РЕФЕРАТ

**на тему:**

**“Програми-архіватори WinARJ”**

1. Коротка історія розвитку теорії стиснення інформації

У сорокових роках учені, що працюють в області інформаційних технологій, ясно зрозуміли, що можна розробити такий спосіб збереження даних, при якому простір буде витрачатися більш ощадливо. Клод Шеннон, вивчаючи нюанси розходжень між семантикою (semantics) (що деяка сутність значить) і синтаксисом (syntax) (як деяка сутність виражається), розробив більшість базових понять цієї теорії. Розуміння того, що те саме значення (семантика) може бути реалізовано різними способами (синтаксис), приводить до закономірного питання: "Який спосіб вираження чого-небудь є найбільш економічним?" Пошук відповіді на це питання привів Шеннона до думки про ентропію, що, простіше говорячи, співвідноситься з кількістю, що міститься у файлі корисної інформації. Методи стиску намагаються збільшувати ентропію файлу, тобто зменшувати довжину файлу, зберігаючи при цьому всю інформацію.

Однак, Шеннон не був першим, хто задумувався про сутність інформації і визначенні її кількості. Перший крок на цьому шляху зробив у 1928 р. Хартлі. Основний отриманий їм результат можна сформулювати приблизно так: якщо в заданій безлічі, що містить N елементів, виділений деякий елемент x, про яке відомо лише, що він належить цій безлічі, то, щоб знайти x, необхідно одержати кількість інформації, рівне log2 N. Цю формулу звичайно називають формулою Хартлі.

Формула Хартлі є часткою случаємо більш загальної формули Шенона, що дозволяє знайти кількість інформації у випадковому повідомленні фіксованого алфавіту. Нехай X1, ..., Xn - символи цього алфавіту, P1, ..., Pn - імовірності їхньої появи в тексті повідомлення, тоді формула Шеннона приймає вид:

*H = P1\*log2(1/ P1) + ... + Pn\*log2(1/ Pn),*

де H - кількість біт інформації в одному символі повідомлення, чи ентропія символу повідомлення. Це число показує мінімальне середнє число біт, необхідних для представлення одного символу алфавіту даного повідомлення.

У деяких випадках алфавіт повідомлення може бути невідомий, тоді висуваються гіпотези про алфавіт повідомлення. Маючи різні алфавіти, можна досягти різних коефіцієнтів стиску. Наприклад, текстовий файл, якщо його розглядати як послідовність бітів, має ентропію порядку 0.7 - 0.9, якщо як послідовність байтів, - 0.5 - 0.7, хоча популярні програми стиску зменшують розміри текстових файлів до 0.3 - 0.4 від вихідного розміру.

Доказ Шенона не було конструктивним, тобто не містило способу побудови цих оптимальних кодів, а лише показувало їхнє існування. До появи роботи Шенона, кодування символів алфавіту при передачі повідомлення по каналах зв'язку здійснювалося однаковою кількістю біт, одержуваним по формулі Хартлі. З появою цієї роботи почали з'являтися способи, що кодують символи різним числом біт у залежності від імовірності появи їхній у тексті. Наприклад, часто у файлах деякі значення байта зустрічаються частіше інших. Таким чином, за рахунок використання для кожного значення байта коду різної довжини можна значно зменшити загальний розмір даних. Ця базова ідея лежить в основі алгоритмів стиску Шенона-Фано (Shannon-Fano) і Хафмана (Huffman). Подібні алгоритми вибирають більш короткі коди для часто зустрічаються і більш довгі для рідко зустрічаються значень байта. Звичайно текстові файли (у який одні значення байтів повторюються набагато частіше інших) вони стискають досить добре.

Більш тридцяти років алгоритм стиску Хафмана і його варіанти залишалися найбільш популярними методами. Однак у 1977 два дослідників з Ізраїлю запропонували зовсім інший підхід до цієї проблеми. Абрахам Лемпел і Якоб Зів висунули ідею формування "словника" загальних послідовностей даних. При цьому стиск даних здійснюється за рахунок заміни записів відповідними кодами зі словника.

# 2. Архіватори MS DOS

З розвитком комп'ютера стали збільшуватися й обсяги інформації збереженої в ньому, що у свою чергу привело до розвитку технологій по збереженню цієї інформації в стиснутому виді, тобто в архівах. Для цього була придумана безліч програм здійснюючих архівацію інформації.

