

процесс разработки. Независимо от того, добавляете ли вы новые функции, исправляете ошибки или отправляете данные для тестирования, миграции предлагают мощный набор инструментов для эффективного и результативного управления этими изменениями.

Литература

- 1) Amazon [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/blogs/database/database-migration-what-do-you-need-to-know-before-you-start/>. – Дата доступа: 25.04.2024.
- 2) PhoenixNap [Электронный ресурс]. – Режим доступа: <https://phoenixnap.com/blog/database-migration>. – Дата доступа: 25.04.2024.
- 3) CloudBees [Электронный ресурс]. – Режим доступа: <https://www.cloudbees.com/blog/database-migration#choosing-the-best-option-for-you>– Дата доступа: 24.04.2024.
- 4) .Net Code Chronicles [Электронный ресурс]. – Режим доступа: <https://amarozka.dev/entity-framework-migrations/>. – Дата доступа: 26.04.2024.
- 5) Forproger [Электронный ресурс]. – Режим доступа: <https://forproger.ru/article/migracii-i-isxodnye-dannye-s-entity-framework-core>. – Дата доступа: 26.04.2024.

УДК 004.223

СИМВОЛЬНЫЕ КОДИРОВКИ В СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЯХ

Кабыш Я.А.

Научный руководитель – Воронич Л.В., ассистент

Стремительное развитие информационных технологий в современном мире привело к необходимости постоянной работы с текстовой информацией, которую нужно было как-то хранить в памяти компьютера. Для решения этой проблемы были придуманы различные методы кодировки информации, каждая из которых имеет свои преимущества и недостатки. Первым методом кодировки стал формат ASCII (American standard code for information interchange) — это стандартный код символов, который используется для представления текстовой информации в компьютерных системах и обмена данными между различными устройствами и программами. ASCII использует 7-битный код (0-127) для представления основных символов английского алфавита, цифр, знаков препинания и

управляющих символов. Первые 31 символ являются управляющими и не отображаются на экране, остальные нужны для отображения символов.

С распространением компьютерных технологий появилась потребность в кодировке, которая могла бы поддерживать и другие языки, кроме английского. Для этого была разработана кодировка ANCI (American National Standards Institute), которая расширяла ASCII за счет использования пустого 8 бита. Она начала использовать кодовые страницы – набор из 256 символов для конкретного языка. Первые 128 символов полностью совпадают с ASCII, остальные используются для конкретного языка. Например, для русского языка была создана кодовая страница windows-1251.

Однако существование таких языков, как, к примеру, китайский не позволяло использовать эти кодировки, так как 8 бит информации не хватало на все иероглифы. Так был создан новый формат Unicode.

Unicode - это стандарт кодирования символов, который предоставляет универсальный метод представления текстовой информации на компьютере, независимо от языка или письменной системы. В отличие от ASCII, который ограничен символами английского алфавита и некоторыми знаками препинания, Unicode включает в себя широкий набор символов, охватывающих практически все существующие языки мира, а также символы из различных культур и областей, математические и специальные символы, эмодзи и т. д.

Основной проблемой нового стандарта стало вдвое увеличившееся потребление памяти, так как любой символ стал занимать 2 байта, и во многих случаях использование кодировки ASCII оставалось более целесообразным.

Для решения данной проблемы была создана кодировка UTF-8 (Unicode Transformation Format, 8-bit) — это переменной длины кодировка Unicode переменной длины, которая позволяет представлять символы Unicode с использованием одного, двух, трех или четырех байтов. Она была разработана для обеспечения эффективного хранения и передачи текстовой информации на компьютерах, поддерживая одновременно как символы из основного многоязычного набора Unicode, так и дополнительные символы и символы редких языков. Первые 128 символов также были равны первым 128 символам кодировки ASCII, поэтому эти символы хранятся в памяти, используя только 1 байт.

Развитие кодировок продолжилось и вскоре был разработан UTF-16 (Unicode Transformation Format, 16-bit) — это кодировка Unicode, которая использует 16-битные кодовые единицы для представления символов. Этот формат позволил хранить более 2 байт информации и предоставлял совместимость со старой кодировкой Unicode. UTF-16 стал использовать для кодирования специальный метод, названный суррогатной парой.

Суррогатная пара в UTF-16 — это пара двух 16-битных кодовых единиц, используемых в кодировке UTF-16 для представления символов из дополнительных плоскостей Unicode, которые не могут быть представлены одной кодовой единицей. Каждая часть суррогатной пары называется суррогатом. Первая часть (high surrogate) находится в диапазоне от 0xD800 до 0xDBFF, а вторая часть (low surrogate) находится в диапазоне от 0xDC00 до 0xDFFF.

Однако использование суррогатных пар замедлило обработку информации, и для решения этой проблемы была придумана кодировка UTF-32.

UTF-32 (Unicode Transformation Format, 32-bit) - это кодировка Unicode, которая использует 32-битные кодовые единицы для представления каждого символа. Каждый символ в UTF-32 представлен одним 32-битным кодом, что делает эту кодировку фиксированной длины.

UTF-32 обеспечивает прямое соответствие между символами Unicode и их представлением в памяти, что обеспечивает простоту работы с символами и их манипулирование. Однако, UTF-32 требует больше памяти для хранения текста по сравнению с UTF-8 и UTF-16, особенно для текста, в основном состоящего из символов ASCII.

Одного наилучшего решения так и не было найдено, и каждая из кодировок используется для решения своих задач:

- UTF-8 занимает в 4 раза меньше памяти, чем UTF-32, поэтому используется для передачи файлов по сети.
- UTF-16 и UTF-32 работают быстрее и предоставляют больше символов, поэтому используются в языках программирования.
- Кодировка ASCII также используется за счет своей простоты и малого размера занимаемой памяти до сих пор в проектах, где не требуется большое количество различных символов.

Исходя из всего вышперечисленного, можно сделать вывод, что использования кодировок символов является обязательной необходимостью для работы с тестовой информацией, а их многообразие позволяет выбрать оптимальное решения под каждую конкретную задачу.

Литература

1. Васильева, И. Н. Криптографические методы защиты информации: учебник и практикум для вузов / И. Н. Васильева. - Москва: Издательство Юрайт, 2024. - 349 с.
2. Вульф А. Криптография. Основы практического шифрования и криптографии / А. Вульф - «Издательские решения», 2023. – 100 с.