Проект№14 «Локалізація»

Виконав студент

1-ї Статистики

Агієнко Ігор

1. **Основні поняття**
   1. С (широкі символи, юнікод, бібліотека <locale>)

**Широкі символи**

Широкий символ — термін, що стосується програмування. Він використовується для позначення символьного типу даних, який ширший за традиційні 8-бітні символи. Це не те саме, що Юнікод.

wchar\_t — це тип даних стандарту ANSI/ISO C (а також використовується в інших мовах програмування) для подання широких символів.

У Windows API тип wchar\_t має назву як WCHAR і має фіксований розмір 16 біт, що не дозволяє кодувати весь набір символів Unicode (більше 1 мільйона). Тому порушується стандарт ANSI/ISO C, який вимагає, щоб символьний тип wchar\_t підтримував усі подані в системі символи в одному об'єкті wchar\_t. По суті, у WinAPI під WCHAR мають на увазі 2-байтове слово з кодування UTF-16LE (як тип WORD), тому символи з кодами вище FFFF16 кодуються парою WCHAR (так звані «сурогати») і всім API-функціям передається не кількість символів, а розмір символьного масиву в машинних словах.

У бібліотеці ANSI C за широкі символи відповідають заголовні файли <wchar.h> і <wctype.h>.

**Юнікод**

Юнікод - промисловий стандарт, розроблений, щоб забезпечити цифрове представлення символів усіх писемностей світу та спеціальних символів. Юнікод складається з асортименту символів, методології кодування та комплекту (набору) стандартів кодування символів, комплекту кодових таблиць для посилань на зображення символів, списку властивостей символів таких, наприклад, як верхній і нижній регістр (розкладка), комплект довідкових даних комп'ютерних файлів, правил нормалізації, декомпозиції, зіставлення і зображення.

Документи, закодовані за стандартом UNICODE, можуть містити в єдиному тексті японські та китайські ієрогліфи, літери латиниці, кирилиці, грецького алфавіту (α, ε, θ, π, σ, λ, φ, Ω ...), математичні символи, символи музичної нотної нотації , символи вимерлих, рідкісних, екзотичних народностей. При цьому немає необхідності в перемиканні кодових сторінок.

UTF-8 - для подання кожного символу використовуються 4 байта, безпосереднє чисельне значення коду UNICODE

UTF-16 - для подання найбільш часто використовуваних символів використовуються 2 байта (перші 65536 позицій), а інші видаються у вигляді у вигляді «сурогатних пар». Таке кодування використовується в операційних системах Windows починаючи з Windows NT.

UTF-32 - для подання кожного символу використовуються сукупність електронних даних змінної довжини: від 1 байта для символів основної таблиці ASCII, до 6 байт для рідко використовуваних символів (символи російського алфавіту кодуються 2-мя байтами). Це кодування створювалася пізніше інших для операційних систем Plan 9 і Inferno в 1992р.

Для роботи та перетворення мультибайтних послідовностей записаних в кодуванні UTF-8 в C / C ++ вводиться сімейство функцій виду mb \* (): mbtowc (), mblen (), mbstowcs (), wcstombs () і ін. Це механізм взаємних перетворень між масивами char [] (в яких також виражаються рядки UTF-8) і wchar\_t []. Якщо ви не стикаєтеся з кодуванням UTF-8 (що з великою ймовірністю має місце в Windows), то ця група функцій вас не повинна займати.

Аналогічно, замість контейнерного класу C ++ string вводиться аналогічний контейнерний клас широких символів wstring.

16-бітові символи Unicode (кодування UTF-16) використовуються в Windows повсюдно; для внутрішнього представлення імен файлів і шляхів доступу в файлової системі NTFS також використовується Unicode. Якщо визначена символьна константа \_UNICODE, то всі виклики функцій Windows вимагають використання рядків, що складаються з розширених символів; в іншому випадку рядки 8-бітових символів перетворюються в розширені рядки.

**Бібліотека <locale>**

locale.h — заголовний файл стандартної бібліотеки мови програмування С, котрий використовується для задач пов'язаних з локалізацію. Цей заголовний файл надає дві ключові функції: localeconv та setlocale. Перша забезпечує доступ до поточної локалі, у той час, як друга дозволяє змінювати її. Заголовний файл також містить структуру struct lconv, яка містить інформацію про поточну локаль locale.

* 1. С++ (бібліотеки <locale> <codecvt>)

**Бібліотека <locale>**

C ++, зрозуміло, успадковує всі можливості C щодо рядків, які подаються як масиви char [] і wchat\_t []. Але C ++ вводить нове (і краще) об'єктної уявлення рядків string і wstring. Велика частина операцій з рядками, що реалізуються в C функціями API, реалізуються для об'єктів цих класів функціями-методами, за винятком ось таких важливих особливостей і відмінностей від рядків в стилі C:

1. Рядки C ++ можна привласнювати операцією = (копіювати значення);

2. Рядки C ++ можна порівнювати типовими операціями: ==,! =, <, <=,>,> =. Рядки порівнюються в лексографіческом порядку. Природно, що результат порівняння одних і тих же рядків залежить від обраної локалі;

3. Рядки можна конкатеніровать (об'єднувати) простою вказівкою операції + (і, відповідно + =);

4. Існує метод c\_str (), який повертає внутрішній вміст рядка у формі масиву символів (const char \*);

Як видно і з останнього твердження, змінні-об'єкти класу string / wstring – це незмінні об'єкти (в тому ж сенсі, як в мові Python і ін.). Це не означає константность, це зовсім інше:

string s = "рядок 1";

s = "рядок 2"

Тут операцією присвоєння змінної s буде присвоєно новий об'єкт, створений викликом конструктора з ініціалізувалися значенням "рядок 2". Попередній об'єкт з значенням "рядок 1 "буде знищений, для нього буде викликаний деструктор при виході з області визначення об'єкта (Блоку). Новий і старий об'єкти будуть розміщені за різними адресами. У цьому сенсі і розуміється незмінюваність: при модифікації значення об'єкта, нове значення не змінює старе, а ініціалізує новий об'єкт.

Всі ці принципи повністю переносяться і на локалізовані рядки широких символів wstring, з тієї єдиною різницею, що string є контейнером однобайтових char, а wstring – це контейнер 4-х байтових широких символів wchar\_t.

**Бібліотека <codecvt>**

Клас std :: codecvt инкапсулирует перетворення рядків символів, включаючи широкі і багатобайтові, з одного кодування в іншу. Всі операції введення-виведення файлів, що виконуються через std :: basic\_fstream <CharT> використовують фасет std :: codecvt <CharT, char, std :: mbstate\_t> локалі, введеної в потік.