UTF-7: Перерождение

Если воспринимать Code Point’ы Юникода как целые числа, то UTF-8 тратит больше байт в сравнении с LEB128; чтобы сделать LEB128 совместимым с ПО для работы с ‘\0’-терминированными строками, достаточно добавить запрет на ‘\0’-символ в мультибайтовых LEB128-последовательностях.

Итак, UTF-7 символ имеет в длинну от 1-го до 3-х байт:

- Множество 1-байтовых символов – это ASCII
  (маска 0XXXXXXX)
- Сабсет 3-байтовых символов состоит из символов, невошедших в 1- и 2-байтовые наборы
  (маска 1XXXXXXX 1XXXXXXX 0XXXXXXX)
- Для определения набора 2-байтовых символов необходимо проделать работу по анализу частоты появления различных символов в текстах

Ёмкость 2-байтового UTF-7 сабсета в ~8 раз выше, чем у сабсета UTF-8 с тем же числом байт: $(2^{14} – 2^{7})$ и 2^{11} соответственно; ~16 KiSym (КибиСимволов) достаточно, чтобы вместить:

- Все символы “фонетических” алфавитов немёртвых языков
- Самые популярные эмодзи
- 3000+ наиболее часто используемых японских иероглифов
- 5000+ символов китайской письменности

Для того, чтобы UTF-7 стал заменой UTF-8, пропагандой (промоушеном) должен заняться кто-то, кто свободно говорит по-английски (т. е., определённо не я).

Что касается возможных проблем с безопасностью при интерпретации UTF-7 как UTF-8, то их ничуть не больше, чем при интерпретации любого другого случайного набора байтов как UTF-8.