Однак у роботі з цією інформацією іноді небажано розкривати повний архів, щоб взяти один чи два необхідних чи файли ж просто подивитися, що в архіві за інформація.

Програми-архіватори, за винятком одиниць, не надають зручних оболонок позволяющих просто, швидко й у наочній формі розібратися з вмістом архівів.

Архіватори -- це програми, що дозволяють створювати й обробляти архівні копії файлів. При цьому з архівні копії мають менший розмір, чим оригінали. За допомогою спеціальних алгоритмів стиску з файлів віддаляється вся надлишкова інформація, а при застосування зворотних алгоритмів розпакування архівна копія відновлюється в первісному виді.

Найбільш відомі программи-архіватори для MS-DOS: ARJ (розроблювач -- Robert K. Jung), pkzip

(компанія PKWARE Inc.), lha (Haruyasu Yoshizaki), zoo (Rahul Dhesi). Безумовним лідером в усьому світі за останні 5 років став архіватор RAR. В даний час RAR активно витісняє ZIP як основну утиліту стиску FTP архівів у мережі INTERNET. RAR я є єдиною всесвітньо використовуваною програмою, створеної російським програмістом (за винятком TETRIS). Всі архіватори відрізняються використовуваними алгоритмами стиску, форматами архівних файлів, швидкістю роботи і т.д.

*Терміни, використовувані в архівації*

Add file Додавання (копировние) файлу в архів. Якщо архів не існує, то він створюється.

CRC Код циклічного контролю. Спеціальна функція від усього умісту файлу. Складається таким чином, що змінити файл так, щоб його CRC залишився незмінним, практично неможливо.

Exclude selected files При архівації НЕ додавати в архів визначені файли.

Extract files Витяг файлів з архіву без збереження структури підкаталогів.

Extract files with pathnames Витяг файлів з архіву зі збереженням структури підкаталогів.

Fresh files Додавання в архів нових версій уже наявних там файлів.

Garble (чи scramble) files with password Архівація файлів з паролем. Витягти файли з такого архіву можна, лише правильно вказавши пароль.

Move files Переміщення файлів в архів.

Multiple volumes Багатотомні архіви -- складаються з декількох файлів (томів). Зручні при архівації великих комплексів файлів, коли тому архіву можна поміщати на окремі дискети.

Ratio Ступінь стиску файлу. Визначається по-різному в різних архіваторах (відношення вихідного до стиснутого або навпаки).

Recurse subdirectories Архівація файлів із заданого каталогу і всіх його підкаталогів. В архіві зберігається інформація про шлях до файлів, і при витягу їх можна виводити не в один каталог, а у відповідні підкаталоги.

Self-extract (sfx)archive архів, Що Саморозпаковується. Архівний файл має розширення .exe .чи com, і після його запуску відбувається автоматичний витяг файлів з архіву.

Test integrity Перевірка цілісності архіву, тобто перевірка CRC файлів архіву.

Update files Додавання в архів нових файлів.

3. Архіватор ARJ

Працює з командного рядка. Виконує усі функції по обслуговуванню архівів.arj , у т.ч. підтримку багатотомних архівів.

Одержати довідку по ключах архіватора arj за допомогою команд:  
 arj(звичайна довідка)

arj /?(докладна довідка)

**Arj** має дуже велике число ключів. Можна автоматизувати багато дій -- створення резервної копії диска, архівування починаючи з якоїсь дати, додавання до імені архіву поточної дати (arh970821.arj), архівування файла з конкретного місця, кілька рівнів стиску і так далі. У версії 2.55 можлива робота з довгими іменами.

*Переваги:* дуже велике колличество ключів, що дає можливість автоматизувати велике число функцій. Захист архіву від ушкоджень.

*Недоліки:* відсутність діалогового режиму, якесь незручності роботи при наявності якогось ключа в перемінної оточення (ARJ\_SW) і рядку запуску -- взаємне знищення.

