

**Kompresja danych**

**PROJEKT**

**„Porównanie archiwizerów”**

**Dariusz Jańczuk**  
**Krzysztof Mazurek**

## Spis treści:

1	Wstęp.....	4
2	Krótką charakterystyka metod kompresji .....	4
2.1	7ZIP .....	4
2.2	ACE.....	4
2.3	ARJ.....	5
2.4	BZIP2 .....	5
2.5	GZIP .....	5
2.6	LZH .....	6
2.7	RAR.....	6
2.8	ZIP .....	6
3	Opis algorytmów kompresji .....	6
3.1	LZ77 .....	6
3.2	LZ78 .....	7
3.3	LZSS.....	7
3.4	LZW .....	7
3.5	LZH.....	8
3.6	DEFLATE .....	8
3.7	LZMA.....	8
3.8	PPM.....	9
3.9	BWT .....	9
3.10	Kodowanie Huffmana/Shannona-Fano .....	9
3.11	Move To Front .....	9
4	Testy archiwizerów .....	9
4.1	Uwagi wstępne .....	9
4.2	Dane testowe .....	10
4.3	Krótką charakterystyka danych testowych.....	12
4.4	Testy.....	12
4.4.1	7-Zip:.....	12
4.4.2	ACE.....	13
4.4.3	ARJ.....	13
4.4.4	BZIP2 .....	13
4.4.5	GZIP .....	13
4.4.6	LZH.....	13
4.4.7	RAR.....	14
4.4.8	ZIP .....	14
4.5	Wyniki testów .....	14
4.5.1	Legenda .....	14
4.5.2	Test archiwizera 7-ZIP .....	14
4.5.3	Test archiwizera ACE .....	16
4.5.4	Test archiwizera ARJ .....	18
4.5.5	Test archiwizera BZIP2.....	19
4.5.6	Test archiwizera GZIP .....	21
4.5.7	Test archiwizera LZH.....	22
4.5.8	Test archiwizera RAR .....	24
4.5.9	Test archiwizera ZIP .....	25
4.5.10	Test archiwizacji danych tekstowych.....	27
4.5.11	Test archiwizacji plików audio .....	27
4.5.12	Test archiwizacji plików graficznych .....	28

4.5.13	Test archiwizacji dokumentów.....	28
4.5.14	Test archiwizacji plików wykonywalnych.....	29
4.5.15	Test archiwizacji sekwencji animowanych.....	29
5	Interpretacja wyników.....	31
5.1	Efektywność poszczególnych metod kompresji.....	31
5.2	Podatność różnych typów danych na kompresję.....	35
5.3	Zależność między typem danych a metodami kompresji.....	36
6	Podsumowanie.....	37

# 1 Wstęp

Niniejsze opracowanie ma na celu porównanie podstawowych parametrów powszechnie znanych i stosowanych archiwizerów. Przez archiwizer rozumiemy tutaj nie tyle konkretną aplikację umożliwiającą bezstratną kompresję i dekompresję danych, co samą metodę używaną do tej kompresji. Tak się akurat składa, że każda z metod kompresji jest reprezentowana przez jakiś „flagowy” program, obsługujący właśnie tę metodę (i być może inne). W testach skorzystano w znacznej mierze z tychże reprezentatywnych aplikacji, przy spełnieniu innych warunków (o czym później). Porównaniu poddano następujące metody kompresji: 7Zip, ACE, ARJ, BZIP2, GZIP, LZH, RAR, ZIP.

## 2 Krótka charakterystyka metod kompresji

### 2.1 7ZIP

Jedną z nowszych metod kompresji danych, charakteryzującą się dużymi współczynnikami kompresji. Rozwijany jest on jako standard otwarty, umożliwiający zastosowanie wielu algorytmów kompresji, kodowania i szyfrowania danych. Umożliwia kompresję katalogów i plików o maksymalnym rozmiarze 16000000000 GB. Nazwy przechowywanych zbiorów są zapisane w formacie UNICODE, umożliwiono również tworzenie samorozpakowujących się archiwów (SFX dla systemów Win32). Mimo, iż jest w dalszym ciągu bardzo intensywnie rozwijany, oferuje już kilka wariantów kompresji dla różnych typów danych, przy użyciu różnych algorytmów. Podstawowy i domyślny engine wykorzystuje algorytm LZMA, będący ulepszoną i zoptymalizowaną wersją algorytmu LZ77. Dostępna jest również standardowa metoda LZ77 (deflate), a także metody BCJ i BCJ2, zorientowane na kodowanie binarnych plików wykonywalnych w architekturze x86. Polegają one na konwersji kodu wejściowego do postaci bardziej kompresowalnej, poprzez umiejętne przetworzenie występujących w kodzie rozkazów CALL, JMP, i JCC (dla BCJ2). Oczywiście jest również dostępny popularny algorytm transformacji z użyciem transformacji Burrowsa – Wheeler’a. Wachlarz algorytmów uzupełnia ponadto predykcyjna metoda PPMd, która z kolei jest modyfikacją metody PPMdH autorstwa Dymitra Shkarina. Metoda ta jest nie tyle jakimś jednolitym standardem, lecz swojego rodzaju kontenerem, w którym można umieszczać dane spakowane różnymi metodami. Do testów tego formatu użyto programu 7-Zip w wersji 3.13 dla systemu Windows. Dostępna jest również aplikacja obsługująca ten format dla systemów UNIX o nazwie p7zip. Z tą metodą zetknąć się można również przy stosowaniu automatycznych narzędzi instalacyjnych NSIS i Inno Setup.

