Disassembling strojového kódu je proces, při kterém je binární strojový kód programu přetransformován do podoby assmbly kódu. Smyslem tohoto procesu je napomoci člověku v porozumění fungování programu, jehož zdrojový kód není znám. Strojový kód vyprodukovaný kompilátory při kompilaci zdrojového kódu je však velmi náročné číst. Na vině jsou mnohé optimalizace a transformace kódu, které kompilátor učinil. Jednou obzvláště problematickou optimalizací je instruction scheduling, jehož úkolem je pozměnit pořadí instrukcí tak, aby výsledný kód byl co možná nejrychlejší.

Cílem této práce je vyvinout disassembler schopný měnit pořadí instrukcí v kódu. Tato funkce by uživateli umožnila přeskládat instrukce tak, aby výsledek byl lépe čitelný. Aby disassembler mohl takovou funkcionalitu nabízet, musí být schopen porozumět významu jednotlivých instrukcí. Proto navrhneme kompilátor s vnitřní reprezentací nezávislé na platformě, s jejíž pomocí budeme reprezentovat libovolný strojový kód. Tuto reprezentaci pak bude možné použít k nalezení závislosti mezi instrukcemi, které budou dále použity pro změny jejich pořadí v kódu. Na konci práce probereme možnost emulace běhu programu.