## ARJZ

ARJZ (з волі автора програми вимовляється як "арж-зет") - це архіватор, заснований на відомій програмі ARJ Роберта Юнга. На відміну від таких сучасних засобів архівирования, як RAR і UC2, ARJZ використовує формат файлів, командний рядок і опції, сумісні з однієї із самих популярних програм стиску даних, а це має свої переваги. Зокрема:  
 1) Практично все програмне забезпечення, розраховане на виклик ARJ, буде працювати так само і з програмою ARJZ без усякої модифікації. Наприклад, не треба буде переписувати ні ARCVIEW, ні NC 4.0, ні DN, ні тех.BAT файлів, що ви могли створити за час користування ARJ'ем.

2) Для того, щоб використовувати можливості ARJZ'а при роботі з вашими старими архівами, вам зовсім не потрібно переархівировать їх заново.

3) Ви так само майже рятуєтеся від необхідності вивчати новий архіватор. Знаючи, як запускається ARJ, ви знаєте, як запускається ARJZ.

Однак, варто мати на увазі, що:

1) ARJZ дозволяє стискати файли, використовуючи більш могутні методи, чим оригінальна програма. У цьому випадку ARJ НЕ ЗМОЖЕ ПРОВОДИТИ ОБРОБКУ ОТРИМАНИХ АРХІВІВ, ЗВ'ЯЗАНУ З РОЗПАКУВАННЯМ, тобто деархівирование, тестування і т.д. У будь-якому випадку ви збережете можливість обновляти і зливати архіви, чи перейменовувати видаляти файли в них, а так само одержувати список файлів в архівах.

2) ARJZ/UNARJZ з одного боку, підтримують не всі команди й опції ARJ'а, а з іншого боку - уводять нові і це може створювати проблеми при роботі. У дійсності такі проблеми зустрічаються надзвичайно рідко і легко розв'язні.  
  
 *Переваги і недоліки*

До переваг ARJZ можна віднести:

1) Версії під DOS (реальний/розширений режими), OS/2 і NT. У програму для розширеного режиму DOS убудований розширник, тому вона працює на комп'ютерах 386+ без якого-небудь додаткового програмного забезпечення.

2) Високу швидкість стиску: ARJZ стискає файли з тією же якістю, що і ARJ приблизно в півтора разу швидше останнього (крім версії, що працює в реальному режимі).

3) Високий ступінь стиску (у цьому випадку отримані архіви не будуть розпаковуватися АRJ'ем). По цьому параметрі ARJZ знаходиться на рівні RAR/UC2 (у цьому ви можете переконається самі - you see too ;-).

4) Так називаний "напівекранний інтерфейс". ARJZ може під час роботи виводити на екран віконце з двома індикаторами процесу, ім'ям архіву й ім'ям пакуємого файлу - це чудова особливість призначена спеціально для таких програм, як ARC- чи ARJVIEW, SHEZ, ARJMENU, NC 4.0+, DN і ін.

5) Тут, звичайно не місце для опису переваг UNаRJZ'а, але проте... Висока швидкість розпакування. Навіть на XT UNARJZ працює в середньому в 1.5-2 рази швидше, ніж ARJ, а при використанні спеціальної опції (див. UNARJZ.DOC) різниця зростає ще в два рази.

Важливо відзначити, що процедури деархіватора оптимізовані окремо під процесори 286, 386, 486 і Pentium.

1. ARJZ написаний таким чином, що його можна використовувати і як окремий архіватор і як надбудову над ARJ'ем: якщо він не може розпізнати чи команд опцій командного рядка, то запускає оригінальну програму. Це, фактично, означає, що, використовуючи ARJZ, ви, проте, не втрачаєте ні однієї опції ARJ'а.  
     
    Недоліки ARJZ:

1) У ARJZ (принаймні, поки) немає підтримки багатотомних (multi volume), резервних (backup) і самораспакующихся (SFX) архівів. Помітьте, що UNARJZ розпаковує будь-які архіви, створені ARJ.  
 2) ARJZ не є повноцінним архіватором у тім змісті, що він самостійно не видаляє і не перейменовує файли в архівах, не може зливати архіви і т.д. Усю цю роботу можна зробити за допомогою оригінальної програми, тому не можна говорити, що пари ARJZ/UNARJZ цілком заміняє собою ARJ.

# Література

1. Фігурнов В.Е. «IBM PC для користувача. Короткий курс.» - М.: ИНФРА-М, 1998. - 480 с.: іл
2. PCMagazine, 1997-1999 р.