### 2.2 ACE

ACE jest komercyjną metodą kompresji danych. Jest ona rozwijana i utrzymywana przez niemiecką firmę WinAce. Umożliwia spakowanie zarówno plików, jak i katalogów, tworzenie archiwów samorozpakowujących się. W wersji 2.0 metody ACE, zostały zaimplementowane specjalne ulepszenia wykorzystywane do kompresji różnych typów danych (metody DELTA i EXE dla plików wykonywalnych, dodatkowe rozszerzenia dla kompresji grafiki i dźwięku). Używanymi algorytmami kompresji są zmodyfikowany

algorytm LZ77, oraz algorytm Huffmana. Ze względu na komercyjny charakter tego formatu do jej testów użyto natywnego programu WinAce w wersji 2.5 (który obsługuje również inne metody kompresji, m. in. ZIP, RAR i inne), przeznaczonego dla systemów Windows, choć istnieje także uniksowe narzędzie do dekompresji zwane unace.

## **2.3 ARJ**

ARJ jest kolejną popularną i szeroko stosowaną metodą kompresji. Jest ona dostępna zarówno w postaciach komercyjnych, jak i bezpłatnych. Umożliwia ona kompresję katalogów, podział spakowanego archiwum na wiele części (w celu np. przeniesienia archiwum na kilka dyskietek), oraz wybranie kilku stopni kompresji. Występuje w wersjach zarówno dla systemów UNIX/Linux, jak i Windows (w tym środowisku był on testowany). Podstawą działania jest algorytm LZSS i kodowanie Huffmana. Mimo, iż nie jest to format najnowszy, charakteryzuje się on przyzwoitymi wynikami kompresji, i cieszy się on uznaniem wielu użytkowników na całym świecie. Do testów użyto aplikacji konsolowej WinARJ32, pracującej w środowisku Windows.

## **2.4 BZIP2**

Format BZIP2 jest stosowany głównie przez użytkowników systemów UNIX/Linux. Operuje on wyłącznie na jednolitych zbiorach danych (plikach), w związku z czym nie nadaje się do kompresji katalogów- można to jednak ominąć poprzez umieszczenie katalogów w zbiorze TAR, a następnie spakowanie go przy użyciu Bzip2. Jest on stosowany również w nowych wersjach systemu pakietów RPM dla systemów Linux. BZIP2 używa blokowego mechanizmu kompresji, dane są dzielone na bloki o długości od 100 do 900 KB, o skokach co 100KB. Dla każdego bloku stosowany jest transformacja Burrowsa – Wheeler’a, następnie używa się algorytmu Move to Front, aby w końcu zakodować tak przekształcone dane metodą Huffmana. BZIP2 osiąga znacznie lepsze (zwykle o 10%-20%) wyniki niż algorytmy strumieniowe typu deflate, jednak nie może być używany do kodowania strumieni. Warto tu wspomnieć, że metoda BZIP2 posiada zoptymalizowane pod własnym kątem algorytmy BWT, zazwyczaj wydajniejsze od standardowych. Metoda zużywa do procesu kompresji maksymalnie 8 MB pamięci, do dekompresji 4 MB. Standardowym narzędziem wykorzystującym tę metodę jest uniksowy archiwizator bzip2. Testy tego formatu kompresji zostały jednak przeprowadzone w systemie Windows przy użyciu windowsowej wersji tego programu.

## **2.5 GZIP**

Kolejną otwartym standardem kompresji jest metoda GZIP (GNU Zip), która jest najbardziej rozpowszechniona wśród użytkowników systemów UNIX/Linux. Podobnie jak BZIP2, operuje ona na skonsolidowanych (zazwyczaj programem TAR) archiwach, mimo iż stosuje inne algorytmy. GZIP jest oparty o algorytm deflate, będący kombinacją algorytmu LZ77 i kodowania Huffmana. Powoduje to, że metoda ta jest szybsza od innej stosowanej metody BZIP2, aczkolwiek okupione to jest słabszym stopniem kompresji. Testy tego formatu przeprowadzono w środowisku Windows przy użyciu natywnego dla tego formatu

programu gzip. Ciekawą opcją tego formatu jest kompresja plików znajdujących się wewnątrz katalogów.

## **2.6 LZH**

LZH jest stosunkowo mało znaną i rzadko stosowaną metodą archiwizacji plików. Powstała ona w Japonii w latach 80-tych, i mimo iż jest teraz stosunkowo mało znana, wpłynęła ona na sposób implementacji innych metod kompresji (zwłaszcza stosujących algorytmy LZ77/78). Wykorzystuje ona algorytm LZH/LHA, będący mieszanką algorytmu LZSS i adaptacyjnej metody Huffmana. Umożliwia pakowanie katalogów, jednak nie udostępnia możliwości utworzenia archiwum SFX. Jest dostępny głównie w systemach Windows (za sprawą dużą liczbą programów obsługujących dekompresję tego formatu), w systemach uniksowych raczej nie występuje (aczkolwiek są dostępne źródła tego formatu). Testy przeprowadzono w systemie Windows z użyciem aplikacji PowerArchiver 8.8.

## **2.7 RAR**

Archiwizer RAR jest bodajże i jedną z najbardziej popularnych, i jedną z najlepszych metod kompresji danych. Jest stosowana powszechnie do przesyłania zarchiwizowanych danych przez Internet. Umożliwia tworzenie archiwów samorozpakowujących się (SFX), magazynowanie katalogów, oraz podział archiwum na wiele części. Wykorzystuje algorytmy LZSS i kodowanie Huffmana, występuje zarówno w systemach Windows (za sprawą koronnej aplikacji WinRAR), jak i w wersjach opensource dla systemów UNIX/Linux. Testy tego formatu przeprowadzono właśnie za pomocą programu WinRAR w wersji 3.4. Archiwizer oferuje ponadto wsparcie dla kompresji obrazów w trybie TrueColor, tekstów, plików audio, oraz plików wykonywalnych.

## **2.8 ZIP**

ZIP jest chyba najpopularniejszą ze wszystkich poddanych porównaniu metod kompresji danych. Występuje powszechnie na wielu platformach i w kilku odmianach (do testów wzięto odmianę utrzymywaną przez WinZip; istnieje również odmiana PKZIP). Nie jest to jednak metoda najefektywniejsza, aczkolwiek oferuje możliwość zarówno magazynowania katalogów, jak i tworzenie archiwów SFX. Wykorzystywany jest tutaj, podobnie jak w GZIP, algorytm deflate. Do dyspozycji mamy kilka stopni kompresji- w zależności od wyboru poziomu wybierana jest metoda redukcji danych- dla opcji najszybszej wybierany jest unshrinking, dla kompresji średniej- imploding z drzewami Shannona- Fano, a dla maksymalnego upakowania deflate. Do testów tego formatu wykorzystano program PowerArchiver w wersji 8.8 w środowisku Windows.

