

Лабораторна робота № 2

Тема: Формати представлення тексту та десяткових чисел в ЕОМ.

Мета: Здобути навички кодування тексту та десяткових чисел в ЕОМ.

Короткі теоретичні відомості

1. Представлення тексту в ЕОМ.

Букви і знаки в пам'яті ЕОМ представляються за допомогою кодування. Кожному символу присвоюється однобайтовий двійковий (шістнадцятковий) код. Для однозначності представлення символів застосовують стандартизацію. Одним із таких стандартів є ASCII (American Standard Code for Information Interchange). Спочатку він був семибітним і представляв 128 символів (тепер – перша половина таблиці). Потім код став восьмибітним і з'явилась можливість представлення 256 різних символів. Перша половина таблиці завжди однакова. Код кожного символу складається з номера стовпчика і номера рядка. Прийнято записувати коди в шістнадцятковому вигляді, хоча можна представляти і в десяткових чи двійкових еквівалентах, наприклад код англійської великої літери "A" дорівнює:

["A"] = 41h = 01000001b = 65d

Код цифри "9" дорівнює:

["9"] = 39h = 00111001b = 57d

Перша половина таблиці має вигляд:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		☺	☹	♥	♦	♣	♠	●	◼	○	◻	♂	♀	♪	♫	⚙
1	▶	◀	↑	!!	¶	§	_	↕	↑	↓	→	←	↔	▲	▼	
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	{		}	~	␣

Не всі коди таблиці представляють друковані символи. Частина кодів є керуючими. Так, наприклад код 07h означає подати звуковий сигнал (BEL).

Код	Назва	Призначення
07h	BEL	Дзвінок
0Ah	LF	Переведення рядка
0Dh	CR	Повернення каретки
7Fh	BackSpace	Забій символа

Друга половина таблиці була використана фірмою IBM для представлення символів західноєвропейських національних мов, символів псевдографіки та деяких математичних символів. Ця таблиця не мала підтримки слов'янських мов і питання в цих країнах вирішувалось самостійно (основне кодування ГОСТ, модифіковане альтернативне кодування ГОСТ, болгарське кодування та ін.). З 1994 року (MS-DOS 6.22) фірма Microsoft упорядкувала це питання, створивши кодову таблицю 866.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
9	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
A	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
B	␣	␤	␥	␦	␧	␨	␩	␪	␫	␬	␭	␮	␯	␰	␱	␲
C	␳	␴	␵	␶	␷	␸	␹	␺	␻	␼	␽	␿	␾	␿	␾	␿
D	␼	␽	␿	␾	␿	␾	␿	␾	␿	␾	␿	␾	␿	␾	␿	␾
E	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
F	≡	±	≥	≤			÷	≈	°	.	.	√	π	²	■	

Для підтримки української та білоруської мов символи з кодами F0h-FFh (240-255) набули вигляду:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
F	Ё	ё	Є	є	Ї	ї	Ў	ў	°	•	√	№	■	■	■	■

Із запровадженням графічних операційних систем типу Windows відпала необхідність у символах псевдографіки і таблиця кодування знову змінилась. Для країн, які використовують кирилицю, таблиця кодування має номер 1251. Друга половина таблиці має вигляд:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Ъ	Ѓ	,	Ѓ	„	...	†	‡	■	%	Љ	<	Њ	Ќ	Ѕ	Ї
9	ђ	`	´	˘	˙	•	—	—	■	™	љ	>	њ	ќ	ћ	џ
A		Ў	ў	Ј	Ѡ	Ѓ	!	§	Ё	©	Є	«	¬	—	®	Ї
B	°	±	І	і	Г	μ	¶	•	ё	№	є	»	ј	ѕ	ѕ	ї
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

Саме тому кодування символів в середовищі MS-DOS і Windows не співпадає.

З метою уніфікації представлення символів усіх писемностей світу та спеціальних знаків в 1991 році Консорціум Юнікоду (англ. Unicode Consortium) представив промисловий стандарт Юнікод, (англ. Unicode), який зараз став дуже поширеним. Його підтримують сучасні операційні системи, прикладні програми, мови програмування. Навіть найпростіший вбудований текстовий редактор Блокнот підтримує кодування Юнікод.

Коди в стандарті Unicode поділені на декілька областей. Область з кодами від U+0000 до U+007F містить символи набору ASCII. Далі розміщені області знаків різних писемностей, знаки пунктуації і технічні символи. Частина кодів зарезервована для використання в майбутньому. Для символів кирилиці виділені коди від U+0400 до U+052F.

