UTF-8, UTF-16, UTF-32, UCS-2, UCS-4

Ignacio Gomis Lli

Juan Pablo Uriol Balbín

UTF 8 (Unicode Transformation Format)

Puede representar cualquier carácter Unicode
Usa símbolos de longitud variable, desde 1 byte a 4, por lo que para acceder al elemento N-ésimo hay que leer la cadena desde el principio
Es compatible con US-ASCII, por lo tanto los mensajes ASCII se representan con el mismo carácter, lo cual lo hace más sencillo de decodificar
Es síncrono, por tanto, si estando en un punto se quiere leer una letra, no hace falta comenzar la secuencia desde el principio
No hay superposición, así que no se puede confundir caracteres entre sí al tomar variables de mayor extensión
Gran ahorro de espacio para textos con caracteres latinos, al ser aquellos de longitudes cortas pero los caracteres ideográficos (chino, japonés, árabe) ocupan 3 bytes, más que en UTF-16
El rendimiento de UTF-8 en coste de computación es el mayor, comparado con UTF-16 y UTF-32

UTF-16

Características:
Puede representar cualquier carácter Unicode, y una gran variedad de caracteres poco frecuentes
Símbolos de longitud variable: 1 o 2 palabras de 16 bits por carácter Unicode.
UTF-16 puede ser considerado de tamaño fijo al emplear los caracteres típicos
No hay superposición, cada palabra representa un único carácter

UCS-2 (Universal Character Set)

Precursor de UTF-16
El carácter se representa por su valor en 16 bits, lo que limita la cantidad representable en gran medida, con la mejora a UTF-16 se representan todos los posibles caracteres

UTF-32

Usa exactamente 32 bits por carácter, por tanto es de tamaño fijo, lo cual implica que los caracteres se puede indexar directamente, acceder al N-ésimo carácter es una operación de tiempo constante, aunque a pesar de ello, no es tan útil como pueda parecer porque hay pocos casos que sea necesario acceder a cierto bit sin leer los anteriores.
Los valores numéricos Unicode son idénticos a los correspondientes en UTF-32.
Su principal desventaja es el coste espacial, ya que usa 4 bytes por carácter y realmente los símbolos poco comunes apenas se usan, causando que el mismo mensaje en UTF-16 ocupe prácticamente la mitad o incluso una cuarta parte UTF-8, sin pérdida de información

UCS-4

Precursor de UTF-32
Cada carácter se representa por un valor de 31 bits
Se pasó a UTF-32 porque el estándar ISO prohibió varios de los bits que empleaba para estandarizar UTF-16

Tiempos:
Mapa ordenado: 0.0403513 s
Mapa desordenado: 0.0171857 s

Como es de esperar los tiempos con la clase de C++11 unordered_map son mucho menores