# **3 Opis algorytmów kompresji**

## **3.1 LZ77**

Istota działania algorytmu polega na poszukiwaniu w słowniku sekwencji podobnej do tej, która właśnie jest odczytywana. Tym co odróżnia ten algorytm od innych jest to, że słownikiem jest przesuwne okno zawierające  $N$  ostatnio odczytanych znaków. Zapamiętywany jest wskaźnik do odnalezionego ciągu, długość ciągu i jego pierwszy znak, którego nie udało się już dopasować. Algorytmicznie można to zapisać jako:

1. Ustaw się na początku kodowanego strumienia
2. Wśród sekwencji już odczytanej będącej oknem przesuwnym znajdź najdłuższy ciąg identyczny z tym, który jest w buforze wejściowym
3. Zapamiętaj wskaźnik do odnalezionego ciągu i pierwszą literę z bufora wejściowego, której nie udało się już dopasować
4. Jeśli bufor wejściowy nie jest pusty przesuń wskaźnik pozycji aktualnie kodowanej od  $L+1$  znaków, gdzie  $L$  jest długością sekwencji, która ostatnio udało się dopasować i wróć do punktu 2.

### **3.2 LZ78**

Powyższy algorytm kompresji można zaszeregować do metod kompresji słownikowych. Mając odczytany jeden znak z sekwencji wejściowej poszukujemy go w słowniku. Jeśli udało się odnaleźć taką sekwencję to odczytujemy kolejny znak i poszukujemy sekwencji dwóch znaków w słowniku. Poszukiwania kończymy mając na wejściu sekwencję, która nie występuje w słowniku. Na wyjście kierujemy słowo kodowe odpowiadające danej sekwencji i ostatni znak z sekwencji, dodajemy ją również do słownika.

W przypadku, gdy w słowniku nie występuje nawet jednoznakowa sekwencja, której poszukujemy na wyjście kierowany jest specjalne słowo kodowe oznaczające brak odpowiednika w słowniku jak również znak, którego nie udało się znaleźć.

### **3.3 LZSS**

Algorytm ten podobny jest do algorytmu LZ77. Jedną z różnic jest to, że wszystkie ciągi kodowane są za pomocą pary długości i pozycji a nie tylko te, które udało odnaleźć się w oknie przesuwnym będącym słownikiem. Kolejnymi krokami algorytmu są:

1. Zainicjalizuj słownik
2. Odczytaj z bufora wejściowego sekwencję wejściową o długości odpowiadającej największej długości, jaka może być odnaleziona w słowniku
3. Poszukaj najdłuższą sekwencję w słowniku odpowiadającą odczytanej sekwencji
4. Jeżeli udało się odnaleźć sekwencję dłuższą lub równą minimalnej dopuszczalnej sekwencji
  - a. zapisz na wyjściu flagę „zakodowano”, pozycję i długość sekwencji
  - b. w przeciwnym wypadku zapisz na wyjściu flagę „nie zakodowano” i pierwszy nie zakodowany symbol
5. Przesuń zakodowany ciąg do słownika
6. Odczytaj z wejścia ciąg znaków o długości równej ciągowi z punktu 4
7. Powtarzaj począwszy od kroku 3 do końca strumienia wejściowego

### **3.4 LZW**

Algorytm ten należy do kategorii algorytmów słownikowych. Na wyjście kierowane są tylko słowa kodowe, zatem słownik na początku nie może być pusty i musi zawierać

wszystkie symbole, jakie mogą pojawić się na wejściu. Ponieważ słownik od samego początku jest zainicjalizowany jednoznakowymi sekwencjami to poszukiwania zawsze rozpoczyna się od sekwencji dwuznakowych.

1. Zainicjalizuj słownik, sekwencja S jest pusta
2. Z – następny odczytany znak z wejścia
3. Czy ciąg S + Z znajduje się w słowniku?
  - a. jeśli tak to  $S:=S + Z$
  - b. jeśli nie to na wyjście prześlij słowo kodowe odpowiadające sekwencji S, dodaj ciąg S + Z do słownika,  $S:=Z$
4. Czy pozostały jeszcze jakieś znaki w strumieniu wejściowym?
  - a. jeśli tak to wróć do kroku 2
  - b. jeśli nie to zapisz na wyjściu słowo kodowe odpowiadające S i zakończ pracę

### **3.5 LZH**

Algorytm ten jest modyfikacją algorytmu LZSS polegającą na dodatkowym dynamicznym kodowaniu Huffmana wyjścia algorytmu LZSS.

### **3.6 DEFLATE**

Istota działania algorytmu polega na wyborze sposobu kodowania ciągu sekwencji wejściowej. Dostępne są trzy tryby kompresji:

1. Brak kompresji – wybierane wtedy, gdy kompresor stwierdzi, że dane już zostały skompresowane. Zapobiega to znacznemu wzrostowi wielkości kodowanego ciągu.
2. Kompresja najpierw algorytmem LZ77 a potem za pomocą kodowania Huffmana na bazie drzew zdefiniowanych w algorytmie Deflate (dzięki temu na wyjściu nie trzeba dopisywać definicji drzewa kodu Huffmana).
3. Kompresja najpierw algorytmem LZ77 a potem za pomocą kodowania Huffmana na bazie drzew zbudowanych w procesie kodowania. Do ciągu danych wyjściowych dopisywana jest definicja samego drzewa.

Dane wejściowe dzielone są na bloki i każdy z nich kodowany jest za pomocą jednej z powyższych metod.

