Software Reliability from Machine Code

Introduction

The growing reliance on Commercial Off-The-Shelf (COTS) components and the emergence of (RISC-V) software-centered satellites [1],[2] represent significant advancements in aerospace technology. These innovations have brought about new paradigms in satellite design, where software plays a pivotal role in operational effectiveness and mission success. Consequently, there is an increased emphasis on software reliability [3], which is fundamentally critical due to the potential ramifications of software failures in space environments.

Central to understanding software reliability is the need to obtain quantitative insights into its performance metrics. Unlike traditional hardware reliability, software reliability must account for complex behaviors arising from dynamic interactions among components. One of the key metrics for assessing software reliability is its relationship with resource utilization, specifically memory and register occupation. In essence, this relationship implicates how a software application utilizes its resources, which inherently affects its reliability.

In parallel, within the domain of real-time systems, monitoring register occupation and worst case execution time (WCET) has garnered attention as a vital metric. The effective tracking of register usage allows developers to optimize their applications, ensuring that critical tasks can execute reliably under time constraints.

The aim of this thesis is to evaluate software reliability at a machine code level, applying techniques from the domain of real-time software development.

Expected outcome

- Literature review on the fundamentals of reliability analysis from machine code.
- Use tools (aiT [4], OTAWA [5]) to obtain relevant parameters from machine code.
- Validate results with FI campaign or radiation experiment.

Requirements

- Completed DCS and RTS1 courses.
- Interest in embedded systems, real-time systems or dependable computing
- Programming experience in C/C++ and/or Python
- Experience with fault injection, compilers, or hardware platforms is benificial

Contact

Kuan-Hsun Chen (k.h.chen@utwente.nl)
Tijmen Smit (t.t.smit@utwente.nl)

References

1. S. Di Mascio, A. Menicucci, G. Furano, C. Monteleone, and M. Ottavi, “The Case for RISC-V in Space,” in Applications in Electronics Pervading Industry, Environment and Society, vol. 573, S. Saponara and A. De Gloria, Eds., in Lecture Notes in Electrical Engineering, vol. 573. , Cham: Springer International Publishing, 2019, pp. 319–325. doi: 10.1007/978-3-030-11973-7_37.
2. Pignol, M., Malou, F., Aicardi, C. (2019). COTS in Space: Constraints, Limitations and Disruptive Capability. In: Velazco, R., McMorrow, D., Estela, J. (eds) Radiation Effects on Integrated Circuits and Systems for Space Applications. Springer, Cham. doi: 10.1007/978-3-030-04660-6_12
3. B. Fang, Q. Lu, K. Pattabiraman, M. Ripeanu, and S. Gurumurthi, “ePVF: An Enhanced Program Vulnerability Factor Methodology for Cross-Layer Resilience Analysis,” in 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Toulouse, France: IEEE, Jun. 2016, pp. 168–179. doi: 10.1109/DSN.2016.24.
4. https://www.absint.com/ait/index.htm
5. Clément Ballabriga, Hugues Cassé, Christine Rochange, Pascal Sainrat. OTAWA: An Open Toolbox
   for Adaptive WCET Analysis. 8th IFIP WG 10.2 International Workshop on Software Technologies for Embedded and Ubiquitous Systems (SEUS), Oct 2010, Waidhofen/Ybbs, Austria. pp.35-46, doi: 10.1007/978-3-642-16256-5_6.