CS Games 2016

Debugging Competition

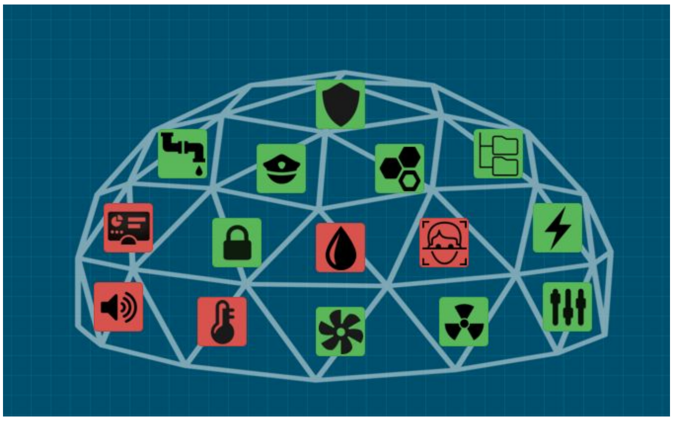
Participants	2
Workstations	1
Value	6%
Duration	3 hours

Breathe

For the government, the Dome is an essential resource which needs protection. This is why we always need more citizens capable of maintaining this 600-year old monster, falling little by little into ruins. Rare are the people still mastering the technologies from the past, pushing the council into searching for new talents to acquire, even amongst the survivors out of the Dome. The council expects to use this competition to measure your capabilities to patch and debug vital systems for the Dome's stability.

You must maintain an architecture comprised of 15 bugged programs. These programs are expected to read specific data and produce reliable outputs based on these inputs. This architecture manages two vital resources: oxygen and energy. These two resources are primordial for your survival and the citizens. After a short period without on of these and you will end up dead. When the competition starts, both resources are at maximum capacity.

Every round (5 minutes), a part of these resources is consumed by the citizens of the Dome. Thus, both gauges are reduced until exhaustion. To counter these losses, every vital system must be brought back online. Each system you manage to repair will produce a small amount of resources on each round.



At the end of each round, the central server rebuilds each system and checks for valid outputs. Two outcomes are therefore possible:

- the system produces the expected outputs: the bug has been fixed and the system is now operational: it can produce oxygen and energy.
- the system fails to produce the expected output or does not manage to compile: the bug has not been fixed, and no oxygen or energy is produced.

Competition initialization

Here is a quick summary on how to begin this competition. Be quick when configuring your environment, the clock is already ticking!

In the following examples, replace TEAM and IP by the values provided to you.

```
$ git clone ssh://TEAM@IP:9291/home/TEAM/debug
$ cd debug/
$ ./clone_repos.sh IP TEAM
```

Repair a program and push modifications to upstream

To repair a program, you will need to fix its source code. Then you must commit your changes and push them on the central repo for your team.

Modify the source code:

```
$ cd bug_hello
$ vim src/hello.nit # make the necessary changes
```

You can test your changes locally with a set of test inputs. Use the diff command to check the differences between your output and the expected one.

```
$ make test
Compile bin...
Run tests...
 * [OK] test1
 * [FAIL] test2 (diff out/test_02.out tests/test_02.res)
...
$ diff out/test_02.out tests/test_02.res
```

Then push all said modifications on your team's git server:

```
$ git add -A
$ git commit -m “Fix message”
$ git push origin master
```

Points will be attributed periodically and are visible through the scoreboard.