

ТРАНСЛЯТОР С С89 В С11/С17

Москва 2021

Содержание

1	Аннотация	2
2	Миграция	2
3	Миграция кода	3
3.1	Способы и инструментальные средства	4
4	Стандарты языка C	5
5	Решение	5
5.1	Формирование правил перехода от старых конструкций языка к новым	5
5.2	Прототип транскомпилятора	6
6	Заключение	6
7	Список литературы	7

1 Аннотация

Миграция кода является актуальной задачей в работе с унаследованной кодовой базой. Необходимость миграции с одной версии языка на актуальную позволяет избежать уязвимостей старых версий, избежать ошибок (исправленных в новых версиях языка), повысить быстродействие и эффективность работы кода. Однако, данная задача является сложной и до сих пор наблюдается дефицит соответствующих инструментальных средств, позволяющих мигрировать с одного языка на другой в автоматическом режиме или хотя бы позволяющих облегчить этот процесс.

Статья посвящена миграции кода с языка C89/C99 на C11/C17.

Ключевые слова: *портирование программного обеспечения, переносимость программ, миграция кода*

2 Миграция

Миграция (англ. Migration) – процесс перехода от одних инструментальных средств к другим. Например от одного языка программирования на другой. Сам процесс миграция встречается в базах данных, прикладных и веб-приложениях, при переходе с одной версии языка на другую. С миграцией связаны следующие проблемы [1]:

1. Неорганизованность – перед началом миграции необходимо составить соответствующий план
2. Потеря данных – при переносе данных из одного места в другое, они могут потеряться. Поэтому необходимо использовать резервные
3. Вопросы совместимости – при переходе на новую версию инструментального средства (язык, библиотека, фреймворк и т.д.) может вызывать проблемы с обратной совместимостью, так как в новых версиях может отсутствовать ранее присутствующая функциональность

4. Проблемы с оборудованием – при переходе на новую аппаратную или программную среду (операционную систему) необходимо учитывать их возможности и совместимость

3 Миграция кода

Миграция кода – процесс непосредственного перехода от одной версии языка на другую версию языка. Например, с C++11 на C++17.

Основной причиной миграций код является выпуск новых версий языка, фреймворков, библиотек или даже появление более совершенных инструментальных средств. Не менее важным является процесс миграции кода, который должен быть периодическим, как его тестирование и отладка. Вот несколько причин, по которым это необходимо:

- Исправление ошибок в новых версиях языка
- Уточнение поведения языка
- Добавление новых возможностей
- Повышение производительности за счёт улучшения работы компилятора
- Повышение надёжности и безопасности кода
- Исправление ошибок в работе компилятора
- Избавление от устаревших возможностей

Примеры миграций приведены в источниках [1][2][3][4], которые подтверждают актуальность данного вопроса.

3.1 Способы и инструментальные средства

Способы миграции условно можно разделить на:

- Ручной – разработчики самостоятельно изменяют или переписывают код
- Автоматический – использование инструментальных средств позволяющие облегчающих миграцию

Ручной способ менее предпочтителен, так как может повлечь за собой появление ошибок, искажение работы исходных алгоритмов. Помимо этого данный способ требует очень много времени на осуществление миграции.

Автоматический же способ наиболее предпочтителен, так как сокращает участие человека в процессе миграции и даже может оптимизировать получающийся код, в свою очередь это сократит количество потенциальных ошибок. Однако, сложность состоит в том, что практически нет средств, осуществляющие полноценную миграцию без участия разработчика. Сложность, в разработке таких средств, состоит в том, что не смотря на строгую формальную систему языка необходимо учитывать и понимать семантику.

Примеры инструментальных средств:

1. Porting Assistant for .NET[5]
2. Port Assist[6]
3. CodeCheck[7]
4. cvtto64[8]
5. Delphi Parser[9]

Дополнительный список инструментальных средств с Windows to Linux, Solaris to Linux, Linux to Linux, Other UNIX to Linux[[10]].

4 Стандарты языка C

Здесь привожу ключевы отличия C89/C99 от C11/C17 и надо составить "таблицу перехода" для разных стандартов, например: C89->C11, C99->C11, C89->C17, C99->C17

Далее я предлагаю решение о том, что по полученной таблице сравнения сформировать "правила" перехода (ниже чуть описал примеры)

5 Решение

Задачу трансляции старых исходных кодов из C89/C99 в C11/C17 можно решить следующими способами:

- Использование регулярных выражений
- Использование абстрактного синтаксического дерева

Однако, в обоих способах необходимо выделить правила перехода от старых языковых конструкций к новым.

