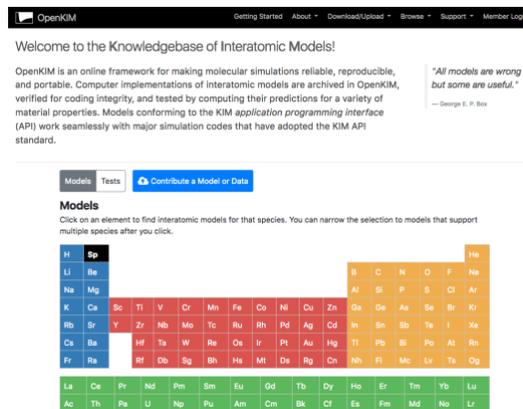# Software Engineer Needed to Implement Windows Development for OpenKIM

University of Minnesota, Spring 2020

Software specialists are being sought for a short-term project to port the KIM API (https://openkim.org/kim-api/) to Windows systems. The position is available immediately (January 2020) and is expected to run for one to three months.

The project is part of a major international effort funded by the National Science Foundation (NSF) to develop standard methods in atomistic simulations of materials called the "Open Knowledgebase of Interatomic Models" (OpenKIM). This includes the development of an application programming interface (API) standard for atomistic simulations. OpenKIM is led by Profs. Ellad Tadmor, Ryan Elliott and George Karypis at the University of Minnesota and Prof. Mark Transtrum at Brigham Young University. You can learn more about KIM here: https://openkim.org/about/

The KIM API is a multi-language (C, C++, Fortran) dynamically loaded library (DLL) framework designed around an explicit-link plug-in architecture that allows an atomistic simulation engine (such as LAMMPS) to systematically access an extensible library of independently written interatomic model DLLs (OpenKIM Models). Currently, the KIM API is designed for use on Linux systems and uses the standard *dlfcn.h* API and a custom CMake configure and build framework. Binary distribution of the KIM API and OpenKIM Models library is currently available via standard mechanisms for macOS (homebrew), Fedora, OpenSUSE, Ubuntu, and Conda.

The objective of this job involves two stages: (1) port the KIM API and OpenKIM Models repository to the Windows operating system, and (2) create a maintainable electronic packaging and distribution mechanism that easily fits into a Windows user's standard software installation/upgrade workflow. Both stages can be done by a single developer or separately depending available skills.

Candidates should have experience in these areas:
- Expertise with DLLs on Windows, including **implicit (load-time dynamic) linking** with the use of associated "import libraries" and **explicit (run-time dynamic) linking**
- Expertise in Fortran, C and C++ DLL creation on Windows
- Working knowledge of the **Linux shared library** framework and the *dlfcn.*h API
- Working knowledge of CMake development on Linux and Windows systems
- Expertise in creation, submission, and maintenance of binary packages for Windows

For more information or to apply, please send a resume to Prof. Ryan Elliott at relliott@umn.edu.