### **3.7 LZMA**

Algorytm LZMA bazuje na metodach słownikowych (jest modyfikacją metody LZ77). Do kompresji używany jest słownik o maksymalnym rozmiarze 256MB. Następnie w celu dopasowania wchodzących na wejście sekwencji symboli wykorzystuje się, w zależności od zadeklarowanego rozmiaru słownika, następujące metody: „Binary Tree with 2 bytes hashing”, „Binary Tree with 2-3(full) bytes hashing”, „Binary Tree with 2-3-4 bytes hashing”, „Binary Tree with 2-3-4(big) bytes hashing”, „Patricia Tree with 2-bits nodes, removing”, „Patricia Tree with 2-bits nodes”, „Patricia Tree with 2-bits nodes, 2-3 bytes hashing”, „Patricia Tree with 3-bits nodes, 2-3 bytes hashing”, „Patricia Tree with 4-bits nodes, 2-3 bytes hashing”, „Hash Chain with 2-3 bytes hashing” i „Hash Chain with 2-3-4 bytes hashing”. Metody „Binary Tree” charakteryzują się małym zużyciem pamięci, podczas gdy „Patricia Tree” zużywają jej więcej, przeszukując szybciej. Najmniejsze wymagania pamięciowe mają za to metody „Hash Chain”, które skutkują jednak najgorszą kompresją. LZMA wykorzystuje także ideę tzw. szybkich bajtów (fast bytes), których liczba waha się od 3 do 255. Większa ich ilość poprawia współczynnik kompresji, kosztem szybkości procesu



pakowania (i odwrotnie). Jest to szczególnie zauważalne dla porcji danych zawierających długie sekwencje powtarzających się symboli.

### **3.8 PPM**

PPM jest algorytmem kompresji opartym na modelu źródła Markova. W ogólności PPM szacuje prawdopodobieństwo znaku mogącego pojawić się na wejściu na podstawie określonej liczby znaków już odczytanych. Algorytm ten konceptualnie jest bardzo prosty jednak jego złożoność obliczeniowa jest bardzo duża.

### **3.9 BWT**

Transformata Burrowsa-Wheelera polega na zamianie kolejności bloku danych wejściowych tak, aby był on bardziej podatny na kompresję. Kolejnymi krokami algorytmu są:

1. Odczytaj blok N danych
2. Wygeneruj wszystkie N rotacji odczytanego bloku
3. Posortuj otrzymane łańcuchy leksykograficznie
4. Zapisz na wyjściu ostatni bajt każdej z rotacji oraz pozycję, na której znalazł pierwszy znak odczytanego bloku danych

### **3.10 Kodowanie Huffmana/Shannona-Fano**

Jest to jeden z najprostszych algorytmów kompresji bezstratnej. Praktycznie nie używa się go samodzielnie a jedynie jako ostatniego etapu kompresji w wielu algorytmach. Algorytm Huffmana polega na przypisywaniu skończonemu zbiorowi symboli o z góry znanych prawdopodobieństwach kodów o zmiennej liczbie bitów. Symbole te zapisywane są na wyjściu algorytmu.

### **3.11 Move To Front**

Jest to prosta transformacja strumienia danych, używana jako część niektórych algorytmów kompresji, zwykle bezpośrednio po transformacie Burrowsa-Wheelera. Algorytm polega na utrzymywaniu tablicy znaków wyjścia odpowiadających każdemu znakowi wejścia. Po każdym znaku zmieniana jest ta tablica poprzez przesunięcie go na początek.

## **4 Testy archiwizerów**

### **4.1 Uwagi wstępne**

Testy przeprowadzono w środowisku Windows. Zdecydowano się na to, ze względu na bogatą obsługę rozważanych metod kompresji przez aplikacje dostępne dla tego środowiska, jak i mając na uwadze wygodę przeprowadzania testów. Systemem operacyjnym był Windows XP SP1, zainstalowany na komputerze wyposażonym w procesor AMD Athlon XP 2500+, taktowany częstotliwością 1833 MHz (moc obliczeniowa 3640 megaflopów), oraz 512 MB pamięci RAM. Testy przeprowadzono na specjalnie spreparowanym zbiorze danych, testom poddano efektywność poszczególnych archiwizerów, oraz, w przybliżeniu, czas trwania procesu pakowania. Na podstawie danych testowych uogólniono statystyki kompresji

dla każdego archiwizera i każdego typu danych. Testy przeprowadzono przy parametrach aplikacji gwarantujących maksymalny stopień kompresji. Wymagało to czasem wykonania testów wstępnych (np. w przypadku 7-Zipa), w celu dopasowania optymalnej metody kompresji do poszczególnych typów danych testowych. Czas mierzony był bardzo niedokładnie i tylko dla danych o większej wolumetrii, jednak uzyskane wyniki są wystarczające do interpretacji wyników i sformułowania wniosków dotyczących porównywanych archiwizerów.

## 4.2 Dane testowe

Z uwagi na zastosowanie archiwizerów do różnych celów, a także w oparciu o ogólnodostępne materiały, wyselekcjonowano zbiór danych podzielony na następujące kategorie: dane tekstowe („txt”), pliki z dźwiękiem („sfx”), pliki graficzne („gfx”), pliki wykonywalne („bin”), dokumenty („doc”), oraz sekwencje animowane („anime”). Każdą kategorię danych uporządkowano w podkategoriach wg. rozmiaru plików – pliki o rozmiarze do 10 KB umieszczono w podkategoriach „small”, pliki z przedziału 10 KB – 100 KB uszeregowano w kategorii „medium”, a większe w kategorii „big”, przy czym dane dobrano tak, aby dla zakresów „small” i „medium” pliki miały rozmiary w górnego zakresu przedziału, a pliki oznaczone jako „big” miały rozmiary przynajmniej 1 MB. Postarano się także o zróżnicowanie struktury i formatu danych. Od tych reguł jest jednak kilka wyjątków, np. pliki „medium” z animacjami mają rozmiary ok. 200 KB. Informacje o danych, podzielone na kategorie i podkategorie, zamieszczono w tabeli.

