Sistemas Distribuídos

Air Lift debugging

António Rui Borges

Why debugging a concurrent program is tricky?

Opposite to what happens in a *sequential* program, where running it with the same input data the same output data is always produced, meaning that one is facing a *deterministic* situation; in a *concurrent* program, this does not happen. The same input data can, and usually will, produce different output data.

Why it is so? The reason is that when there is a competition of different threads for access to a common resource, the way in detail how the interaction takes place is not controlled by the application programmer. It is the scheduler of the operating system that decides the processor(s) assignment to the different threads and this is made in a manner completely transparent to the user.

Thus, one may say there is a *random factor* involved and, in successive runs, the assignment order may not be the same.

How to deal with the situation? - 1

- try and prove the solution correctness before running the program
  - be sure that all accesses to non-preemptable resources (the shared regions, in this case) are made in mutual exclusion and that all methods upon entering the critical region, always exit it (check carefully what happens in all circumstances, both normal and exceptional)
  - be sure that all variables defined inside the shared regions, belonging to the internal data structure, are only changed within the critical region
  - be sure that if a thread blocks, because conditions are not present for it to proceed, means are present for another thread to wake it up later on
  - remember that if you are using semaphores to implement synchronization, blocking must be performed *outside* the critical region
- perform single runs of the program
  - check carefully, after each run, the logging file (line by line) in order to ensure that the solution evolves as specified
  - if there is a deadlock, and this will probably happen for the first several runs, check carefully the last few lines of the logging file and try to figure out what are the threads that are blocked and why

How to deal with the situation? - 2

- perform multiple runs of the program
  - use a shell script similar to the following

```
for i in $(seq 1 500)
do
  echo -e "\nRun n. " $i
  java main.AirLiftConc
done
```