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

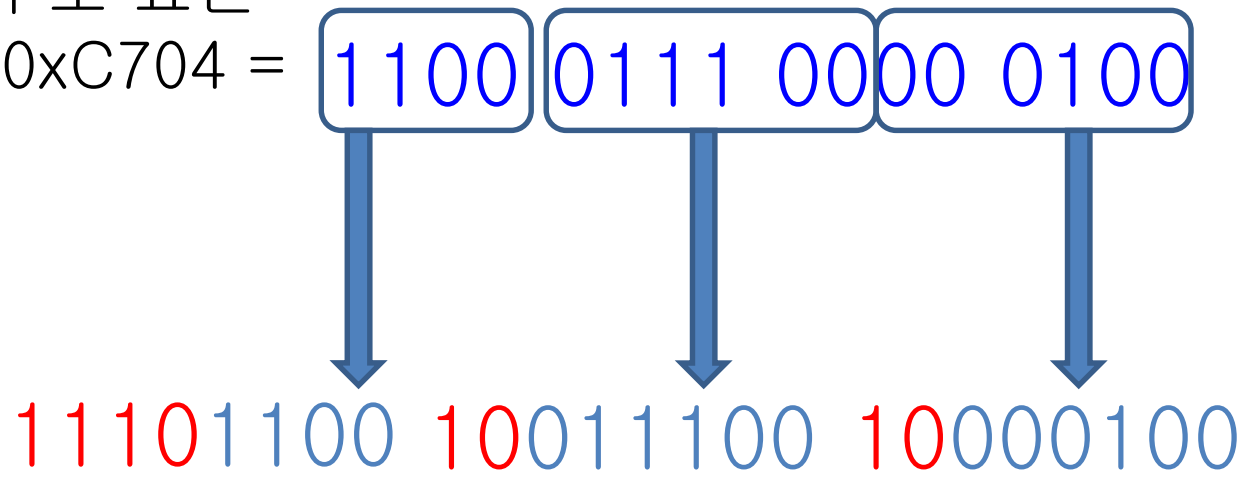
코드범위	UTF-8	설 명
000000-00007F 1Byte	0xxxxxxx	ASCII와 동일한 범위 (MSB = 0)
000080-0007FF 2Byte	110xxxxx 10xxxxxx	첫 바이트는 '1'로 그 문자를 표시: 110(2Byte) or 1110(3 바이트) 나머지 바이트: 10
000800-00FFFF 3Byte	1110xxxx 10xxxxxx 10xxxxxx	
010000-10FFFF 4Byte	11110zzz 10zzxxxx 10xxxxxx 10xxxxxx	SMP 영역

