The group has initiated a study of the application of this technique to the verification of operating system structures. This technique allows one to describe and infer complex structural invariants. Furthermore, in the last five years, the group has engaged in the design of the MemCAD framework for memory data refinement. This transformation provides information about the alarm context, which may help to disprove it manually or automatically. Even in the case of bugs, as demonstrated by the Heartbleed security hole: due to a wrong protocol implementation, many websites could leak private information over years.

Verifying safety critical systems (such as avionics systems) is an important motivation to compute such properties. Indeed, a fault in an avionics system, such as a runtime error in the fly-by-wire command software, may cause an accident, with loss of life. As these systems are also very complex and are developed by large teams and maintained over long periods, their verification has become a crucial challenge. Safety critical systems are not limited to avionics: software runtime errors in cruise control management systems were recently blamed for causing unintended acceleration in certain Toyota models (the case was settled with a 12 billion dollars fine in March 2014, after years of investigation and several trials). Similarly, other transportation systems (railway), energy production systems (nuclear power plants, grid power management), and medical systems (pacemakers, surgery and patient monitoring systems) rely on complex software, which should be verified.

Beyond the field of embedded systems, other pieces of software may cause very significant harm in case of bugs, as demonstrated by the Heartbleed security hole: due to a wrong protocol implementation, many websites could leak private information, over years.

An important example of semantic properties is the class of safety properties. A safety property typically specifies that some undesirable event will never occur, whatever the execution of the program that is considered. For instance, the absence of runtime error is a very important safety property. Other important classes of semantic properties include liveness properties (i.e., properties that specify that some desirable event will eventually occur) such as termination and security properties, such as the absence of information flows from private to public channels.

All these software semantic properties are not decidable, as can be shown by reduction to the halting problem. Therefore, there is no chance to develop any fully automatic technique able to decide, for any system, whether or not it satisfies some given semantic property.

The antique group focuses on the design of semantic analysis techniques that should be sound (i.e., compute semantic properties that are satisfied by all executions) and automatic (i.e., with no human interaction), although generally incomplete (i.e., not able to compute the best—in the sense of: most precise—semantic property). As a consequence of incompleteness, we may fail to verify a system that is actually correct. For instance, in the case of verification of absence of runtime error, the analysis may fail to validate a program, which is safe, and emit false alarms (that is reports that possibly dangerous operations were not proved safe), which need to be discharged manually. Even in this case, the analysis provides information about the alarm context, which may help disprove it manually or refine the analysis.

In the last five years, the group has engaged in the design of the MemCAD framework for memory data structures abstraction, which allows to describe and infer complex structural invariants. Furthermore, it has initiated a study of the application of this technique to the verification of operating systems components.
Assignment

The purpose of this Post-Doctorate position is to improve the techniques to design and implement safe Operating System components. It will investigate the verification that operating system components generated by a high level programming environment satisfy important correctness properties.

The Ipanema environment, designed in the WHISPER project team aims at simplifying the design of scheduling policies. It relies on a domain-specific programming language that features high level abstractions (processes, events, waiting queues) adapted to the design of kernel schedulers. The purpose of this language is to ease the design of a scheduler. Moreover, the Ipanema compiler generates C code and checks that a scheduling policy satisfies a set of consistency and security rules. These rules guarantee that a scheduler cannot break the Linux kernel at run-time.

We would like to verify that the C code generated by the Ipanema compiler meets high level correctness structural properties, such as the integrity of the data structures that are manipulated inside the code of the scheduler. For instance, we would like to make sure that the scheduler functions may not lose records describing a process. This will increase the confidence in the generated schedulers.

The MemCAD framework consists of a library of memory abstractions, that are tailored for specific families of data-structures (including abstract domains for the description of arrays, lists and other inductive dynamic data-structures). It computes program invariants statically (with no execution) and automatically, by abstract interpretation of C programs. Recent works have allowed to extend it to deal with data-structures involving sharing, through a careful abstraction of sets of memory locations. This work can be used as a basis to verify the structural correctness of the code generated by Ipanema, though new abstractions will need to be designed so as to express the data-structure invariants of the schedulers, and the existing static analysis algorithms will need to be improved.

During this Post-Doctorate position, the candidate will carry out the following tasks:

1. Generation of test cases using the Ipanema environment.
2. Assessment of the MemCAD abstractions and static analysis algorithms on the generated code of these test cases.
3. Design of new abstract domains so as to allow the verification of scheduler implementations generated with the Ipanema environment, and integrate them into the MemCAD framework.
4. Assess the results of the analysis of the generated code using the resulting analysis.

More generally, this work will allow to study an original approach to the design of operating system components, that involves both high level languages featuring domain specific abstractions and the static verification of generated code using static analysis by abstract interpretation.

Main activities

Main activities:

- Understand the theoretical foundations and the practical use of the Ipanema framework and of the MemCAD static analyzer;
- Generation of test cases using, and assessment of the results produced by the existing toolchain;
- Design of static analysis techniques (abstract domains, static analysis components) to allow the validation of the case studies;
- Implementation and experimental validation of the techniques proposed;
- Publication of the results.

Additional activities:

- Communication with researchers working on connected topics

Skills

The candidate should have a good understanding of programming languages techniques (compilation, program analysis) and of formal methods.

A large part of the work will be carried out in English (scientific communication, writing or articles).

Benefits package

- Subsidised catering service
- Partially-reimbursed public transport

Remuneration

- Location: Paris 12ème
- Gross Salary per month: 2 653€ brut/mensuel

Security and defense procedure
This position is likely to be situated in a restricted area (ZRR), as defined in Decree No. 2011-1425 relating to the protection of national scientific and technical potential (PPST).

Authorisation to enter an area is granted by the director of the unit, following a favourable Ministerial decision, as defined in the decree of 3 July 2012 relating to the PPST. An unfavourable Ministerial decision in respect of a position situated in a ZRR would result in the cancellation of the appointment.