# Code Profiling Report

Miguel Angel Bermeo Ayerbe
Guillermo Lopez Diest
David Vegas Romero
Javier Manjón Prado

October, 2018

## 1 *Perf* tool

In order to analyze the code, we used two programs: *Perf* and *Valgrind*. *Perf* was used to obtain information on how much time was spent on each function defined in the code. The image below is the resulting report.

As we can see, almost have of the processing time is spent in the function *isAlive*. This function is used to check each position (and its surroundings) in the matrix to determine if the cell it contains should stay alive, die or be created. Knowing this, it is not unexpected that this process is the one where most time is spent.

```
sudo sh -c 'echo 1 >/proc/sys/kernel/perf_event_paranoid'
perf record
perf report
```

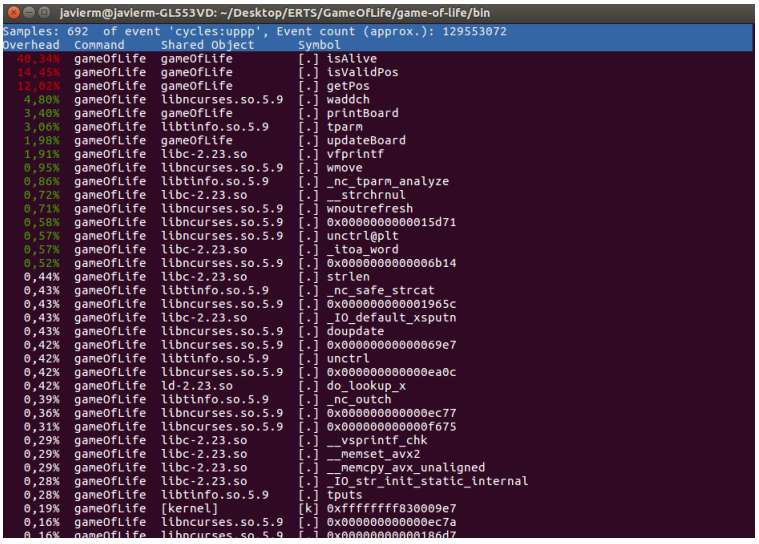


Figure 1: Perf report.

## 2 *Valgrind* tool

When using *Valgrind* to check for memory leaks and memory allocation statistics, we can see that there are no leaks in the code. There are many more allocations that frees in the code, but these are done within the library of *ncurses*. The only memory allocation used in the code is to assign memory for the board.

```
valgrind --tool=memcheck --leak-check=full ./gameOfLife
```

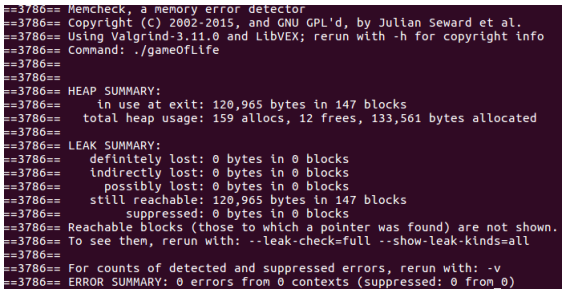


Figure 2: Valgrind report.