Access to the log file:

- How do you read it?
- How do you access a specific line in it?
- How do you fetch an error in the log file into the error database? (see below)

2- Create the database and update it:

- What are the entry of the database?
- When you fetch the error from the log file into your program, how do you check if it already exist in the database?
- if it already exists how do you update it?

- if it is not how do you create a new entry?

3- How do you interact with the user?

Here you have few choices. I propose you to choose between two different solution:

- A command line interface that you could make like this:

```
python paranal_error_db.py -l logfile.log -s line_in_log -i instrument
```

Then you can use different 'input' statements to give information to the user and ask him question. Finally you can propose a plot of the history of a given error.

- A graphical user interface where you would display everything in a window. It could go to something like the figure below. You would go with buttons to fetch from the log monitor. If you want to plot the history of the error you just fetch you can include a matplotlib graph in it (https://matplotlib.org/examples/user_interfaces/embedding_in_qt5.html). For pyqt5 a great tutorial can be found here: http://zetcode.com/gui/pyqt5

I strongly suggest to start with the CLI that we have seen during the camp. If you have time you can start thinking about the GUI.