Systematizing Error Handling in Linux

Résumé du projet de recherche (Langue 1)

The C programming language does not provide any built-in error handling abstractions such as the exceptions found in Java and other languages. Thus, the Linux developer has to explicitly construct code for handling each error. Such code typically involves freeing all previously allocated resources and returning some error number to describe the error condition. In a study of Linux code done in the Coccinelle project (www.emn.fr/x-info/coccinelle), we have observed that error handling code is frequently itself a source of errors, arising either when a developer adds new error handling code and overlooks some required cleanup operations or when a developer adds a new operation requiring cleanup and does not update all of the affected error handling code. This issue has indeed been noticed by Linux developers, who have introduced a coding style in which all cleanup operations are placed in a sequence at the end of the function, after a successful return, and are reached by appropriate gotos into the sequence. This style ensures that no cleanup operations are overlooked, at least if the right goto operations are used. Nevertheless, much of Linux still does not use this style, and converting to it is tedious and error prone. The goal of this PhD is to develop a tool for checking error handling code in Linux and converting it to the new style if needed.

Résumé du projet de recherche (Langue 2)

Because error handling code is complex, we will divide the development of the tool into two phases: 1. In the first phase, we will develop a checker that determines whether the provided error handling code performs all of the required cleanup operations and reports to the user any omissions that it finds. 2. In the second phase, we will develop a corrector, that converts the old form of error handling code to the new form using gotos and that inserts any missing cleanup code, as needed.

Informations complémentaires (Langue 2)

The candidate should have a background in programming languages and program analysis and an interest in open source infrastructure software.