kategoria	podkategoria	plik	rozmiar	
txt	small	jspro.xml	10 076	
		Ksiega13.txt	9 465	
		Readme.htm	9 442	
	medium	index.html	32 079	
		Informatiker0(...)+EDVStatistik.htm	93 139	
		menu.vim	54 760	
	big	man_page.html	648 183	
		manual.txt	3 871 480	
		manual.html	4 447 182	
	sfx	small	ringin.wav	10 026
			ringout.wav	5 212
			start.wav	1 192
			Windows XP Dymek.wav	6 400
		medium	chord.wav	97 016
			flourish.mid	24 253
onestop.mid			40 075	
town.mid			22 097	
big		Dźwięk (...) Windows XP.wav	190 208	

		tada.wav	171 100
		Windows XP - Autostart.wav	424 644
<b>gfx</b>			
	<b>small</b>		
		8eee(...)e0fc.jpg	2 820
		pic0downloads.png	10 894
		sm_workstation6.gif	3 536
	<b>medium</b>		
		hol_257.jpg	44 766
		kpzr.jpg	99 269
		mizer2.gif	56 626
		pic-downloads.bmp	39 270
	<b>big</b>		
		4.jpg	314 054
		4.png	1 193 324
		wallpaper.bmp	2 359 350
<b>doc</b>			
	<b>small</b>		
		m1655108.pdf	6 625
		m1655116.pdf	6 776
		m1655122.pdf	7 181
	<b>medium</b>		
		doc.doc	72 192
		Księga 13.doc	42 496
		vlans.pdf	73 014
	<b>big</b>		
		11.ps	2 648 952
		Linux Kernel.pdf	1 307 314
		PHP Kompendium.doc	4 771 328
<b>bin</b>			
	<b>small</b>		
		hh.exe	10 752
	<b>medium</b>		
		lsb_un20.exe	119 808
	<b>big</b>		
		CoverDes.exe	2 199 552
<b>anime</b>			
	<b>small</b>		
		flaga_kuba.gif	26 836
		flaga_zsrr_2.gif	21 011
	<b>medium</b>		
		phpbb2_admin_disallowname.swf	184 259
		phpbb2_admin_edituser.swf	209 232
		phpbb2_admin_prune.swf	160 931
	<b>big</b>		
		better off bread.swf	3 091 456
		class act.swf	3 837 637
		remains to be seen.swf	4 131 201

### 4.3 Krótka charakterystyka danych testowych

**Dane tekstowe (txt)** – zbiór plików składających się w całości ze znaków z przedziału ASCII – zarówno pliki zawierające czysty tekst, jak i dokumenty HTML i XML.

**Pliki audio (sfx)** – zestaw plików magazynujących dźwięk w różnych formatach – zarówno proste strukturalnie pliki MID, jak i bardziej zaawansowane pliki WAV.

**Pliki graficzne (gfx)** – seria przykładowych obrazów zapisanych w popularnych i szeroko stosowanych formatach GIF, JPG, PNG, oraz - dla urozmaicenia – bitmap (pliki BMP). Należy wziąć pod uwagę fakt, iż pliki graficzne są już przeważnie skompresowane jakąś metodą (np. RLE, LZW), toteż należy się liczyć z ograniczonym efektem działania archiwizerów.

**Dokumenty (doc)** – zestaw plików w powszechnie stosowanych formatach służących przechowywaniu treści o charakterze dokumentalnym. Wyróżniono tu binarne pliki DOC i PDF, oraz plik w formacie Postscript, składające się wewnątrz ze znaków w kodzie ASCII. Dla tego formatu uczyniono wyjątek – testując metodę 7-Zip użyto do jego kompresji metody PPMd, podczas gdy dla pozostałych tryb LZMA.

**Pliki wykonywalne (bin)** – zbiór plików wykonywalnych w środowisku Microsoft Windows, zawierających sekwencje instrukcji języka maszynowego architektury x86.

**Sekwencje animowane (anime)** – zbiór plików graficznych GIF przechowujących sekwencję obrazów, oraz plików multimedialnych typu Flash (format SWF). Warto zwrócić uwagę na ten format, gdyż umożliwia on tworzenie również prezentacji interaktywnych. Ale, co bardziej istotne, stosuje on również kompresję typu Sorenson, stosowaną niegdyś do kompresji materiałów filmowych, co również wskazywałoby na niewielką efektywność kompresji. Jednak pliki SWF, wbrew oczekiwaniom, wykazały sporą podatność na spakowanie, co było dosyć nieoczekiwaną niespodzianką.

W testach zrezygnowano z kompresji plików multimedialnych (MP3, AVI, MPG), gdyż użycie tych archiwizerów przyniosłoby efekt raczej mizerny, a poza tym stosowane tam techniki zmniejszania objętości danych są tak skomplikowane, że nadają się na odrębny temat projektu z przedmiotu KODA.

### 4.4 Testy

Testy archiwizerów przeprowadzono (o czym już wspomniano wcześniej) przy pomocy następujących programów i aplikacji, które charakteryzowały się następującymi ustawieniami:

#### 4.4.1 7-Zip:

- aplikacja 7-Zip File Manager w wersji 3.13
- zastosowano następujące ustawienia (wartości odpowiednich pól):

Archive format: 7z

Compression level: Ultra  
Compression method: PPMd (dane tekstowe); LZMA (inne typy danych)  
Dictionary size: 32 MB (dla LZMA standardowo); 192 MB (standard dla PPMd)  
Word size: 64 (standardowo dla LZMA); 32 (standardowo dla PPMd)

Przed właściwymi testami wykonano serię wstępnych kompresji, z której wynikało, że dla danych tekstowych najlepszą typem kompresji jest tryb PPMd, podczas gdy dla danych o innym charakterze optymalny jest tryb LZMA.

