Bastani Judge Server Guide

Bastani Judge:

Bastani Judge is a simple ACM judge written in python, for python acm contests. It is developed by Omid Davoodi, Mohammad Reza Barazesh and Hooman Behnejad of Iran University of Science and Technology. The most unusual aspect of this judge, is the fact that it uses the python "socket" module as the mean for transferring the data between participants and the host.

Setting up the server:

To setup the server, you should first edit the two files SERVERCONFIG.TXT and SERVERPARTICIPANTS.TXT in the judge folder

SERVERCONFIG.TXT:

HOST: Put the IP address of the server here. Meaning that you should use "your" IP. 127.0.0.1 doesn't work.

PORT: Put the intended port. If you put X here, the judge will use port X for submissions and port X+1 for sending the results of the contest. So you need to make sure both those ports are available to you and the participants.

PYTHONPATH: The path of your python 3.3 executable.

PYTHON27PATH: The path of your python 2.7 executable.

CONTESTNAME: Then name of the contest

PROBLEM: use the following syntax to add problems to your contest: ProblemName->ProblemIDInServer

For example "PROBLEM=a->00001" will add problem 00001 (a+b problem) to the contest as problem "a"

START: the time of the start of the contest. Use this python program to get the time you want:

import time

OFFSET = seconds from now to the start of the contest (e.g. if you want the contest to begin 300 seconds from the time you write this, set it to 300. Generally it is better to change this just before starting the contest)

print(int(time.time())+OFFSET)

PENALTY: Number of seconds a person gets penalty for a failed submission.

SERVERPARTICIPANTS.TXT:

Here you add all of the participants in the syntax described:

Username(Student Number):Nick Name:Password

For example:

91521131:Omid Davoodi:1234

You should also delete any “ContestName_Log.txt” from the directory of the server.

Important notes to tell the participants:

1: They should disable their firewall, or let python get past. If not, they won’t be able to send or receive any content.

2: They should change their CONFIG.TXT files before attempting any submission.

3: If they want to submit their answers in python 2.7, They should add the following line to the beginning of their submission:

#2.7

4: Generally it is better to use python 3.3 for submissions, as the judge is designed for that particular version

5: They can see the results of the contest by running “judgeresultclient_Python_X.X.pyc”. A file called results.html will be generated that contains the results of the contest.

Running the server:

To run the server, you need python 3.3 . The judge accepts submissions for python 2.7 and there is a python 2.7 compatible version of the client in the directory “BastaniJudgeClientSide”. Make sure all of the participants have that, and only that folder on their computer. Also make sure they have set their CONFIG.TXT files.

Run “judgeserver-Python-3.3.py” and “judgeresultsserver-Python-3.3.py”. The first one will handle the submissions and the second one will handle the “results” table.

Notes when running the server:

You might want to see the result table of the contest the same way the participants do. For this, you have a copy of “judgeresultclient” in the server directory. You just should change the IP in the CONFIG.TXT file that is present in the server folder to match your IP. Then run the client to have the html file generated. There is no need to change username or password in CONFIG.TXT for this action.

You might want to change the configuration while running the server. For it to register the changes to the config files (SERVERCONFIG.TXT or SERVERPARTICIPANTS.TXT) while the server is running, you should first set the file CONFIG.TXT present in the server folder in this way:

HOST: Your IP. 127.0.0.1 doesn’t work.

STUDENTNUMBER=admin

PASSWORD=tolombe

Then, run the copy of the “judgeclient_Python_3.3.py” present in the server folder. Choose to send “CONFIG.TXT”, for the first problem (e.g. if the first problem in the contest is “a”, you should write “a”),

and press “enter”. You should now see the message “The configuration of the server is reset” in the server console.

How to end the contest and some other notes about the judge:

There is no technical “end” to the contest, and it can be decided by the host (By closing the “judgeserver-Python-3.3.py”). Also you can rerun the server anytime you want, meaning that you can have contests with multiple “rounds”. You can also close the “resultserver” minutes before the end of the contest to increase the tension and excitement in the match. Then you can rerun the “resultserver” some minutes after the end for the participants to see the final results.

The reason closing and restarting the server, or the resultsserver, is possible is because the results are generated from “ContestName_log.txt” file, which is updated each time someone submits an answer, and is independent of time, or the fact that the server is closed and restarted multiple times.

Adding your own problems:

The judge enables you to create your own problem sets for the contests. To add a problem, follow these steps:

1: Write “the” answer to the problem in python.

2: Give it a set of inputs for a single run (Meaning that you shouldn’t get inputs for “multiple” runs).

3: Put these inputs in a file called “problemname.in” in “problems” directory in the server folder. The inputs should be separated by “enters”.

4: Run this code in your system’s command interface (cmd in windows, shell in linux)

python “the address of the answer program you wrote” < “the address of problemname.in” > “the address of the problems folder, followed with ‘/problemname.out’ ”

5: Create a file called “problemname.conf” in the same folder, put this line in it:

TIMEOUT=2

You can change then number “2” to any other number of seconds of timeout you want to set for the problem.

6: Test the new problem in a dummy contest.