Institute of Engineering and Management of Grenoble Alpes University

Grenoble INP - UGA is a member of **international** engineering and management education and research **networks.** It is widely recognized in national and international rankings**.**

8 schools + **39** laboratories
8300 students
1 300 teaching, research, administrative and technical staff

Grenoble INP - UGA is a renowned public institution of higher education and research, and a major player in the Grenoble ecosystem. It is the engineering and management institute of Grenoble Alpes University, and plays a leading role in the scientific and industrial community.

PhD in Digital Electronics

Job number	2024-PHDDIGITELECT-LIG
Research field	Digital Electronics (Embedded systems design, digital architecture design in disruptive technologies)
Host laboratory	(UMR 5217 (Grenoble-INP, UGA and CNRS) / Website : www.liglab.fr/
Required profile	PhD / Master
Location	Grenoble, France
Hiring date / contract term	01/05/2024 (36 months contract)
Contacts	alain.tchana@grenoble-inp.fr

Grenoble INP - UGA is a leading public institution accredited with the French label "Initiative d'excellence". It offers innovative engineering and management programs, with an increasing internationalization of its course offers. The courses are grounded in sound scientific knowledge and linked to digital, industrial, organizational, environmental and energy transitions. The Engineering and Management Institute of Grenoble Alpes brings together more than 1300 staff members (teacher-researchers, lecturers, administrative and technical staff) and 8300 students, located on 8 sites (Grenoble INP - Ense3, Grenoble INP - Ensimag, Grenoble INP - Esisar, Grenoble INP - Génie industriel GI, Grenoble INP - Pagora, Grenoble INP - Phelma, Polytech Grenoble, Grenoble IAE and the INP Prepa). Grenoble INP is also a highly-ranked institution of higher education and research, leading the way in the fields of engineering and management on an international scale. It is a member of a large number of international academic and research networks. It is part of the European University UNITE!.
As part of Grenoble Alpes University, Grenoble INP has associated guardianship of 39 national and international research laboratories and of technological platforms. The research conducted there benefits both its socio-economic partners and its students. Grenoble INP is at the heart of the following scientific fields: physics, energy, mechanics and materials; digital; micronanoelectronics, embedded systems; industry of the future, production systems, environment; management and business sciences.

Grenoble INP - UGA is an equal opportunity employer committed to sustainability. Grenoble INP-UGA celebrates diversity and equity and is committed to creating an inclusive environment for all employees. All qualified applications will be considered without discrimination of any kind.

Research

Description of the laboratory and its immediate research environment (team):

The LIG laboratory is essential in the field of computer science research, at both the local, national, and international levels. It boasts nearly 450 researchers, faculty members, doctoral students, and research support staff. The ambition of LIG is to contribute to the development of fundamental aspects of computer science (models, languages, methods, algorithms) and to foster synergy between the conceptual, technological, and societal challenges associated with this discipline.

LIG is organized around five thematic research axes, including Distributed Systems, Parallel Computing, and Networks, which comprise five teams. The thesis work will take place within the ERODS project of LIG (erods.imag.fr), under the supervision of Alain TCHANA. The team conducts research on the construction and administration of operating systems and distributed systems. It is particularly interested in virtualization mechanisms, as well as the autonomy and robustness of systems.

Job description :

Over the past decade, the exponential advancement of digital technology in our societies (healthcare, politics, security, transportation, entertainment, etc.) has led to the emergence of a wide variety of applications and hardware platforms that operating systems (OS) must accommodate. Given these diverse environments, traditional monolithic operating systems (such as Linux, FreeBSD, Mac, and Windows) have become challenging to maintain, vulnerable, and suboptimal. To illustrate this research challenge, consider the versatile memory reclamation subsystem of Linux, kswapd, which comes into play during memory pressure situations. kswapd prioritizes least recently used (LRU) pages for eviction from RAM. However, when this policy encounters a media player application that reads a byte and then moves on without revisiting it, kswapd performs poorly. For such an application, the most recently used (MRU) pages should be evicted first from RAM. Customizing kswapd presents a challenge and requires machine reboot. The fundamental issue with popular OSes lies in their monolithic nature, which mandates the execution of all services in kernel space, making customization complex. How can services in monolithic OSes be easily and swiftly customized and deployed without interruption?

Revival of the microkernel approach The need for customization and speed (rapid deployment) of OS services has become increasingly urgent over the last decade. This is attributed to the emergence of numerous new applications (AI, search engines, key-value stores, virtual machines, etc.) and hardware acceleration pedals (NUMA, asymmetric CPUs, GPUs, TPUs, PIM, NVRAM, etc.). Generic policies in a monolithic OS become unsatisfactory (in terms of performance) for all environments (application and hardware combinations). Two approaches are currently being explored to customize OS services. The first approach, hot kernel hooking, involves evolving the monolithic OS by injecting code at runtime. This approach is based on two assumptions: the presence of hooks in the OS and the presence of a virtual machine in the OS capable of securely executing the injected code. In Linux, most contributions are based on the BPF framework. For example, we can mention Syrup. The second approach is simply a return to microkernels. However, instead of developing microkernels from scratch, researchers are pragmatic: they try to transform the monolithic Linux OS into a microkernel. This strategy ensures adoption of the resulting OS as it will remain a Linux OS, capable of running existing applications without any modification. We support the microkernel approach as the former presents several limitations. Firstly, code injection frameworks in Linux impose strong constraints on control structures and the accessibility domain of injected code. For instance, BPF prohibits infinite loops, and BPF code cannot access or modify data structures manipulated by the OS. Secondly, this approach relies on the availability of hooks in the OS services one wishes to customize. In summary, this approach does not allow for elaborate policy writing.

The goal of MiLK is to provide a Linux standard/model upon which developers will rely to externalize Linux services into user space following the microkernel approach. The major innovation of MiLK is the introduction of a new abstraction called MilkT (for MiLK Task), much like Process and Thread tailored for execution of OS services in user space. Indeed, OS services have different needs compared to application services for which Thread and Process concepts were designed. We have identified several limitations with these two concepts, the most significant ones being:

• The virtual address space of the OS when in kernel space is contiguously mapped in physical memory, which facilitates address translation. By using the classic process concept to externalize OS services, this optimization is no longer guaranteed.

• Some OS services share or act upon the same data structures. This is the case, for example, with the scheduler and memory manager, which manipulate the struct_task structure in Linux. This is impossible when the process abstraction is used to externalize them into user space. Using the Thread abstraction would solve this problem, but would also introduce the issue of fault tolerance of services relative to each other. The reader may think that the recently developed orbital abstraction by Jing et al [orbit.osdi.22] would suffice. This is not true as orbit was invented to represent auxiliary tasks of other main tasks. In the context of MiLK, all OS services are equally important.

• Using system calls or filesystems /proc or /sys as classic processes do would incur considerable overhead due to the fact that OS services communicate extensively.

• Given their criticality, OS services in user space must natively enforce fault tolerance policies, which is unnecessary for application processes. The goal of the doctoral thesis is to study this aspect of MiLK.

Specific requirements

The ability to work in both French and English is essential. Additionally, international experience will be an added advantage.

Specifics of the position

The research may be led on 2 locations: Grenoble and St Martin-d'Hères.

Position assigned to a restricted area: YES

(Device for the protection of the scientific and technical potential of the nation, conditioning the appointment of the researcher to the authorization of the Defense Security Officer).

How to apply

Applications must be sent to : alain.tchana@grenoble-inp.fr

Application deadline : 27/03/2024