

UTF ENCODING

(UTF-8, 16, 32)

유니코드의 Character Set

- UCS-2 (Universal Character Set 2)
 - 2Byte Character Set
 - 1개의 언어판 (BMP)을 정의
- UCS-4 (Universal Character Set 4)
 - 4Byte Character Set
 - 32,768 언어판 정의

왜 인코딩이 필요한가?

- UCS-2,4 는
 - 유니코드를 저장하는 변수의 크기를 정의
 - But, 바이트 순서에 대해서 표준화하지 못했음.
- 파일처리 프로그램들이 바이트 단위로 동작
 - UCS와는 잘 맞지 않음
 - 즉, 파일을 인식 시 이 파일이 UCS-2, UCS-4인지 인식하고 각 경우를 구분해서 모두 다르게 구현해야 하는 문제 발생

We Need

Suitable external encoding of Unicode

유니코드 인코딩(UTF)

- UTF(Unicode Transformation Format)
- UTF-8(in web)
 - MIN: 8bit, MAX: 32bit(1 Byte * 4)
- UTF-16(in windows, java)
 - MIN: 16bit, MAX: 32bit(2 Byte * 2)
- UTF-32(in unix)
 - MIN: 32bit, MAX: 32bit(4 Byte * 1)

UTF-8

코드범위	UTF-8	설 명
000000-00007F 1Byte	0xxxxxxx	ASCII와 동일한 범위 (MSB = 0)
000080-0007FF 2Byte	110xxxxx 10xxxxxx	첫 바이트는 '1'로 그 문자를 표시: 110(2Byte) or 1110(3 바이트) 나머지 바이트: 10
000800-00FFFF 3Byte	1110xxxx 10xxxxxx 10xxxxxx	
010000-10FFFF 4Byte	11110zzz 10zzxxxx 10xxxxxx 10xxxxxx	SMP 영역

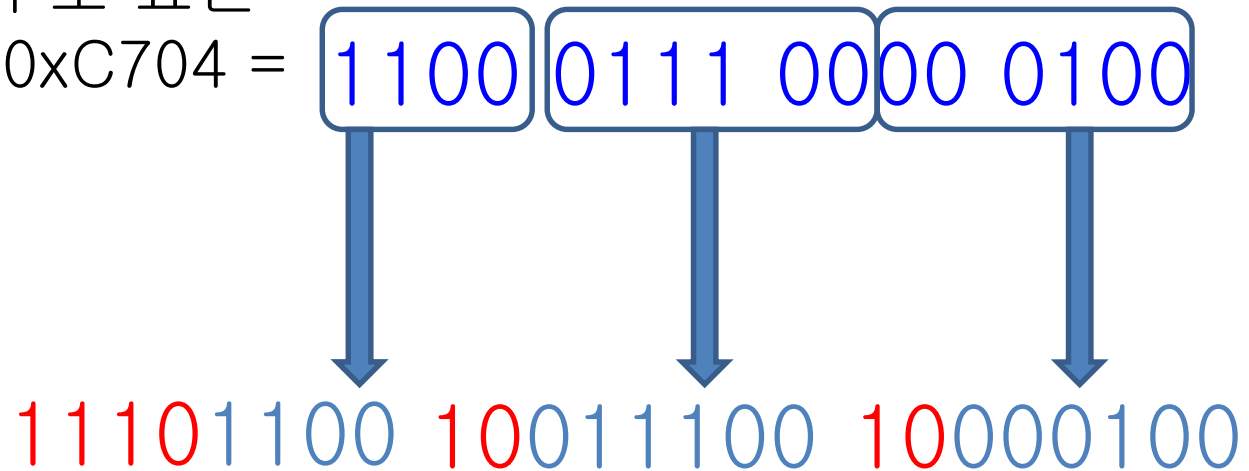
C700

Hangul Syllables

	C70	C71	C72	C73	C74	C75	C76	C77	C78	C79	C7A
0	웁 C700	윽 C710	유 C720	음 C730	은 C740	읏 C750	일 C760	익 C770	읷 C780	자 C790	잠 C7A0
1	웰 C701	윰 C711	육 C721	읍 C731	을 C741	응 C751	ړ C761	ړ C771	ړ C781	작 C791	잡 C7A1
2	윸 C702	윹 C712	윺 C722	읃 C732	응 C742	읆 C752	ړ C762	ړ C772	ړ C782	작 C792	잡 C7A2
3	웁 C703	윽 C713	유 C723	음 C733	은 C743	읏 C753	일 C763	익 C773	읷 C783	자 C793	잠 C7A3
4	위 C704	웁 C714	윺 C724	읃 C734	응 C744	읆 C754	ړ C764	이 C774	임 C784	잔 C794	쟈 C7A4

