



코드 변환, 인코딩 기법, 빈도조사 프로그램

국민대학교 소프트웨어학부

강 승 식

완성형-조합형 코드 변환

<완성형, 조합형> 코드변환표

```
struct hancode {  
    unsigned short wan;  
    unsigned short joh;  
} hcode[2350] = {  
    0xB0A1, 0x8861, // 가  
    0xB0A2, 0x8862, // 각  
    0xB0A3, 0x8865, // 간  
    .....  
    0xC8FD, 0xD3B5, // 헛  
    0xC8FE, 0xD3B7 // 흥  
};
```

<완성형, 유니코드> 코드변환표

```
struct hancode {  
    unsigned short wan;  
    unsigned short uni;  
} hcode[2350] = {  
    0xB0A1, 0xAC00, // 가  
    0xB0A2, 0xAC01, // 각  
    0xB0A3, 0xAC04, // 간  
    .....  
    0xC8FD, 0xD79B, // 헛  
    0xC8FE, 0xD7A3 // 흥  
};
```

완성형-조합형 코드 변환

// 완성형 → 유니코드 변환표

unsigned short toUNI[25][94] = {

0xB0A1, 0xAC00, // 가

0xB0A2, 0xAC01, // 각

0xB0A3, 0xAC04, // 간

.....

0xC8FD, 0xD79B, // 헛

0xC8FE, 0xD7A3 // 힙

};

// 유니코드 → 완성형 변환표

unsigned short toKSC5601[] = {

0xB0A1, // 가

0xB0A2, // 각

0xB0A3, // 간

.....

0xC8FD, // 헛

0xC8FE, // 힙

};

KS 완성형 한자 → 한글 변환

```
struct hanjatype {
    unsigned short hanja;
    unsigned short kswan;
} hcode[4888] = {
    0xC8A1, 0xB0A1, /* 伽 */
    0xC8A2, 0xB0A1, /* 佳 */
    0xC8A3, 0xB0A1, /* 假 */
    .....
    0xFDFC, 0xC8F1, /* 稀 */
    0xFDFD, 0xC8F1, /* 羲 */
    0xFDFE, 0xC8FA /* 詰 */
};
```

```
// 효율적인 코드 변환표
unsigned short toHangul[52][94] = {
    0xB0A1, /* 伽 */
    0xB0A1, /* 佳 */
    0xB0A1, /* 假 */
    .....
    0xC8F1, /* 稀 */
    0xC8F1, /* 羲 */
    0xC8FA /* 詰 */
};
```

한글 인코딩 방법

- uuencode, uudecode
 - 8비트 3문자 → 6비트 4문자
 - 6비트 문자
 - 32(0x20) 를 더해 줌
 - 프린트 가능한 문자로 변환
- Base64
 - 8비트 3문자 → 6비트 4문자
 - 0x00~0x19 → 'A'~'Z'
 - 0x1A~0x33 → 'a'~'z'
 - 0x34~0x3D → '0'~'9'
 - 0x3E → '+'
 - 0x3F → '/'
 - 예: '안녕하세요' → vsiz58fPvLy/5A==

한글 인코딩 방법

- QP 변환

- 한글 코드값은 그대로 표현
- 예: '가' → =B0=A1
- 예: '안녕하세요' →
=BE=C8=B3=E7=C7=CF=BC=BC=BF=E4

- ISO 2022-KR

- 한글시작 표시 <ESC>\$)C
- <SO>, <SI> 사이에 한글
- MSB를 0으로 setting
- 예: “안녕하십니까?”
<ESC>\$)C
<SO>>H3gGO=J4O1n<SI>?

UTF-8 : unicode 인코딩

- 영문자 : 1 바이트 ASCII 코드
 - 2바이트 문자 : 3~4 바이트로 변환
- 인코딩 방식
 - 0000 ~ 007F → 0xxxxxxx
 - 0080 ~ 07FF → 110xxxxx 10xxxxxx
 - 0800 ~ FFFF → 1110xxxx 10xxxxxx 10xxxxxx
 - 010000 ~ 10FFFF
 - 11110zzz 10zzxxxx 10xxxxxx 10xxxxxx
- 참고. UTF-7 인코딩
 - null 문자(0x0000)를 2byte로 인코딩
 - C 언어의 null문자와의 충돌 문제 해결

