Karakterkódolás
(jegyzet)

Bérci Norbert

2015. november 12-i óra anyaga

Tartalomjegyzék

1. Karakterek és kódolásuk

1.1. Karakterek és karakterkészletek

A *karakter*t [character] (a számhoz hasonlóan) fogalomnak tekintjük, amit meg kell tudni jeleníteni írott formában, illetve el kell tudni tárolni a memóriában vagy bármely más tárolóeszközön, fájlban, illetve továbbítani kell tudni egy informatikai hálózaton. Karakter lehet az ABC egy betűje, egy szám, egy írásjel (a szóközt is beleértve) vagy egyéb más írásrendszerben használt jel illetve vezérlő karakter (például soremelés) is.

Karakterkészlet [character set, charset] alatt karakterek kiválasztott csoportját értjük (a kiválasztás lehet tetszőleges, de általában valamely ország, nemzetiség, nyelv, régió alapján történik).

1.2. Karakterek kódolása

Karakterek kódolása [character encoding, coded character set] alatt a karakterekhez valamilyen érték (kód) rendelését értjük. Ilyen például a Morze-kód, ami karakterekhez hosszabb és rövidebb impulzusokból álló kódokat rendel (amiket aztán könnyen lehet továbbítani például egy rádió adó-vevő segtségével), vagy a Braille-kód, ami karakterekhez 3D objektumokat rendel (amit aztán megfelelő technológiával „kinyomtatva” látásukban sérült emberek is képesek elolvasni). Az informatikában a karakterkódolás általában a karakterhez egy szám rendelését jelenti (amit aztán valamilyen módszerrel tárolunk). Pl: a 65 jelentse az „A” betűt.

Karakterek tárolási formáján [characer encoding form, character encoding scheme] a karakter kódok (számok) konkrét tárolási módját értjük. Ez lehet triviális, például hogy egy megfelelő hosszúságú, egészek tárolására használt ábrázolást alkalmazunk, vagy lehet szofisztikáltabb, például egy tömörebb – de bonyolultabb – változó kódhosszúságú kódolás esetében. Pl. a 65-öt egy bájtos előjel nélküli egészként ábrázolva a 01000001 jelenti az „A” betűt.

Egyes kódolási módszerek egyszerre meghatározzák a karakterek kódolását és a kódok tárolási formáját. Tipikusan ilyen kódolási módszerek a klasszikus kódtáblák [code page, character map, charmap], amelyek előjel nélküli egészként, egy bájton ábrázolnak egy karaktert.

1.3. Klasszikus kódtáblák

1.3.1. Az ASCII kódtábla

Az American Standard Code for Information Interchange (ASCII) 7-bites kódtábla látható az 1. ábrán. Az egyes karakterek kódja a karakter sorának és oszlopának fejlécéből adódik: például a @ kódja 0x40, a W kódja 0x57.

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

1. ábra. Az ASCII 7-bites kódtábla (forrás: wikipedia.org/wiki/ASCII)

Nagyon fontos kiemelni, hogy az ASCII kódtábla 7-bites, azaz 128 karakter kódolására alkalmas. Ha 8-biten tároljuk vagy továbbítjuk, a legnagyobb helyiértékű bitet nullára állítjuk.

1.3.1. feladat. A home1.paulschou.net/tools/xlate/ honlapon ellenőrizhetők az ASCII karakterek bináris átváltásai.

1.3.2. Az ISO 8859-X kódtáblák

Az ISO/IEC 8859-X kódtáblák az ISO és az IEC szabványosító testületek közös kódtáblái, céljuk, hogy minél több regionális karaktert tartalmazzanak. Az ASCII kódtábla az ISO 8859-X kódtáblák része, azaz minden ASCII kód egyben érvényes ISO 8859-X kód is. Mi a leggyakrabban az ISO 8859-1 (nyugat-európai) és az ISO 8859-2 (közép-európai) kódtáblákkal találkozhatunk, ezeket szokás latin-1 és latin-2 kódtábláknak is nevezni. Az ISO 8859-1 kódtáblát ISO 8859-15 (latin-9) néven frissítették (többek között belekerült az euro karakter), és hasonló történt az ISO 8859-2-vel is: ISO 8859-16 (latin-10) néven frissítették.