- 2진수로 표현

‘위’: 0xC704 =



- Unicode ‘위’(0xC704) 는
UTF-8로 3바이트(EC 9C 84)로 인코딩 됨

The UniSearcher: Search Results - Windows Internet Explorer

http://www.isthisthingon.org/unicode/index.phtml

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(D) 도움말(H)

연결 국민대 종합정보시스템 네이버 다음 성곡도서관 싸이 옥션 우리은행 W 위디스크 CS EVERRICH Google MAGENTA NHN PRESSBOX 통신실험실 +

The UniSearcher: Search Results

--Jump To Character Block--

[Unisearcher Help](#) |
 [Every Character by Codepoint](#) |
 [Every Character by Block](#) |
 Searches (one at a time):

Shift-JIS (hex)	Codepoint (hex)	Codepoint (dec)	UTF8 Hex Code	UTF8 String	Description	Eng. Definition	Pronunciation	Go
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Go"/>

(URL to the results below) **Glyph Entry Key:**

Unicode Hex
 UTF-8 Hex
 Shift-JIS Hex
 Decimal Code
X

"Top-level" is the first Hex digit of the desired chunk. "Next-level" is the second Hex digit. "Last-level" is the third. Put "00" after that and you have the starting point in Hex; e.g.: 1C800 through 1C8FF if Top, Next, and Last are 1, C, and 8. Characters in **RED** have been combined with another character because they're combining forms.

Top-level:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Page/Subpage:
Next-level:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0C000 - 0CFFF (49,152 - 53,247)
Last-level:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0C700 - 0C7FF (50,944 - 51,199)

Block Nav: [Previous \(0x0C600\)](#) | [Next \(0x0C800\)](#)

Click on a character (or its name) to copy it to the [Unisearch Clipboard](#).
 Click to copy "위" to the clipboard.

No Description	
Unicode (Hex):	0C704
UTF-8 (Hex):	EC9C84
Shift-JIS (Hex):	None
Unicode (HTML):	위
Hangul Syllables (0x0AC00-0x0D7AF)	

위

No Description	
Unicode (Hex):	0C704
UTF-8 (Hex):	EC9C84
Shift-JIS (Hex):	None
Unicode (HTML):	위

Hangul Syllables (0x0AC00-0x0D7AF)

Uni-Searcher Site
<http://www.isthisthingon.org/unicode/index.phtml>

인터넷 100%

UTF-8 장점 & 단점

- 장점
 - 하위 호환성(ASCII)
 - XML문서의 표준 인코딩
 - 모든 유니코드 문자 표현 가능
 - 미리 바이트 크기를 알 수 있다
 - 간단한 비트 연산만 사용해서 효율적
- 단점
 - 크기가 크다(가변적 인코딩)
 - 문자열 처리가 간단하지 않다

UTF-16

- 인코딩의 기본 단위는 16비트(2바이트)
- 기본 언어판(BMP) 2Byte 인코딩
 - 63,488개 (= 65,536 – 2,048) 문자 표현 가능
 - 대행문자 영역 2,048개를 제외한 BMP 63,488개의 코드를 문자로 사용. ucs-2와 동일
- 보충 언어판(SMP) 4Byte 인코딩
 - U+10000 ~ U+10FFFF : 100만여개 (1,048,576개)
 - 대행문자(surrogate) 영역 2개의 16-bit 쌍을 이용
 - 16개 SMP 언어판 코드(1,048,576개 문자) 표현 가능
 - Surrogate <High, Low> : 1024*1024
 - High Surrogate : U+D800 ~ U+DBFF
 - Low Surrogate : U+DC00 ~ U+DFFF