Unicode Encoding Forms

Figure 2-11. Unicode Encoding Forms

| | | | | |
|----------|----------|----------|-------------|--------|
| A | Ω | 語 | Ⅲ | UTF-32 |
| 00000041 | 000003A9 | 00008A9E | 00010384 | |
| A | Ω | 語 | Ⅲ | UTF-16 |
| 0041 | 03A9 | 8A9E | D800 DF84 | |
| A | Ω | 語 | Ⅲ | UTF-8 |
| 41 | CE A9 | E8 AA 9E | F0 90 8E 84 | |

Figure 2-12. Unicode Encoding Schemes

| | | | | |
|-------------|-------------|-------------|-------------|----------|
| A | Ω | 語 | Ⅲ | UTF-32BE |
| 00 00 00 41 | 00 00 03 A9 | 00 00 8A 9E | 00 01 03 84 | |
| A | Ω | 語 | Ⅲ | UTF-32LE |
| 41 00 00 00 | A9 03 00 00 | 9E 8A 00 00 | 84 03 01 00 | |
| A | Ω | 語 | Ⅲ | UTF-16BE |
| 00 41 | 03 A9 | 8A 9E | D8 00 DF 84 | |
| A | Ω | 語 | Ⅲ | UTF-16LE |
| 41 00 | A9 03 | 9E 8A | 00 D8 84 DF | |
| A | Ω | 語 | Ⅲ | UTF-8 |
| 41 | CE A9 | E8 AA 9E | F0 90 8E 84 | |

프로그램 연습 : 영어 대문자 출력

```
#include <stdio.h>

void main()
{
    char ch;

    printf("문자 : 10진수 : 16진수\n");
    for (ch = 'A'; ch <= 'Z'; ch++)
        printf("%c : %d : %x\n", ch, ch, ch);
}
```

KS 완성형 한글 2,350자 출력

```
#include <stdio.h>

void main()
{
    unsigned char c1, c2;

    for (c1 = 0xB0; c1 <= 0xC8; c1++)
        for (c2 = 0xA1; c2 <= 0xFE; c2++)
            printf("%c%c : %x\n", c1, c2, (c1<<8 | c2));
}
```

상용조합형 음절 11,172자 출력

```
unsigned i, j, k, hbyte, lbyte;

for (i = 2; i < 21; i++) {           // 초성
    for (j = 3; j < 30; j++) {       // 중성
        if (j == 8 || j == 9 || j == 16 ||
            j == 17 || j == 24 || j == 25)
            continue; /* 중성이 정의되지 않은 것 */
        for (k = 1; k < 30; k++) {   // 종성
            if (k == 18)
                continue; /* 종성이 정의되지 않은 것 */
            hbyte = 0x80 | (i << 2) & 0x7C | (j >> 3) & 0x03;
            lbyte = (j << 5) & 0xE0 | (k & 0x1F);
            printf("%c%c", hbyte, lbyte);
        } putchar('\n');
    } putchar('\n');
}
```

영문자 빈도 조사 프로그램

```
int abc[26] = { 0 }, ABC[26] = { 0 };
int ch, i;

while ((ch=getchar()) != '\n' {
    if (ch >= 'A' && ch <= 'Z')
        ABC[ch-'A']++; /* 대문자 */
    else if (ch >= 'a' && ch <= 'z')
        abc[ch-'a']++; /* 소문자 */
    else
        ; /* 대-소문자 이외의 문자 */
}

for (i =0; i < 26; i++)
    printf("%c : %d, %c : %d\n", 'A'+i, ABC[i], 'a'+i, abc[i]);
}
```

한글 빈도 조사 프로그램

```
int han[25][94] = { 0 };
int i, j, c1, c2;

while ((c1=getchar()) != '\n') {
    if (c1 >= 0xB0 && c1 <= 0xC8) {      /* 한글 검사 */
        c2 = getchar();
        if (c2 >= 0xA1 && c2 <= 0xFE)
            han[c1-0xB0][c2-0xA1]++;
    }
}

for (i = 0; i < 25; i++)
    for (j = 0; j < 94; j++)
        if (han[i][j])
            printf("%c%c : %d\n", 0xB0+i, 0xA1+j, han[i][j]);
}
```