REVERSE ENGINEERING

Kristina Goncharenko
Master's degree student, Faculty of Information Technology
branch: Computer Graphics and Multimedia in English, xgonch00@vutbr.cz

Kateryna Yatsenko
Master's degree student, Faculty of Information Technology
branch:Mathematical Metods,xyatse00@vutbr.cz

Abstract: It is well known that reverse engineering is a research of a certain device or the program, as well as documentation of them, in order to understand how it works and, most often, reproduced a device, program, or another object with similar functions, but without copying it as such. But if you go deeper into this topic, the decompilation process is an important part of reverse engineering. There are several different definitions of a decompiler in the literature. For example, a decompiler is a program that receives an input program in machine code and outputs an equivalent program in a programming language. This definition suffers from some uncertainty and excessive optimism. The definition should emphasize the close relationship between the compiler and the decompiler, as well as the fact that, as described below, the decompilation process is not unconditional and always successful.

In this talk, we are going to concentrate on the definition of the decompiler. Decompiler is a program that tries to perform the reverse process produced by the compiler: for a given executable file of a program proceeds by any high-level language, it seeks to output a program to a high-level language (and not necessarily the language in which the program was originally written) that will perform the same functions as the input executable program. We will see the general structure of binary program decompilers, as well as approaches to solving the most useful problems. In the area of data flow analysis, it is possible to delete and propagate dead registers and flags. In the sequel, we will review some results available in the literature.