C700 Hangul Syllables

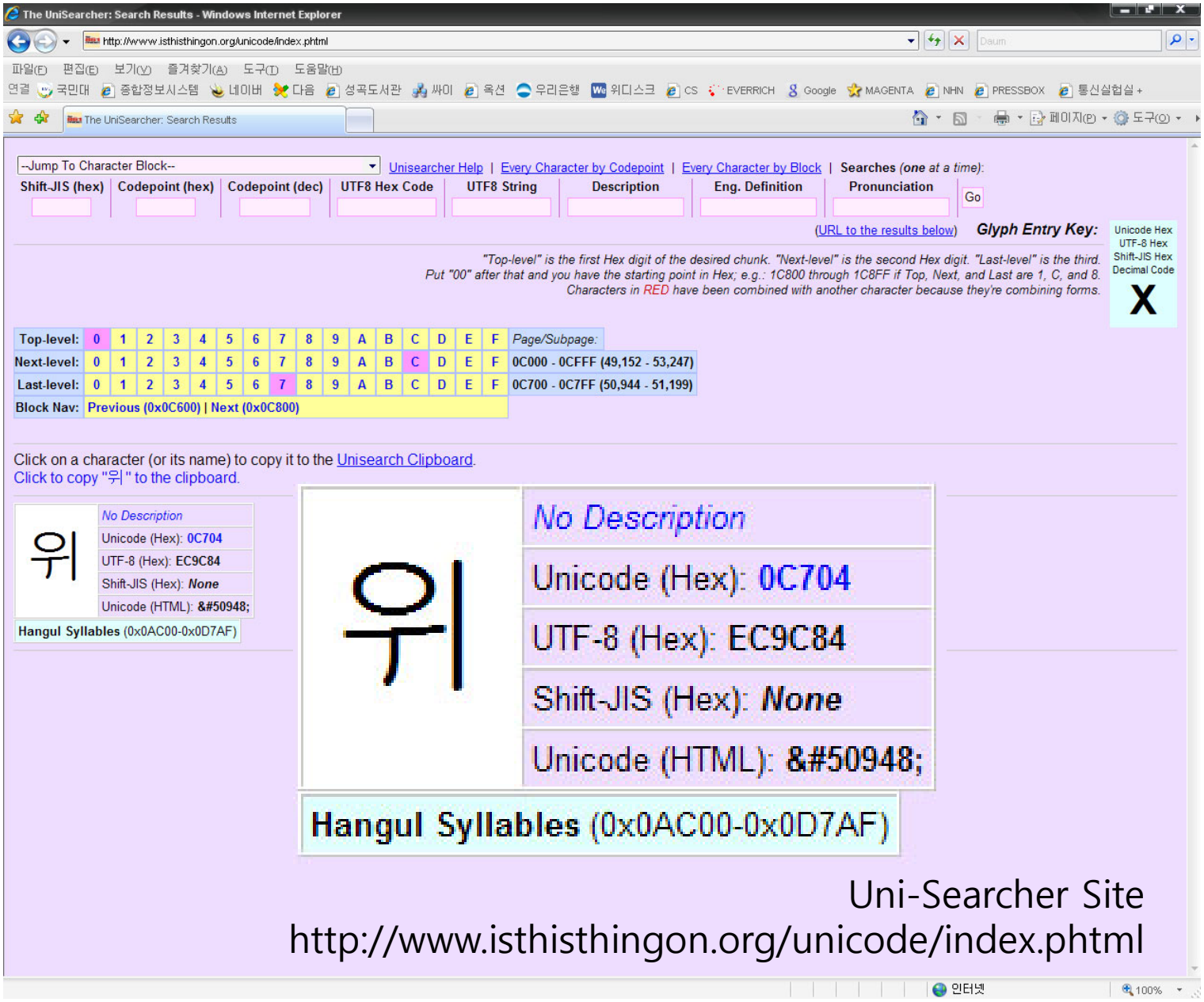
	C70	C71	C72	C73	C74	C75	C76	C77	C78	C79	C7A
0	웩 C700	윍 C710	유 C720	윰 C730	은 C740	읐 C750	읠 C760	읙 C770	읡 C780	자 C790	잠 C7A0
1	웯 C701	윑 C711	육 C721	윱 C731	읒 C741	응 C751	읡 C761	읱 C771	읥 C781	작 C791	잡 C7A1
2	웹 C702	윒 C712	윢 C722	윲 C732	읓 C742	읒 C752	읣 C762	읲 C772	읦 C782	잒 C792	잢 C7A2
3	웽 C703	윓 C713	윣 C723	윳 C733	은 C743	읔 C753	읤 C763	읳 C773	읧 C783	잓 C793	잣 C7A3
4	위 C704	윔 C714	윤 C724	윴 C734	을 C744	읔 C754	읥 C764	이 C774	임 C784	잔 C794	잤 C7A4

- 2진수로 표현

‘위’: 0xC704 =



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



Uni-Searcher Site

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

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	yyyyyyyy xxxxxxxx
UTF-16 BE (Big Edian)	yyyyyyyy xxxxxxxx
UTF-16 LE (Little Edian)	xxxxxxxx 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

- CodeValue = (HighSurrogate - 0xD800) * (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 =

    (HighSurrogate - 0xD800)*(LowSurrogate - 0xDC00)
    + 0x10000 (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