#### 4.4.2 ACE

-aplikacja WinAce 2.5  
-konfiguracja poniżej:

W zakładce „Selection” ustawiony checkbox „Include subfolders”. W zakładce “Options” ustawiono pole “Archive Type” na „ACE”, pole „Action” na „Add”, a „Compression” na „Maximum”. Zakładka „Additional ACE options” ma ustawiony checkbox „Create solid archive”, pole „Dictionary” standardowo na 4096K, włączoną kompresję ACE 2.0 (checkbox „use ACE 2.0 compression”), oraz zaznaczone wszystkie opcje w panelu „Details” (wspomaganie kompresji specyficznych typów danych).

#### 4.4.3 ARJ

-konsolowa aplikacja WinARJ32 wersja 3.11  
-kompresja z następującymi opcjami:  
*arj32 a nazwa\_archiwum.arj nazwa\_pliku*  
-przy tak ustawionych opcjach archiwizator dopasowuje optymalną pod względem rozmiaru metodę kompresji.

#### 4.4.4 BZIP2

-program bzip2 w w wersji 1.0.2  
Wywołanie programu: *bzip2 -k nazwa\_pliku\_do\_spakowania*  
Tak wywołany archiwizator tworzy plik o nazwie takiej jak plik oryginalny z rozszerzeniem bz2 i o maksymalnym rozmiarze bloku (900K).

#### 4.4.5 GZIP

-program gzip wersja 1.2.4  
Wywołanie: *gzip nazwa\_pliku\_do\_spakowania* uruchamia archiwizator z domyślnymi opcjami (maksymalny stopień kompresji).

#### 4.4.6 LZH

-aplikacja PowerArchiver 8.80.04.

Kontrolka z formatami archiwów ustawiona na „LHA”, pole „Action” przyjęło wartość „Add (and replace) files”, pole „Compression” ustawiono na „Maximum (slowest) Frozen6”. Reszta opcji ustawiona domyślnie.

#### **4.4.7 RAR**

-aplikacja WinRAR 3.41

Kontrolka „Archive format” ustawiona na „RAR”, polu „Compression method” przypisano wartość „Best”, reszta opcji domyślna. W zakładce „Advanced” w okienku „Compression” ustawiono wszystkie typy kompresji na „Auto”, rozmiar słownika ustawiono na 4096K, uaktywniono opcjonalną kompresję binarną kodu dla procesorów Pentium i Itanium oraz kompresję delta.

#### **4.4.8 ZIP**

-program PowerArchiver 8.80.04

Kontrolka z formatami archiwów ustawiona na „ZIP”, pole „Action” przyjęło wartość „Add (and replace) files”, pole „Compression” ustawiono na „Maximum (slowest)”. Reszta opcji ustawiona domyślnie.

### **4.5 Wyniki testów**

Wykonano jeden wielki test dla wszystkich rozważanych archiwizerów na zgromadzonych danych testowych. Poniżej zaprezentowano wyniki zorientowane na różne aspekty testów.

#### **4.5.1 Legenda**

tenc – mierzony czas kompresji (w sekundach). Wartości zaznaczone na szaro zostały uznane za niedokładne.

Dla każdej kategorii w kolumnie z „[%]” w nazwie umieszczono wyrażony w procentach stosunek rozmiaru skompresowanego archiwum do rozmiaru oryginalnego. Pola zaznaczone na żółto zawierają procent średni dla danej podkategorii, a pola z jasnoniebieskim średni procent dla całej kategorii. Wartości pól na zielonym tle oznaczają czas kompresji dla danej podkategorii, ważony rozmiarem oryginalnym danych testowych (pomiar czasu przeprowadzono tylko dla dużych plików, dla których wpływ czasu był zauważalny). Pasek na samym dole zestawienia (kolor niebieski) zawiera informację zbiorczą na temat efektywności kompresji (w procentach) i ważonego czasu. Warto tu wspomnieć, że ważony czas jest miarą raczej „wirtualną”, i ma za zadanie zilustrowanie charakterystyki czasowej poszczególnych archiwizerów w celu ich porównania.

#### **4.5.2 Test archiwizera 7-ZIP**

kategoria	podkategoria	plik	rozmiar	7Z	7Z [%]	teno	
txt		średnie(txt)			19,62%		
	Small	średnie(small)			31,64%		
		jspro.xml	10 076	2 961	29,39%		
		Ksiega13.txt	9 465	4 099	43,31%		
			Readme.htm	9 442	2 098	22,22%	
	medium	średnie(medium)				14,13%	
		index.html	32 079	4 631	14,44%		
		Informatiker0(...)+EDVStatistik.htm	93 139	5 920	6,36%		
			menu.vim	54 760	11 830	21,60%	
	big	średnie(big)				13,09%	3,18
man_page.html		648 183	43 327	6,68%		0,2	
manual.txt		3 871 480	670 890	17,33%		3,3	
manual.html		4 447 182	678 082	15,25%		3,5	
sfx		średnie(gfx)			56,71%		
	small	średnie(small)				61,08%	
		ringin.wav	10 026	7 785	77,65%		
		ringout.wav	5 212	1 919	36,82%		
		start.wav	1 192	890	74,66%		
			Windows XP Dymek.wav	6 400	3 531	55,17%	
	medium	średnie(medium)				43,99%	
		chord.wav	97 016	64 094	66,07%		
		flourish.mid	24 253	8 606	35,48%		
		onestop.mid	40 075	20 862	52,06%		
			town.mid	22 097	4 942	22,37%	
	big	średnie(big)				67,85%	0,15
		Dźwięk (...) Windows XP.wav	190 208	108 864	57,23%		0,1
		tada.wav	171 100	126 794	74,11%		0,1
		Windows XP - Autostart.wav	424 644	306 648	72,21%		0,2
gfx		średnie(gfx)			86,27%		
	small	średnie(small)				102,60%	
		8eee(...)e0fc.jpg	2 820	2 912	103,26%		
		pic0downloads.png	10 894	11 108	101,96%		
		sm_workstation6.gif	3 536	3 627	102,57%		
	medium	średnie(medium)				83,05%	
		hol_257.jpg	44 766	44 845	100,18%		
		kpzr.jpg	99 269	99 867	100,60%		
		mizer2.gif	56 626	56 376	99,56%		
			pic-downloads.bmp	39 270	12 507	31,85%	
	big	średnie(big)				74,23%	2,31
		4.jpg	314 054	313 755	99,90%		0,2
		4.png	1 193 324	1 195 065	100,15%		1,5
wallpaper.bmp		2 359 350	534 256	22,64%		3	
doc		średnie(doc)			41,87%		
	small	średnie(small)				57,48%	
		m1655108.pdf	6 625	3 750	56,60%		
		m1655116.pdf	6 776	3 869	57,10%		
		m1655122.pdf	7 181	4 217	58,72%		
	medium	średnie(medium)				25,21%	
		doc.doc	72 192	12 086	16,74%		
		Księga 13.doc	42 496	6 824	16,06%		
			vlans.pdf	73 014	31 267	42,82%	
	big	średnie(big)				42,92%	6,95
11.ps		2 648 952	278 169	10,50%		1,5	