5.1 Формирование правил перехода от старых конструкций языка к новым

Например, функция не имеет аргументов. Однако, при её вызове можно указать фактические аргументы в виде литер или переменных.

```
void [function_name]()
{
    // code
}

// [function_name](1,2,3,4) <- compilation ok
```

В соответствии с новыми стандартами следует писать так

```
void [function_name](void)
{
// code
}
```

```
// [function_name](1,2,3,4) <- compilation error
```

В формальном виде можно было бы записать: [C89]:void *() -> [C11]:void *(void)
ещё пару примеров

5.2 Прототип транскомпилятора

Транскомпилятор должен принимать исходный код на C89/C99 и выдавать новые исходные код на языке C11/C17. Помимо этого он должен быть кроссплатформенным и работать в операционных системах Windows, Linux, MacOS X. Также он должен представлять собой консольное приложение, принимающие соответствующие аргументы.

```
user@example:~$ CtoC -stdIn=C89 -stdOut=C11 -in=/old -out=/new
```

6 Заключение

Портирование кода, со старой версии стандарта на новый, может дать следующие преимущества:

- Рост производительности
- При введении повсеместной практики портирования старого кода, новые версии языка могут стать более компактными, более ограниченными

7 Список литературы

1. Common Data Migration Problems [Электронный ресурс]. – URL: <https://www.vxchnge.com/blog/data-migration-problems-solutions> (Дата обращения: 24.08.2021)
2. Как мы перевели 10 миллионов строк кода C++ на стандарт C++14 (а потом и на C++17) [Электронный ресурс]. – URL: <https://habr.com/ru/company/1c/blog/429678/> (Дата обращения: 24.08.2021)
3. Миграция баз данных: зачем и почему [Электронный ресурс]. – URL: <https://lanit.ru/press/smi/migratsiya-baz-dannykh-zachem-i-pochemu/> (Дата обращения: 24.08.2021)
4. HipHop for PHP [Электронный ресурс]. – URL: https://en.wikipedia.org/wiki/HipHop_for_PHP (Дата обращения: 24.08.2021)
5. Migrating C/C++ Software to Mobile Platforms in the ADM Context [Электронный ресурс]. – URL: https://digital.cic.gba.gob.ar/bitstream/handle/11746/5678/11746_5678.pdf-PDFA.pdf?sequence=1&isAllowed=y (Дата обращения: 24.08.2021)
6. Porting Assistant for .NET – Amazon Web Services [Электронный ресурс]. – URL: <https://aws.amazon.com/ru/porting-assistant-dotnet/> (Дата обращения: 17.08.2021)
7. PortAssist: Visual Analysis for Porting Large Code Bases [Электронный ресурс]. – URL: <https://quelknas.myqnapcloud.com:8081/uploads/PAPERS/InfoVis11/poster.pdf> (Дата обращения: 17.08.2021)
8. CodeCheck Program Overview – URL: <https://www.abxsoft.com/codchk.htm> (Дата обращения: 17.08.2021)

9. Code migration tools – IBM Documentation [Электронный ресурс]. – URL: <https://www.ibm.com/docs/en/ztpf/2019?topic=tools-code-migration> (Дата обращения: 24.08.2021)
10. Delphi Parser [Электронный ресурс]. – URL: https://delhiparser.com/?gclid=EAIaIQobChMIk0Gr80HB8gIVAamyCh0Lnw_ZEAAyAiAAEgK0wPD_BwE (Дата обращения: 24.08.2021)
11. Porting and Migration Tools [Электронный ресурс]. – URL: https://www.novell.com/developer/porting_and_migration_tools.html (Дата обращения: 24.08.2021)
- 12.
13. – URL: https://en.wikipedia.org/wiki/Source-to-source_compiler (Дата обращения: 24.08.2021)
14. Programming language specification [Электронный ресурс]. – URL: https://en.wikipedia.org/wiki/Programming_language_specification
15. Scientific Visualization and Computer Graphics [Электронный ресурс]. – URL: <http://www.cs.rug.nl/svcg/SoftVis/Refactor>
16. Д.В. Силаков. Методы обеспечения переносимости ПО // Статья, 2009
17. – URL: <https://valgrind.org/> (Дата обращения: 17.08.2021)
- 18.