Лабораторная работа № 4

Алгоритм кодирования/декодирования Base64

4.1 Цель работы

Изучение алгоритма кодирования/декодирования base64.

4.2 Задание на лабораторную работу

В лабораторной работе необходимо разработать приложение реализующее алгоритм кодирования/декодирования base64.

4.3 Методические указания

4.3.1 Вступление

Base64 буквально означает — позиционная система счисления с основанием 64. Здесь 64 — это наибольшая степень двойки (2^6), которая может быть представлена с использованием печатных символов ASCII. Эта система широко используется в электронной почте для представления бинарных файлов в тексте письма (транспортное кодирование). Все широко известные варианты, известные под названием Base64, используют символы A-Z, a-z и 0-9, что составляет 62 знака, для остальных двух знаков в разных системах используются различные символы.

Долгое время для кодирования бинарных файлов в 6-битный формат (чтобы обеспечить их пересылку по почтовой системе Internet) использовалась кодировка UUENCODE, имеющая ряд технических ограничений. Стандарт ***MIME***(описание стандарта дано в лабораторной работе №5) предполагает использование более устойчивой кодировки "Base64", которая специально разработана для обеспечения сохранности данных, пересылаемых по email, при различных преобразованиях, имеющих место в ходе прохождения почтовых шлюзов.

4.3.2 Идеология base64

Как известно, байт состоит из восьми бит. В один байт можно вложить 256 цифр, от 0 до 255. Однако, если вместо восьми байт использовать только шесть, то объем вложенной информации уменьшается до 64 цифр, от 0 до 63. Теперь главное: любую цифру 6-ти битового байта можно представить в виде печатного символа. 64 символа это не так много, но для us-ascii символов вполне хватит. Ниже представлен 64-х символьный base64 "алфавит".

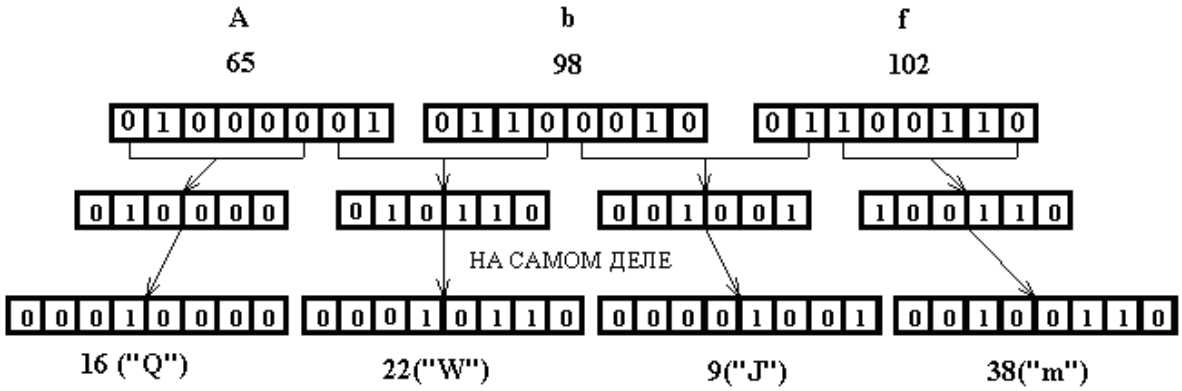
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz 0123456789+/

где код символа A - 0, а код символа / - 63.

А далее берутся три последовательных байта по восемь бит (всего 24 бита), и побитно делятся на четыре 6-ти битных байта (всего 24 бита).

Немного странно звучит: "шестибитный байт". На самом деле бит восемь, однако используются только 6 младших бит, два старших бита игнорируются.

Схематично такое "деление три к четырем" можно представить себе так:



В приведенном примере три символа A, b и f были закодированы в base64 формат. В результате мы получили 4-х символьную строку QWJm. Таким образом, на практике увидели идеологию перевода двоичной информации в текст по принципу 3 к 4.

Основываясь на этом принципе, мы можем закодировать любую двоичную информацию в текст, увеличивая ее объем примерно на 30%.

Примечание: *если у нас нет трех байтов, то в конец четырех символьной строки добавляется символ = (равно). Если не хватает (до трех) одного байта, то добавляется один символ "равно":*

QWJ=

если не хватает двух байт, то добавляются два символа "равно":

QW==

с символами "равно" надо разбираться только один раз - при чтении конца файла.

4.3.3 Алгоритм кодирования/декодирования base64

В качестве примера закодируем и раскодируем строку Abf. Приведенный ниже алгоритм не является единственно верным, это один из множества алгоритмов кодирования/декодирования base64.

а) Кодирование

1. Считываем первые три байта.

(01000001 01100010 01100110)

2. Делаем побитовый сдвиг первого байта вправо на 2 разряда

(00010000 = 16)

3. Ищем символ в алфавите Base64 с этим числом+1(т.к. А имеет 0 индекс), это первый символ закодированной последовательности. Запоминаем его.