		Linux Kernel.pdf	1 307 314	1 010 472	77,29%	2,5
		PHP Kompendium.doc	4 771 328	1 954 912	40,97%	11,2
<b>bin</b>		średnie(bin)			35,38%	
	<b>small</b>	średnie(small)			31,59%	
		hh.exe	10 752	3 397	31,59%	
	<b>medium</b>	średnie(medium)			46,07%	
		lsb_un20.exe	119 808	55 194	46,07%	
	<b>big</b>	średnie(big)		czas ważony	28,47%	5
		CoverDes.exe	2 199 552	626 322	28,47%	5
<b>anime</b>		średnie(anime)			77,66%	
	<b>small</b>	średnie(small)			100,38%	
		flaga_kuba.gif	26 836	26 916	100,30%	
		flaga_zsrr_2.gif	21 011	21 106	100,45%	
	<b>medium</b>	średnie(medium)			73,13%	
		phpbb2_admin_disallowname.swf	184 259	135 952	73,78%	
		phpbb2_admin_edituser.swf	209 232	153 912	73,56%	
		phpbb2_admin_prune.swf	160 931	115 943	72,05%	
	<b>big</b>	średnie(big)		czas ważony	67,05%	6,45
		better off bread.swf	3 091 456	2 520 526	81,53%	5
		class act.swf	3 837 637	2 490 320	64,89%	6,5
		remains to be seen.swf	4 131 201	2 261 247	54,74%	7,5
		<b>ŚREDNIA OGÓLNA</b>			<b>55,35%</b>	<b>4,97</b>

### 4.5.3 Test archiwizera ACE

kategoria	podkategoria	plik	rozmiar	ACE	ACE [%]	tenr
<b>txt</b>		średnie(txt)			23,87%	
	<b>small</b>	średnie(small)			37,21%	
		jspro.xml	10 076	3 441	34,15%	
		Ksiega13.txt	9 465	4 857	51,32%	
		Readme.htm	9 442	2 470	26,16%	
	<b>medium</b>	średnie(medium)			16,99%	
		index.html	32 079	5 722	17,84%	
		Informatiker0(...)+EDVStatistik.htm	93 139	7 469	8,02%	
		menu.vim	54 760	13 756	25,12%	
	<b>big</b>	średnie(big)		czas ważony	17,41%	5,53
		man_page.html	648 183	57 861	8,93%	0,4
		manual.txt	3 871 480 4 447	880 894	22,75%	6,2
		manual.html	182	913 543	20,54%	5,7
<b>sfx</b>		średnie(gfx)			52,75%	
	<b>small</b>	średnie(small)			60,91%	
		ringin.wav	10 026	7 718	76,98%	
		ringout.wav	5 212	2 423	46,49%	
		start.wav	1 192	901	75,59%	
		Windows XP Dymek.wav	6 400	2 854	44,59%	
	<b>medium</b>	średnie(medium)			42,94%	
		chord.wav	97 016	49 201	50,71%	
		flourish.mid	24 253	9 076	37,42%	
		onestop.mid	40 075	23 175	57,83%	
		town.mid	22 097	5 696	25,78%	
	<b>big</b>	średnie(big)		czas ważony	54,96%	0,35
		Dźwięk (...) Windows XP.wav	190 208	90 435	47,55%	0,3



