

меня данный подход, мы значительно увеличим производительность, при этом код будет написан в современном безопасном стиле.

Был приведен один из типовых примеров использования `Span<T>`. Для получения дополнительной информации о способах и примерах использования `Span<T>` смотрите [1, 2].

Заключение. Использование языков программирования и платформ с системами автоматического управления памятью и сборкой мусора при разработке современных приложений значительно уменьшило количество ошибок и положительно повлияло на разработку программного обеспечения в целом. Однако, еще остается ряд ситуаций, в которых для оптимизации производительности приходится писать «небезопасный» код с использованием указателей. Предложенный компанией Microsoft подход с применением класса `System.Span<T>` позволяет использовать современные безопасные подходы для работы с непрерывными областями памяти, значительно увеличивая производительность приложения.

1. Toub, S. All About Span: Exploring a New .NET Mainstay / S. Toub // MSDN Magazine. – 2018. – V. 33, N. 1.

2. Writing High-Performance Code Using `Span<T>` and `Memory<T>` in C# [Электронный ресурс]. – Режим доступа: <https://www.codemag.com/Article/2207031/Writing-High-Performance-Code-Using-SpanT-and-MemoryT-in-C>. – Дата доступа: 30.01.2023.

ОБ ОТОБРАЖЕНИИ СТРОКОВЫХ ЛИТЕРАЛОВ В ПРОСТРАНСТВО ТИПОВ ЯЗЫКА ПРОГРАММИРОВАНИЯ СИ++11

*С.В. Сергеевко
Витебск, ВГУ имени П.М. Машерова*

Обработка текстовых данных является одной из основных составляющих информационных технологий. Вопрос эффективной программной реализации такой обработки тесно связан с представлением и обработкой констант на этапе трансляции. Язык программирования Си++ позволяет описывать обработку констант с помощью шаблонов и `constexpr` функций. Однако строковые литералы не могут быть параметрами шаблонов и результатами функций [1].

Цель работы – получить представление строковых данных (констант) программы, доступное компилятору на этапе трансляции, средствами языка программирования Си++ стандарта 2011 года.

Материал и методы. Материалом исследования являются способы описания в исходном коде программы преобразований строковых литералов на этапе трансляции. Поставленная цель достигается средствами обобщенного программирования по средствам шаблонов в языке программирования Си++ с учетом возможностей стандарта 2011 года.

Результаты и их обсуждение. Для представления строк предлагается размещать отдельные символы строки в параметрах шаблона. То есть, для каждого строкового литерала строить экземпляр следующего шаблона класса:

```
template<class T, T ... items>
struct Pack {};
```

Для того чтобы допускать в качестве элементов строки символы различных типов остальные шаблоны классов, используемые для работы с описываемым представлением строки будут описываться внутри специального шаблона класса

```
template<class C>
struct Char_u {
//...
};
```

Для построения типа по строковому литералу необходимо использовать в качестве инициализатора для массива символов и передать его в качестве параметра шаблона `Unstring_tf`. Фактическая реализация сосредоточена в шаблоне `Unstring_tf_impl` и его частичных специализациях:

```
template<const C *s, size_t i, const C c, C ...
chars>
struct Unstring_tf_impl
: Unstring_tf_impl<s, i + 1, s[i + 1], chars ..., c>
{};
```

```
template<const C *s, size_t i, C ... chars>
struct Unstring_tf_impl<s, i, '\0', chars ...> {
    using value = Pack<C, chars...>;
};
```

```
template<const C *s>
struct Unstring_tf : Unstring_tf_impl<s, 0, *s> {};
```

Заключение. Был описан способ получения представления строковых литералов в исходном коде в виде пригодном для его обработки, выполняемой на этапе трансляции, что позволяет задействовать более широкий спектр выполняемых компилятором оптимизаций.

1. Вандевурд, Д. Шаблоны C++. Справочник разработчика / Д. Вандевурд, Н. Джосаттис, Д. Грегор. – 2-е. – СПб. : «Альфа-книга», 2018. – 848 с.

2. Сергеенко, С. В. Об организации обработки строковых литералов во время компиляции средствами языка C++ / С. В. Сергеенко // Наука – образованию, производству, экономике : материалы 72-й Регион. науч.-практ. конф. преподавателей, науч. сотрудников и аспирантов, Витебск, 20 февраля 2020 г. – Витебск : ВГУ имени П. М. Машерова, 2020. – С. 29–30. URL: <https://rep.vsu.by/handle/123456789/20799> (дата доступа: 02.02.2023).