(”Q”)

4. Делаем побитовый сдвиг первого байта на 4 влево, а второго байта на 4 вправо

(00010000 00000110)

5. Применяем бинарную операцию ИЛИ

```
    00010000
Или 00000110
    00010110
```

6. К полученному результату применяем операцию И с 111111, чтобы избавиться от двух старших разрядов.

```
    00010110
И   00111111
    00010110
```

00010110 = 22

7. Ищем символ в алфавите Base64 с этим числом+1(т.к. А имеет 0 индекс), это второй символ закодированной последовательности. Запоминаем его.

(”W”)

8. Делаем побитовый сдвиг второго байта на 2 влево, а третьего байта на 6 вправо

(10001000 00000001)

9. Применяем бинарную операцию ИЛИ

```
    10001000
Или 00000001
    10001001
```

10. К полученному результату операцию И с 111111, чтобы избавиться от двух старших разряда.

```
  10001001
И 00111111
  00001001
```

00001001 = 9

11. Ищем символ в алфавите Base64 с этим числом+1(т.к. А имеет 0 индекс), это третий символ закодированной последовательности. Запоминаем его.

(”J”)

12. К третьему байту применяем операции И с 111111

```
  01100110
И 00111111
  00100110
```

13. Ищем символ в алфавите Base64 с этим числом+1(т.к. А имеет 0 индекс), это четвертый символ закодированной последовательности. Запоминаем его.

(”m”)

14. Считываем следующие три байта и так далее.

Необходимо также следить за признаком конца файла и, в случае необходимости, добавлять “=”.

б) Декодирование

1. Считываем первые 4 символа

(QWJm)

2. Определяем индексы этих символов по алфавиту base64

(16 22 9 38)

3. Делаем побитовый сдвиг первого байта на 2 влево, а второго байта на 4 вправо

(01000000 00000001)

4. Применяем бинарную операцию ИЛИ

```
    01000000
Или 00000001
    01000001
```

01000001 = 65

5. По таблице ASCII 65 = ”**A**”

6. Делаем побитовый сдвиг второго байта на 4 влево, а третьего байта на 2 вправо

(0110000000000010)

7. Применяем бинарную операцию ИЛИ

```
    01100000
Или 00000010
    01100010
```

01100010 = 98

8. По таблице ASCII 98 = ”**b**”

9. Делаем побитовый сдвиг третьего байта на 6 влево и применяем бинарную операцию ИЛИ с 4 байтом.

```
    01000000
Или 00100110
    01100110
```

01100110 = 102

10. По таблице ASCII 102 = ”**f**”

4.4 Приложение. Таблица ASCII кодов

Таблица 4.1

Код	Символ	Обозначение	Код	Символ	Код	Символ	Код	Символ
0	(null)	nul	32	Пробел	64	@	96	`
1	☺	soh	33	!	65	A	97	a
2	☻	stx	34	“	66	B	98	b
3	♥	etx	35	#	67	C	99	c
4	♦	eot	36	$	68	D	100	d
5	♣	enq	37	%	69	E	101	e
6	♠	ack	38	&	70	F	102	f
7	•	bel	39	'	71	G	103	g
8	◘	bs	40	(	72	H	104	h
9	○	ht	41	)	73	I	105	i
10	◙	lf	42	*	74	J	106	j
11	♂	vt	43	+	75	K	107	k
12	♀	ff	44	,	76	L	108	l
13	♪	cr	45	-	77	M	109	m
14	♫	so	46	.	78	N	110	n
15	☼	si	47	/	79	O	111	o
16	►	dle	48	0	80	P	112	p
17	◄	dc1	49	1	81	Q	113	q
18	↕	dc2	50	2	82	R	114	r
19	‼	dc3	51	3	83	S	115	s
20	¶	dc4	52	4	84	T	116	t
21	§	nak	53	5	85	U	117	u
22	▬	syn	54	6	86	V	118	v
23	↨	etb	55	7	87	W	119	w
24	↑	can	56	8	88	X	120	x
25	↓	em	57	9	89	Y	121	y
26	→	sub	58	:	90	Z	122	z
27	←	esc	59	;	91	[	123	{
28	∟	fs	60	<	92	\	124	|
29	↔	gs	61	=	93	]	125	}
30	▲	rs	62	>	94	^	126	~
31	▼	us	63	?	95	_	127	del

4.5 Контрольные вопросы и задания

1. Закодировать в base64 файл в котором находиться текст Xz25
2. Закодировать в base64 файл в котором находиться текст Fh7k
3. Закодировать в base64 файл в котором находиться текст gRn4
4. В файле, закодированном в base64 находиться текст Pt54. Раскодировать файл.
5. В файле, закодированном в base64 находиться текст Rj0v. Раскодировать файл.
6. В файле, закодированном в base64 находиться текст zbTM. Раскодировать файл.