Detecting binary incompatible software components using dynamic loader

Áron Baráth, Zoltán Porkoláb

Department of Programming Languages and Compilers, Eötvös Loránd University

`[baratharon,gsd]@caesar.elte.hu`

Modern programming languages support modular development dividing the system into separate translation units and compile them individually. A linker is used then to assemble together these units either statically or dynamically. This process, however, introduces implicit dependences between the translation units. When one or more units are modifyed in inconsistent way binary incompatibility occurs and may result unexpected program behavior. Current mainstream programming languages neither specify what are the binary compatibility rules nor provide tools to check them. In this paper we discuss the details of various cases of binary incompatibility. We implemented a prototype solution in the Welltype programming language [**?**, **?**] to detect binary compatibility by dynamic loader.

References

[1] Á. Baráth and Z. Porkoláb, "Welltype: Language elements for multiparadigm programming," in *Position Papers of the 2017 Federated Conference on Computer Science and Information Systems*, pp. 91–101, 2017. `http://dx.doi.org/10.15439/2017F546`.

[2] Á. Baráth, "Welltype project web page." `http://repo.hu/projects/welltype`, 2018.