Юнікод має декілька реалізацій, але найпоширенішими є дві: UCS (Universal Character Set) та UTF (Unicode Transformation Format) — Формат Перетворення Юнікоду. Реалізація UTF-8 є системою кодування зі змінною довжиною кодування символів. Це означає, що для кодування символів він використовує від 1 до 4 байт на символ. Коди з таблиці ASCII пред-

ставлені в UTF-8 одним байтом, для символів інших мов використовують два або більше байтів. Частина таблиці з кодами 0400–047F показана на рисунку.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0400	È	É	Ђ	Ѓ	Є	Ѕ	І	Ї	Ј	Љ	Њ	Ћ	Ќ	Й	Ў	Ц
0410	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
0420	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
0430	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
0440	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
0450	è	é	ђ	ѓ	є	ѕ	і	ї	ј	љ	њ	ћ	ќ	й	ў	ц
0460	ω	ω	ѳ	ѳ	Ю	ю	А	А	Љ	Љ	Ж	Ж	Љ	Љ	Љ	Љ
0470	Ψ	ψ	Θ	θ	Υ	υ	Ϊ	ϋ	Ου	ου	Ο	ο	Ω	ω	Ω	ω

Текстові файли, представлені в Юнікод, починаються із послідовності байтів FE FF, які отримали назву маркера послідовності байтів BOM (byte order mark). Файли у UTF-8 мають маркер послідовності байтів EF BB BF.

2. Представлення десяткових чисел в ЕОМ.

Десяткові числа можуть бути представлені в ЕОМ в кількох форматах:

- десяткові неупаковані числа;
- десяткові упаковані числа (BCD);
- десятковий ASCII формат;

Десяткові числа можуть бути представлені в ЕОМ наступними форматами:

- **неупакований BCD формат:** на кожен цифру десяткового числа відводиться 1 байт. Значення кожної десяткової цифри записується в молодшу тетраду. Старша тетрада найстаршого байту позначає знак.

Приклад. [2368]=00000010 00000011 00000110 00001000.

- **упакований BCD формат:** В кожному байт записують дві десяткові цифри. Число складається з 20 цифр і займає 10 байт. Найстарший біт найстаршого байта позначає знак.

Приклад. [2368] = 00000000 00000000 ... 00100011 01101000.

[-2368] = 10000000 00000000 ... 00100011 01101000.

- **символьний формат:** для представлення десяткового числа використовують ASCII-коди, тобто число складається із символів. Кількість цифр довільна.

Приклад. [2368]=00110010 00110011 00110110 00111000 = 32h 33h 36h 38h

"2" "3" "6" "8"

Слід враховувати, що байти чисел в пам'яті IBM PC розташовуються в зворотному порядку, тобто за молодшими адресами записуються молодші цифри, а за старшими – старші. Якщо передбачаються арифметичні операції над числами в ASCII форматі, то попереднє число в пам'яті слід розташувати в такій послідовності: 38h, 36h, 33h, 32h.

Порядок виконання роботи

1. Записати текст "Hello, friend!" в ASCII-кодах (формат – 14 байт). Перевірити результат за допомогою програми data2.exe, запущеної під керуванням програми TurboDebugger. Для цього на диску D: у своєму каталозі створити папку TASM і скопіювати в неї файли td.exe та data2.exe, запустити програму td та відкрити файл data2.exe, (або виконати команду td data2 в командному режимі). Виконати за допомогою F8 п'ять команд програми, ввести символи "Hello, friend!", натиснути клавішу Enter, відкрити вікно даних командою

View→Dump, знайти текстовий рядок, починаючи з адреси ds:001Ch.
Не закривати вікно даних програми.

2. Записати число $(N \cdot 100 + N + 32)$ в неупакованому BCD форматі (формат числа – 6 байтів). Для цього ознайомитися із записом числа –326 в сегменті даних за адресою ds:0000h.
3. Записати число $(N \cdot 100 + N + 32)$ в упакованому BCD форматі (формат числа – 10 байт), ознайомившись із записом числа 12345 в сегменті даних за адресою ds:0006h.
4. Записати число $(N \cdot 100 + N + 32)$ в символному ASCII форматі (формат числа – 4 байти). Порівняти з відповідним записом числа –12345 в сегменті даних за адресою ds:0010h.
5. Закінчити роботу програми наступними двома командами F8.