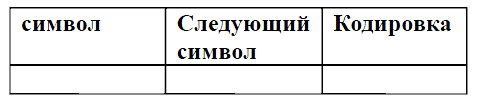
**ЛЕКЦИЯ. МЕТОДЫ СЖАТИЯ ИНФОРМАЦИИ**.

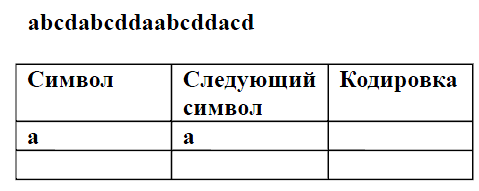
Существует несколько базовых алгоритмов сжатия информации. Основная идея этих алгоритмов – заменять часто встречающиеся комбинации более короткими (алгоритм Хаффмана), либо заменять более длинные последовательности более короткими (метод Лемпеля-Зива). Сжатие графических файлов основано на принципах частичной потери информации (JPEG). В операционной системе Windows получил применение алгоритм Лемпеля-Зива. Познакомимся с этим алгоритмом.

Рассмотрим алгоритм Лемпеля-Зива на примере. Пусть дана последовательность: **aabcdabcddaabcddacd**

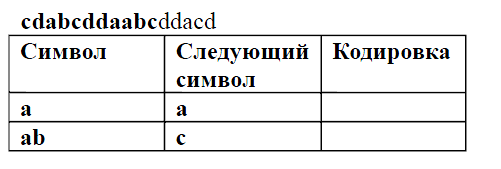
Строим таблицу:

****

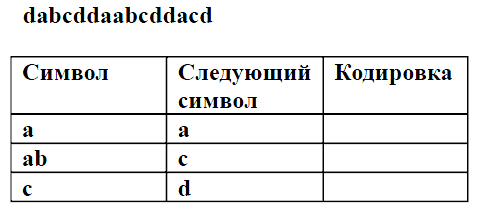
Первый символ – a. Его в таблице (в левом столбце) нет. Добавляем его в таблицу:

****

Следующий символ – а. В таблице он уже есть. За ним следует символ b. Получаем новую комбинацию символов – ab, которой в таблице еще нет. Помещаем ее в таблицу:

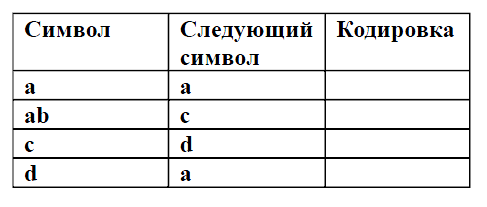
****

Продолжаем с символа c. Такого символа в таблице нет. Помещаем его в таблицу и выписываем следующий символ:

****

**Следующий символ – d. В таблице его нет. Помещаем в таблицу:**

**abcddaabcddacd**

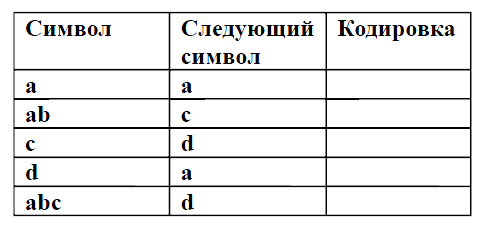
****

Cледующий символ – a. В таблице он есть. За ним идет символ b. Получаем

комбинацию ab. Она также есть в таблице. За ab идет символ с. Получаем

комбинацию abc. Такой комбинации нет. Включаем ее в таблицу и получаем

**abcddaabcddacd**

****

Продолжаем далее по аналогии и получаем следующую результирующую таблицу



В результате входная последовательность **aabcdabcddaabcddacd** преобразуется к такому виду:

0001 0101 0100 0101 0110 0111 1000 1001 1010 1001

Это – двоичная последовательность. Она содержит 40 битов. А какова же

последовательность битов исходной комбинации? Она такая

0001 0001 1000 0011 0100 0001 … и т.д. всего 76 битов. Если кодировать буквы по два бита, то потребуется 38 битов. Однако при увеличении размера исходной

последовательности достигаемый по Лемпелю –Зива эффект будет значительным.

Приведем пример программы для чтения файла, сжатого по методу Лемпеля-Зива:

**ЧТЕНИЕ КОМПРЕССИРОВАННОГО (по Лемпелю-Зиву) ФАЙЛА**

#include "stdafx.h"

#include <windows.h>

#include <stdio.h>

// #pragma comment(lib, "LzExpand")

#pragma comment(lib, "LZ32")

int CopyFile(LPWSTR Source, LPWSTR Destination)

{

INT hsource;

INT hdest;

DWORD iret;

LPOFSTRUCT OpenStruct = {0};

// Allocate buffer

OpenStruct = (LPOFSTRUCT)malloc(sizeof(LPOFSTRUCT));

if(OpenStruct != NULL)

wprintf(L"malloc() - memory allocated!\n");

else

wprintf(L"malloc() - memory allocation failed! error %u\n",

GetLastError());

// Open the source and destination files.

// Opens the file for reading only.

hsource = LZOpenFile(Source, OpenStruct, OF\_READ);

if(hsource)

wprintf(L"LZOpenFile() for reading compressed file looks OK!\n");

else

// Can't use GetLastError() lol

wprintf(L"LZOpenFile() for reading compressed file failed miserably!\n");

// Directs LZOpenFile to create a new file. If the file already exists,

// it is truncated to zero length.

hdest = LZOpenFile(Destination, OpenStruct, OF\_CREATE);

if(hdest)

wprintf(L"LZOpenFile() for creating file looks OK!\n");

else

// Can't use GetLastError() lol

wprintf(L"LZOpenFile() for creating file failed miserabily!\n");

// Copy the source file to the destination location, and

// decompress the Source file if it was compressed.

// Fail - returns less than 1 www.installsetupconfig.com

// Success - returns the size, in bytes, of the destination file

iret = LZCopy(hsource, hdest);

// Close those files. This function does not return a value.

LZClose(hdest);

LZClose(hsource);

if(iret == -1)

wprintf(L"File transfer failed\n");

else

wprintf(L"Copy is successful: %u bytes were transferred.\n");

// Failed to free the allocated buffer lol!

// free(OpenStruct);

// Let see what is returned

return iret;

}

int wmain(int argc, WCHAR \*argv[])

{

LPWSTR pSourceFileName = NULL;

LPWSTR pDestFileName = NULL;

DWORD dwRetVal = 0;

// Just a test. iexplore.chm is a compressed file

pSourceFileName = L"\\\\?\\h:\\ZDD.bkf";

// We uncompress and copy to C:

pDestFileName = L"\\\\?\\C:\\Work\\ZDD";

// Call CopyFile()

dwRetVal = CopyFile(pSourceFileName,pDestFileName);

wprintf(L"CopyFile() returns %u\n", dwRetVal);

getchar();

return 0;

}

Для чтения файла, сжатого по Лемпелю-Зиву, создается буфер в памяти, используемый как рабочая область при выполнении алгоритма

OpenStruct = (LPOFSTRUCT)malloc(sizeof(LPOFSTRUCT));

Выполняется открытие файла:

hdest = LZOpenFile(Destination, OpenStruct, OF\_CREATE);

Декомпрессия выполняется по команде

iret = LZCopy(hsource, hdest);