УДК 004.42

ДЕКОМПІЛЯЦІЯ ЯК ЗАСІБ РЕІНЖЕНЕРІЇ ПЗ

Неміш В.М. [1], **Демковський Ю.І.** [2]
Тернопільський національний економічний університет, магістрант
[1] *к.ф.-м.н., доцент;* [2] *магістрант*

І. Постановка проблеми

Створення та розробка складних програмних систем різного призначення часто ведеться за допомогою інтеграції окремих компонент, виконаних як власними, так і сторонніми розробниками. Це дозволяє значно скоротити вартість і час розробки програмного забезпечення. Зовнішні модулі можуть поставлятися без вихідного коду. В якості одного з засобів для підвищення рівня абстракції представлення програми може використовуватися декомпіляція.

Декомпіляція - це процес автоматичного відновлення програми на мові високого рівня з програми на мові низького рівня. Декомпілятор інструментальний засіб, що одержує на вхід програму на мові асемблера або інше аналогічне низькорівневе уявлення і видає на вихід еквівалентну їй програму на деякій мові високого рівня.

Також декомпіляція може використовуватися для забезпечення сумісності програмних додатків, а саме для аналізу протоколів взаємодії у разі, коли вони описані недостатньо повно або не описані взагалі. Декомпіляція дозволяє спростити відновлення станів і структур даних протоколу взаємодії.

В силу вищесказаного є актуальною розробка декомпілятора в припущенні, що вихідна програма була реалізована на конкретній мові програмування. Декомпілятор визначається як транслятор з мови низького рівня в мову високого рівня, який мінімізує кількість артефактів трансляції в результуючій програмі й виконує відновлення високорівневих конструкцій цільової мови максимально повно в припущенні, що вихідна програма була написана на цій мові високого рівня.

ІІ. Мета роботи

Метою науково-дослідної роботи є розробка методів та інструментальної середовища для декомпіляції програм. Алгоритми і шаблони, які використовуються для декомпіляції, повинні бути застосовні до широкого спектру цільових низькорівневих програм і дозволяти автоматично повно відновити всі високорівневі конструкції цільової мови високого рівня.

Висновки

У роботі запропоновано поліпшити існуючу класичну архітектуру декомпілятора додавши в неї нову фазу, що дозволяє більш якісно провести розбір виконуваного файлу.

На відміну від існуючих робіт, був розроблений і докладно описаний алгоритм отримання проміжної форми подання і спроектовані структури даних для зберігання цього проміжного представлення. Описано алгоритми, що здійснюють аналіз потоку даних на основі цього внутрішнього подання.

Треба відзначити, що в рамках даної роботи був реалізований прототип, що працює тільки з виконуваними файлами сімейства ОС Microsoft Windows (PE-файли), але в майбутньому планується його розширити для роботи з іншими видами виконуваних файлів. Можливість подібного розширення вже закладена в архітектуру прототипу.

Список використаних джерел

1. Chang B.Y.E., Harren M., and Necula G.C. The 13th International Static Analysis Symposium (SAS) // Analysis of Low-Level Code Using Cooperating Decompilers. 2006. pp. 318-335.
2. Durfina L., Kroustek J., Zemek P., Kolar D., Hruska T., Masarik K., and Meduna A. Advanced Static Analysis for Decompilation Using Scattered Context Grammars. Angers, France. 2011. pp. 164-170.