UTF-16 인코딩 방법

내 용	UTF-8
UTF-16	yyyyyyyyy xxxxxxxxx
UTF-16 BE (Big Edian)	yyyyyyyyy xxxxxxxxx
UTF-16 LE (Little Edian)	xxxxxxx yyyyyyyy
High Surrogate	110110ZZ ZZxxxxxx
Low Surrogate	110111yy yyyyyyyy
보충 언어판 UTF-16	00000000 000zzzzz xxxxxxyy yyyyyyyy

* ZZZZ = zzzzz-1

UTF-16 인코딩

예) Old Italic Letter A (0x10300)



내 용	UTF-8
High Surrogate	Hi-surrogate = (Unicode - 0x10000) / 0x400 + 0xD800; = D800
Low Surrogate	Low-surrogate = (Unicode - 0x10000) % 0x400 + 0xDC00; = DF00
UTF-16 (보충 언어판)	0xD800_DF00

UTF-16 surrogate → UTF-32

- $\text{CodeValue} = (\text{HighSurrogate} - 0xD800) * (\text{LowSurrogate} - 0xDC00) + 0x10000$
 - 0xD800은 상위대행코드 영역의 시작점
 - 0xDC00은 하위대행코드 영역의 시작점
 - 보충 언어판은 0x10000부터 시작하므로
- 예) Old Italic Letter A (0x10300)
 - UTF-16 : 0xD800_DF00
 - UTF-32 : $(1 * 0x300) + 0x10000 = 0x10300$



UTF-32

- 4 Byte로 모든 유니코드 문자를 표현
 - 고정 길이 인코딩
- UCS-4와 동일하지 않음
 - UCS-4의 부분집합
 - (영역이 0x 0000 0000 ~ 0x 0010 FFFF)
 - CodeValue =
$$(\text{HighSurrogate} - 0xD800) * (\text{LowSurrogate} - 0xDC00) + 0x10000 \text{ (UTF-32 시작점)}$$

= U+10000 ~ U+10FFFF까지 1,048,576개의 값을 가짐.

ANSI, UCS-2, UTF-8

- `int WideCharToMultiByte {`
 - `UINT CodePage, // code page`
 - `DWORD dwFlags, // performance and mapping flags`
 - `LPCWSTR lpWideCharStr, // wide-character string`
 - `Int cchWideChar, // number of chars in string`
 - `LPSTR lpMultiByteStr, // buffer for new string`
 - `Int cbMultiByte, // size of buffer`
 - `LPCSTR lpDefaultChar, // default for unmappable chars`
 - `LPBOOL lpUsedDefaultChar // set when default char used`
- `};`

ANSI(MultiByte) -> UCS-2(WideChar) -> UTF-8(MultiByte)

ANSI, UCS-2, UTF-8

- IconV

```
$ iconv -f CP949 -t UTF-8 -o out.txt in.txt
```

- 입/출력 형식 지정:

- -f, --from-code=<이름> 원 문서 인코딩
 - -t, --to-code=<이름> 출력 인코딩

- 출력 조정:

- -o, --output=FILE 출력 파일

UTF 인코딩 기법 비교

	‘가’
UTF-8	EA B0 80
UTF-16BE	AC 00
UTF-16LE	00 AC
UTF-32BE	00 00 AC 00
UTF-32LE	00 AC 00 00

참고문헌 및 사이트

- The Unicode standard, version 4.0 : the Unicode consortium, Aliprand, Joan / Addison-Wesley
- Unicode demystified a practical programmer's guide to the encoding standard, Gillam, Richard / Addison-Wesley
- 위키백과 <http://ko.wikipedia.org/wiki/UTF-8>
- 유니코드 사이트 Q&A http://www.unicode.org/unicode/faq/utf_bom.html
- 유니코드 사이트 용어해설 <http://www.unicode.org/glossary/>
- 유니코드와 인코딩,
http://cafe.naver.com/q69.cafe?iframe_url=/ArticleRead.nhn%3Farticleid=44522
- 컴퓨터속의 한글 <http://b.mytears.org/2005/01/101>
- 유니코드 인코딩 컨버터 소스
<http://www.unicode.org/Public/PROGRAMS/CVTUTF/>
- Uni-Searcher Site, <http://www.isthisthingon.org/unicode/index.phtml>