1.3.3. A Windows kódtáblák

Az ASCII kiterjesztésével a Microsoft megalkotta saját 8-bites kódtábláit, külön-külön egyes régiókra. Mi a leggyakrabban a windows-1250 (közép-európai) és a windows-1252 (nyugat-európai) kódkészletekkel találkozhatunk. Ezeket szokás windows latin-2 illetve windows latin-1 kódtábláknak is nevezni.

Szintén fontos tulajdonsága ezeknek a kódoknak is, hogy az ASCII-t módosítás nélkül tartalmazzák, azaz annak csak kiegészítései. Az ISO 8859-X kódtáblák és a windows-xxxx kódtáblák nagy részben egyeznek (nem kizárólag a közös ASCII részek), de nem teljes mértékben kompatibilisek egymással.

1.3.4. Feladatok

1.3.2. feladat. Megváltozik-e egy adott ASCII karaktert tároló bájt, ha előjeles vagy előjel nélküli egészként tároljuk?

1.3.3. feladat. Keressük meg az ISO 8859-1, az ISO 8859-2, a windows-1250 és a windows-1252 kódtáblák kiosztásait!

1.3.4. feladat. Helyes eredményt kapok, ha egy ASCII kódolt karakterekből álló fájlt a.) windows-1250 b.) windows-1252 kódtáblák alapján próbálom értelmezni?

1.3.5. feladat. Mi lehet a magyarázata annak, hogy régebben (sajnos sokszor még ma is) az ő illetve Ő betűk helyett (hibásan) õ vagy Õ szerepelt?

1.3.6. feladat. Tudunk-e több, különböző kódkészlethez tartozó karaktert egyetlen szövegfájlban tárolni (és azokat helyesen megjeleníteni olvasáskor)?

1.3.7. feladat. Adjunk példát olyan szövegfájlra, ami nem csak ASCII karaktereket tartalmaz, de mégis azonos módon értelmezhető a.) ISO 8859-2 és windows-1250 b.) ISO 8859-1 és windows-1252 kódtáblákkal!

1.4. A Unicode

Az előzőekben ismertetett kódtáblák közös problémája, hogy önmagukban nem képesek több nyelvű szövegek tárolására (sőt, egyes esetekben még egyetlen nyelv esetében sem, például: kínai, japán), mivel a – 8 bites tárolásból adódó – lehetséges 256 különböző karakter nyilván nem elegendő a világ összes nyelvében használt betű és írásjel ábrázolására. Ennek a problémának a megoldására jött létre a Unicode konzorcium, ami létrehozta és karbantartja a Unicode ajánlást. A konzorcium 2014 után minden évben júniusban kiad egy fő verziószámú frissítést (2014 - 7.0, 2015 - 8.0, stb.). A szabvány közel száz írásrendszert és több, mint százezer karaktert tartalmaz, amik között élő és holt nyelvek mellett megtalálhatók matematikai és egyéb szimbólumok is.

1.4.1. feladat. A www.unicode.org/charts/ oldalon nézzük meg a Unicode által támogatott karaktereket!

A Unicode azonos az ISO/IEC 10646 szabvánnyal. A Unicode egy karakter kódolási szabvány (figyelem, önmagában nem kódtábla!), ami meghatározza, hogy egy konkrét karakternek mennyi a Unicode értéke [code point]. Általában U+hexa_kód formában jelöljük: például az euro jel kódja: U+20AC. Ezt a Unicode értéket különböző módokon lehet tárolni.

