УДК 621.391. 037

В.Я. Певнєв, к.т.н., доц.
І.Л. Яценко, інж.
Національний технічний університет "ХПІ"

ПРО ОДИН ЗАСІБ СТИСКУ ТЕКСТОВОЇ ІНФОРМАЦІЇ

У статті розглядається засіб стиску текстової інформації, заснований на створенні статичних словників, збережених у програмах стиску.

Задача про обмін інформації в телекомунікаційних системах стояла і стоїть достатньо гостро. Це зумовлено тим, що з кожним роком збільшується кількість користувачів у мережах, збільшуються об'єми інформації, що передається в системах. Засобів розв'язання цієї задачі досить багато. Це і створення нових алгоритмів маршрутизації, протоколів обміну, апаратних засобів. Підвищення продуктивності інформаційних систем можна досягнути за допомогою збільшення пропускної здатності мережі або зменшення об'ємів файлів, що передаються при збереженні всієї інформації. Зменшення об'ємів можна досягнути за допомогою стиску інформації.

Існує достатньо велика кількість алгоритмів стиску інформації. При всій їхній різноманітності їх можна класифікувати за областю застосування. Алгоритми стиску текстових файлів і малюнків в численному відрізняються. Метою роботи, що пропонується, буде уявлення методу стиску текстової інформації, основаної на надмірності уявлення символів.

Існуючі алгоритми стиску (говорячи про алгоритми стиску, надалі будемо вважати тільки текстову інформацію) умовно можна поділити на два класи. Перший клас – це алгоритми, основані на статистичних засобах стиску. Ідея засобу – символи , що повторюються, потрібно кодувати більш короткими ланцюжками бітів, ніж ланцюжки рідких символів. Найбільш відомим представником даного класу буде засіб Хаффмена [1]. До другого класу відносяться алгоритми, в яких здійснюється кодування послідовності з двох або більше символів, що зустрічалися, новим символом. До даних алгоритмів відносяться алгоритми Лемпеля-Зива [2].

Головною ідеєю всіх перетворень в обох класах засобів стиску буде створення словників. В цих словниках відбувається запам'ятовування символів, що повторюються, або їхніх ланцюжків, що кодуються ланцюжками меншої довжини. Таким чином, при обміні інформацією необхідно передавати не тільки самий текст, але й відповідний словник. У цьому випадку файл, що передається, стає дуже вразливим до різноманітних викривлень. Викривлення навіть одного біта призводить до того, що файл не можна розпакувати на приймальній стороні. У цьому випадку необхідно вдаватися до завадостійкого кодування. Як відомо [1], застосування завадостійкого кодування веде до зростання обсягу повідомлення , що передається. При цьому коди можуть бути з виявленням помилок і з виправленням помилок. Перші в разі появи помилки вимагають повторної передачі повідомлення, другі – за рахунок надмірності відновлюють повідомлення.

Сутність засобу , що пропонується, полягає в перетворенні кодової таблиці, складеної з 256 символів, в декілька таблиць розміром в 64 символи. Як відомо, для відображення будь-якого символа використовується один байт. З його допомогою можна передати латинський алфавіт (малі і великі літери), національний алфавіт (малі і великі літери), цифри, розділові знаки, спеціальні знаки, елементи псевдографіки. Це все відноситься до коду ДКОИ в DOS. Якщо розглянути операційні системи Windows XX, то кожному символу, що відображується на екрані або передається в лінію зв'язку, відповідає два байта.

При використанні методу, що пропонується, нові кодувальні таблиці або, як їх називають, використовуючи термінологію засобів стиску даних, словники містять символи, які можна кодувати за допомогою 6 біт. Дані словники повинні міститися в програмі, що забезпечує стиск-відновлення даних. Цілком очевидно, що в цьому випадку відпадає необхідність їхньої передачі зі стислим файлом. Викривлення одного або декількох біт наведе тільки до втрати одного або декількох символів.

