

ПРОФИЛИРОВЩИК ИСХОДНОГО КОДА

Бобрик П.В.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Леванцевич В.А. – старший преподаватель

Описана актуальность использования профилировщиков исходного кода, как инструмента оценки эффективности выполнения программ. Проведен сравнительный анализ работы профилировщика инструментального типа в зависимости от способов его вызова. Показано, что время вызова кода профилирования практически не зависит от способа вызова

Оценка эффективности выполнения программ является актуальной и востребованной задачей. Для этого используются различные инструменты, одними которых является профилировщики.

Профилирование – сбор характеристик программы, во время её работы. Профилировщики могут собирать множество информации о работе программы, среди которой можно выделить время выполнения фрагментов кода, используемый объём памяти, число кэш-промахов, количество верно предсказанных условных переходов, на каком ядре выполнялся код и т.д. [1]. При профилировании также может учитываться информация о работе периферийных устройств, например работа конвейера видеокарты и время на чтение/запись у внешних накопителей.

Существует несколько основных видов профилировщиков, отличающихся по способу изъятия информации из целевой программы: статистические и инструментальные [2]. Статистические профилировщики работают в отдельном процессе и получают информацию о работе целевой программы путём семплирования целевого процесса через примерно равные промежутки времени. Получение информации о работе целевого процесса происходит при помощи системных вызовов, работа целевого процесса при этом практически не нарушается, но точность измерений ограничена промежутком семплирования. Инструментальные профилировщики вызывает сам процесс, который профилируется. В этом случае информация о работе процесса получается как через системные вызовы, так и кодом, например, вызовом инструкции RDTSC [3,4]. У каждого из способов имеются свои достоинства и недостатки, которые следует учитывать при работе с конкретным инструментом.

Был разработан профилировщик инструментального типа и проведены исследования скорости его работы в зависимости от способов вызова:

- подстановка кода профилирования в исходный код целевой программы путём макроподстановок или средствами компилятора;
- вызов кода профилирования, как функции в исходном коде;
- вызов кода профилирования, как функции из динамической библиотеки.

Суть исследования заключалась в проведении множественных замеров времени, затраченного на выполнения каждого из перечисленных способов, и вычисления их среднеарифметического значения. Для замеров использовался шестидесяти четырёх битный счётчик процессора Time Stamp Counter (TSC) [4]. Начальные условия проведения исследований:

- процессор Intel i5-8265U;
 - тип оперативной памяти DDR4;
 - количество проведённых замеров 30 миллионов;
- Результаты исследований приведены в таблице 1

Таблица 1 – результаты исследования

Способ вызова программного кода	Среднее количество тактов процессора
Подстановка кода в исходный код	26.1050244937
Вызов кода, как функцию	26.1048281396
Вызов кода из динамической библиотеки	26.1055323698

Анализ таблицы показывает, что при работе профилировщиков инструментального типа разница между тем, будет ли код профилирования подставляться в исходный код программы, вызываться в качестве функции из целевого кода или из библиотеки, незначительна. Поэтому при разработке профилировщика следует учесть, что эти три способа его вызова с точки зрения временных затрат на их выполнение практически одинаковы.

Список использованной литературы:

1. *Fundamentals of Performance Profiling* [Электронный ресурс] – Режим доступа: <https://smartbear.com/learn/code-profiling/fundamentals-of-performance-profiling/> - Дата доступа 10.04.2024.
2. *Introduction to profiling* [Электронный ресурс] – Режим доступа: <https://www.chpc.utah.edu/presentations/images-and-pdfs/Profiling20.pdf> - Дата доступа 10.04.2024.

3. *How does QueryPerformanceCounter measure time?* – Режим доступа: https://www.youtube.com/watch?v=pZ0MF1q_LUE
– Дата доступа 10.04.2024.
4. *Intel® 64 and IA-32 Architectures Software Developer's Manual.* – Intel inc. Order Number: 325462-080US, 2023 – 5066 с.