1.4.1. Az UTF-8

Az UTF-8 a Unicode egy tárolási formája (Unicode Transformation Format) ami a Unicode kódokat változó hosszon, 1-6 bájton tárolja, a kód értékétől függően. Mivel a Unicode-ban jelenleg nincsen `0x10FFFF`-nél nagyobb code point, ezért nincs 4 bájtnál hosszabb UTF-8 kód a gyakorlatban. Az UTF-8 legfontosabb tulajdonságai:

- ASCII kompatibilis, azaz minden ASCII szöveg egyben helyes UTF-8 szöveg is,
- önszinkronizáló, azaz nem kell az UTF-8 bájtsorozat elejéről kezdeni az olvasást, hogy pontosan el lehessen határolni az egyes karaktereket reprezentáló UTF-8 byte csoportokat.

1.4.2. Az UTF-16

Az UTF-16 tárolási forma változó hosszon, 2-4 bájton tárolja a Unicode kódokat. Nem ASCII kompatibilis.

1.4.3. Az UTF-32

Az UTF-32 tárolási forma fix hosszon, 4 bájton tárolja a Unicode kódokat. A kódolás nagyon egyszerű: az Unicode kódokat kell 4 bájtos egészekként tárolni. Nem ASCII kompatibilis.

1.4.4. Feladatok

1.4.2. feladat. Hogyan befolyásolják a tárolási méretet a használt karakterek? Mikor érdemes az UTF-8 és mikor az UTF-16 kódolási formát választanunk?

1.4.3. feladat. Az UTF-8 önszinkronizáló tulajdonságának milyen szerepe van a

- véletlenül kiválasztott pozícióból történő megjelenítésre,
- a hibás átvitelből adódó értelmezési problémákra?

1.4.4. feladat. Az UTF-8, UTF-16, UTF-32 kódolások közül melyek alkalmasak szövegek véletlenszerű elérésére (azaz például ha az x. karakterhez akarok közvetlenül ugrani)?

1.5. Szövegfájlok

Ha egy fájlban csak szöveget akarunk tárolni (pontosabban csak olyan karaktereket, amelyek mindegyike megtalálható egy kiválasztott karakterkészletben), akkor nincs más dolgunk, mint a karakterek (kódtáblája vagy valamely Unicode tárolási formája szerinti) bájtjait egymás után írni, és ezt eltárolni. Valójában is ez történik, ezeket a fájlokat nevezzük *egyszerű szövegfájl*oknak [plain text file], kiterjesztésük (általában): TXT.

1.6. Feladatok

1.6.1. feladat. Hasonlítsuk össze a 7 tárolási formáit: a.) előjel nélküli egészként, b.) kettes komplemens ábrázolású előjeles egészként, c.) ASCII kódolással d.) UTF-8 kódolással!

1.6.2. feladat. Honnét tudjuk, hogy egy szövegfájl beolvasásakor a kódokat melyik kódtábla szerint kell értelmeznünk?

1.6.3. feladat. Pusztán a szövegfájlba történő beleolvasással eldönthető-e általános esetben, hogy az milyen kódtábla szerint értelmezendő?

1.6.4. feladat. Töltsünk be a böngészőnkbe egy szövegfájlt, majd módosítsuk a kódtáblát! Kódoljuk át a szövegfájlt a iconv parancs segítségével, és nézzük meg az eredményt a böngészővel illetve az od programmal!

1.6.5. feladat. Töltsük be a www.itk.ppke.hu/oktatas oldalt, és a böngészőnkben állítsuk át a karakterkészletet! Mi történik?

1.6.6. feladat. Mi a mojibake? (Keressünk rá!)

1.6.7. feladat. Ha ékezetes karaktereket használunk egy SMS-ben, akkor van-e különbség az egy SMS-ben elküldhető karakterek száma között, ha magyar (jobbra dőlő) ékezetekkel rendelkező karaktereket vagy csak balra dőlő ékezetekkel rendelkező karaktereket használunk? Indokoljuk meg!