Очевидно, що при створенні таких словників необхідно піти на деякі хитрощі, що дозволять мінімізувати кількість символів, що передаються. Наприклад, для того, щоб передати велику літеру, необхідно передати спеціальний символ і малу літеру. При розкритті файлу

після цього символа буде відновлена відповідна велика літера. Таким чином, можна зменшити словник на кількість літер в алфавіті. Слід відзначити і той факт, що в словниках необхідно передбачити можливість переходу від однієї мови до іншої. Аналогічний підхід здійснений в так званих «гарячих клавішах» на клавіатурі, коли натиск двох клавіш призводить до переключення алфавіту. Приклад таких словників наведений в табл. 1.

Таблиця 1

Приклад словників кодування

Десяткове уявлення	Двоїчне уявлення	Символи алфавіту 1	Символи алфавіту 2	Символи алфавіту 3	Символи алфавіту 4	Символи алфавіту 5
1	2	3	4	5	6	7
0	000000	а	а	a	α	0
1	000001	б	б	b	β	1
2	000010	в	в	c	χ	2
3	000011	г	г	d	δ	3
4	000100	д	д	e	ε	4
5	000101	е	е	f	ϕ	5
6	000110	ж	є	g	γ	6
7	000111	з	ж	h	η	7
8	001000	и	з	i	ι	8
9	001001	й	і	j	φ	9
10	001010	к	і	k	κ	=
11	001011	л	ї	l	λ	+
12	001100	м	й	m	μ	-
13	001101	н	к	n	ν	}
14	001110	о	л	o	ο	{
15	001111	п	м	p	π	]
16	010000	р	н	q	θ	[
17	010001	с	о	r	ρ	‘
18	010010	т	п	s	σ	“
19	010011	у	р	t	τ	<
20	010100	ф	с	u	υ	>
21	010101	х	т	v	ϖ	/
22	010110	ц	у	w	ω	\
23	010111	ч	ф	x	ξ	|
24	011000	ш	х	y	ψ	
25	011001	щ	ц	z	ζ	&
26	011010	ъ	ч			#
27	011011	ы	ш			@
28	011100	ь	щ			$
29	011101	э	ь			%
30	011110	ю	ю			^
31	011111	я	я			№
32	100000		’			~
33	100001	,	,	,	,	,
34	100010	:	:	:	:	:
35	100011	;	;	;	;	;
36	100100	_	_	_	_	_
37	100101	?	?	?	?	?
38	100110	!	!	!	!	!

Закінчення таблиці 1

39	100111	(	(	(	(	(
40	101000	)	)	)	)	)
41	101001	Кінець абзацу	Кінець абзацу	Кінець абзацу	Кінець абзацу	Кінець абзацу
42	101010	Перенос	Перенос	Перенос	Перенос	Перенос
43	101011	Мала	Мала	Мала	Мала	Мала
44	101100	Прописна	Прописна	Прописна	Прописна	Прописна
45	101101	. Пробіл	. Пробіл	. Пробіл	. Пробіл	. Пробіл
46	101110	.	.	.	.	.
47	101111	Всі строчні	Всі строчні	Всі строчні	Всі строчні	Всі строчні
48	110000					≥
49	110001	Всі пропис.	Всі пропис.	Всі пропис.	Всі пропис.	Всі пропис.
50	110010					≠
51	110011		Перехід до алфав. 1	Перехід до алфав. 1	Перехід до алфав. 1	Перехід до алфав. 1
52	110100	Перехід до алфав. 2		Перехід до алфав. 2	Перехід до алфав. 2	Перехід до алфав. 2
53	110101					÷
54	110110					≤
55	110111					↔
56	111000	Перехід до алфав. 3	Перехід до алфав. 3		Перехід до алфав. 3	Перехід до алфав. 3
57	111001					≡
58	111010					⊂
59	111011					⊃
60	111100	Перехід до алфав. 4	Перехід до алфав. 4	Перехід до алфав. 4		Перехід до алфав. 4
61	111101					∃
62	111110	Пробіл	Пробіл	Пробіл	Пробіл	Пробіл
63	111111	Перехід до алфав. 5	Перехід до алфав. 5	Перехід до алфав. 5	Перехід до алфав. 5	⊕

При використанні такого способу кодування знаменита фраза «Мама мыла раму.» буде записана як:

101100 001100 000000 001100 000000 111110 001100 011011
001011 000000 111110 010000 000000 001100 010011 101110.

Дану послідовність можна розглядати як безперервну, і в лінію зв'язку представляється послідовність шістнадцятирічних цифр:

B0 C0 0C 03 E3 1B 2C 0F 90 00 C4 EE.

Таким чином, дане повідомлення можна уявити за допомогою 12 байт.

Як відомо, найбільш розповсюджені архіватори стиску текстової інформації, що використовуються в інформаційних мережах, RAR і ZIP. Розміри файлів до стиску і після з використанням означених архіваторів уявлені в табл. 2. Слідує відзначити і той факт, що використання текстового редактора Word призводить до значного зростання розміру файлу.

Таблиця 2

Розміри файлів залежності від типу архіватора

Номер тексту	Розширення файлу	Відкритий текст	Тип архіватора	
			RAR	ZIP
1	. txt	15	75	129
	. doc	19456	1716	1814
	. htm	277	296	430
	. rtf	2157	812	842
2	. txt	137	188	237
	. doc	19456	1711	1822
	. htm	407	369	410
	. rtf	2675	1002	1048
3	. txt	2725	1103	1134
	. doc	25600	3168	3345
	. htm	3279	1290	1326
	. rtf	6314	2068	2119

Як видно з таблиці, використання архіваторів на файлах, що існують в маленьких розмірах, призводить до збільшення розмірів файлів. Використання файлів з розширенням rtf і doc призводить до значного зростання їхнього розміру. Використання засобу, що пропонується, дозволяє передати текст 1 за допомогою 12 байт, текст 2 – 109 байт, а текст 3 – 2057 байт.

При розгляді трафіка інформаційних систем бачимо, що велика кількість файлів, що передаються в лінії зв'язку, мають малі розміри, що коливаються в межах до 1 кілобайта. Мова в даному випадку йде про корисний розмір файлу, тобто про те повідомлення, що необхідно передати, а не про те, яким чином повідомлення , що передається, буде представлене в тому або іншому текстовому редакторі, в тому або іншому форматі збереження. Як видно з табл. 2, файли малих розмірів стискати за допомогою існуючих архіваторів не має сенсу. Таке перетворення інформації призводить тільки до збільшення розміру файлу , що передається.

У вигляді висновку слід визначити галузь застосування запропонованого засобу стиску текстової інформації. Даний засіб більш прийнятний при архівації файлів малих розмірів за умови достатньо надійного каналу. Якщо у вигляді каналу зв'язку можливо використання телефонної лінії, то ефективність застосування розробленого засобу підвищується за рахунок того, що існуючі архіватори вимагають додаткових заходів щодо кодування інформації. Слід відзначити і той факт, що час перетворення файлу при використанні розглянутого засобу менший, ніж у існуючих.

ЛІТЕРАТУРА:

1. *Хемминг Р.В.* Коди з відкриттям і виправленням помилок / Коди з відкриттям і виправленням помилок. – М.: ІЛ, 1956. – С. 7–22.
2. *Кричевський Р.Є.* Стиск і пошук інформації. – М.: Радіо і зв'язок, 1989. – 168 с.

ПЕВНЄВ Володимир Якович – кандидат технічних наук, доцент кафедри «Системи інформації» Національного технічного університету «Харківський політехнічний інститут».

Наукові інтереси:

– оптимізаційні задачі на графах;

– системи захисту інформації та їх криптоаналіз.

Тел. (057) 400-951.

E-mail: pevnev@kpi.kharkov.ua

ЯЦЕНКО Ірина Леонідівна – інженер кафедри «Системи інформації» Національного технічного університету «Харківський політехнічноий інститут».

Наукові інтереси:

– системи захисту інформації та їх криптоаналіз.

Тел. (057) 400-951.

Подано 15.10.2002