		tada.wav	171 100	104 016	60,79%	0,3
		Windows XP - Autostart.wav	424 644	240 114	56,54%	0,4
<b>gfx</b>		<b>średnie(gfx)</b>			<b>85,36%</b>	
	<b>small</b>	<b>średnie(small)</b>			<b>102,46%</b>	
		8eee(...)\e0fc.jpg	2 820	2 916	103,40%	
		pic0downloads.png	10 894	10 999	100,96%	
		sm_workstation6.gif	3 536	3 643	103,03%	
	<b>medium</b>	<b>średnie(medium)</b>			<b>82,09%</b>	
		hol_257.jpg	44 766	44 719	99,90%	
		kpzr.jpg	99 269	99 365	100,10%	
		mizer2.gif	56 626	56 494	99,77%	
		pic-downloads.bmp	39 270	11 225	28,58%	
	<b>big</b>	<b>średnie(big)</b>		<b>czas ważony</b>	<b>72,61%</b>	<b>0,78</b>
		4.jpg	314 054	312 373	99,46%	0,5
		4.png	1 193	1 199		
			324	293	100,50%	1,6
			2 359			
		wallpaper.bmp	350	421 149	17,85%	0,4
<b>doc</b>		<b>średnie(doc)</b>			<b>44,69%</b>	
	<b>small</b>	<b>średnie(small)</b>			<b>60,59%</b>	
		m1655108.pdf	6 625	3 956	59,71%	
		m1655116.pdf	6 776	4 084	60,27%	
		m1655122.pdf	7 181	4 436	61,77%	
	<b>medium</b>	<b>średnie(medium)</b>			<b>28,90%</b>	
		doc.doc	72 192	15 227	21,09%	
		Księga 13.doc	42 496	8 609	20,26%	
		vlans.pdf	73 014	33 113	45,35%	
	<b>big</b>	<b>średnie(big)</b>		<b>czas ważony</b>	<b>44,58%</b>	<b>2,85</b>
			2 648			
		11.ps	952	351 433	13,27%	5
			1 307	1 022		
		Linux Kernel.pdf	314	764	78,23%	1,6
			4 771	2 015		
		PHP Kompendium.doc	328	284	42,24%	2
<b>bin</b>		<b>średnie(bin)</b>			<b>38,20%</b>	
	<b>small</b>	<b>średnie(small)</b>			<b>33,43%</b>	
		hh.exe	10 752	3 594	33,43%	
	<b>medium</b>	<b>średnie(medium)</b>			<b>48,43%</b>	
		lsb_un20.exe	119 808	58 024	48,43%	
	<b>big</b>	<b>średnie(big)</b>		<b>czas ważony</b>	<b>32,73%</b>	<b>1,5</b>
			2 199			
		CoverDes.exe	552	719 872	32,73%	1,5
<b>anime</b>		<b>średnie(anime)</b>			<b>77,51%</b>	
	<b>small</b>	<b>średnie(small)</b>			<b>100,15%</b>	
		flaga_kuba.gif	26 836	26 854	100,07%	
		flaga_zsrr_2.gif	21 011	21 060	100,23%	
	<b>medium</b>	<b>średnie(medium)</b>			<b>73,81%</b>	
		phpbb2_admin_disallowname.swf	184 259	137 273	74,50%	
		phpbb2_admin_edituser.swf	209 232	155 285	74,22%	
		phpbb2_admin_prune.swf	160 931	117 010	72,71%	
	<b>Big</b>	<b>średnie(big)</b>		<b>czas ważony</b>	<b>66,12%</b>	<b>3,85</b>
			3 091	2 295		
		better off bread.swf	456	640	74,26%	3,5
			3 837	2 560		
		class act.swf	637	365	66,72%	4,5
			4 131	2 370		
		remains to be seen.swf	201	310	57,38%	3,5
		<b>ŚREDNIA OGÓLNA</b>			<b>55,71%</b>	<b>3,46</b>

## 4.5.4 Test archiwizera ARJ

kategoria	podkategoria	plik	rozmiar	ARJ	ARJ [%]	tenc
txt		średnie(txt)			25,80%	
	small	średnie(small)			37,52%	
		jspro.xml	10 076	3 486	34,60%	
		Ksiega13.txt	9 465	4 878	51,54%	
		Readme.htm	9 442	2 496	26,44%	
	medium	średnie(medium)			17,85%	
		index.html	32 079	5 939	18,51%	
		Informatiker0(...)+EDVStatistik.htm	93 139	7 995	8,58%	
	big	średnie(big)		czas ważony	22,04%	1,06
		man_page.html	648 183	67 844	10,47%	0,4
		manual.txt	3 871 480	1 121 335	28,96%	1
manual.html		4 447 182	1 187 482	26,70%	1,2	
sfx		średnie(gfx)			63,83%	
	small	średnie(small)			66,77%	
		ringin.wav	10 026	7 957	79,36%	
		ringout.wav	5 212	2 570	49,31%	
		start.wav	1 192	920	77,18%	
		Windows XP Dymek.wav	6 400	3 919	61,23%	
	medium	średnie(medium)			51,14%	
		chord.wav	97 016	73 588	75,85%	
		flourish.mid	24 253	9 667	39,86%	
		onestop.mid	40 075	24 832	61,96%	
		town.mid	22 097	5 939	26,88%	
	big	średnie(big)		czas ważony	76,85%	0,53
		Dźwięk (...) Windows XP.wav	190 208	132 092	69,45%	0,4
		tada.wav	171 100	136 843	79,98%	0,5
		Windows XP - Autostart.wav	424 644	344 470	81,12%	0,6
gfx		średnie(gfx)			87,61%	
	small	średnie(small)			103,42%	
		8eee(...)\e0fc.jpg	2 820	2 956	104,82%	
		pic0downloads.png	10 894	11 036	101,30%	
		sm_workstation6.gif	3 536	3 682	104,13%	
	medium	średnie(medium)			84,37%	
		hol_257.jpg	44 766	44 644	99,73%	
		kpzr.jpg	99 269	99 291	100,02%	
		mizer2.gif	56 626	56 277	99,38%	
		pic-downloads.bmp	39 270	15 057	38,34%	
	big	średnie(big)		czas ważony	76,12%	0,46
4.jpg		314 054	312 424	99,48%	0,7	
4.png		1 193 324	1 193 444	100,01%	0,5	
wallpaper.bmp		2 359 350	681 118	28,87%	0,4	
doc		średnie(doc)			46,19%	
	small	średnie(small)			61,46%	
		m1655108.pdf	6 625	4 020	60,68%	
		m1655116.pdf	6 776	4 147	61,20%	
	medium	średnie(medium)			29,72%	
		m1655122.pdf	7 181	4 489	62,51%	
		doc.doc	72 192	15 378	21,30%	

		Księga 13.doc	42 496	8 880	20,90%	
		vlans.pdf	73 014	34 298	46,97%	
<b>big</b>		<b>średnie(big)</b>		<b>czas ważony</b>	<b>47,39%</b>	<b>0,99</b>
		11.ps	2 648 952	450 441	17,00%	1
		Linux Kernel.pdf	1 307 314	1 043 431	79,81%	0,6
		PHP Kompendium.doc	4 771 328	2 163 373	45,34%	1,1
<b>bin</b>		<b>średnie(bin)</b>			<b>44,04%</b>	
<b>small</b>		<b>średnie(small)</b>			<b>35,54%</b>	
		hh.exe	10 752	3 821	35,54%	
<b>medium</b>		<b>średnie(medium)</b>			<b>55,74%</b>	
		lsb_un20.exe	119 808	66 779	55,74%	
<b>big</b>		<b>średnie(big)</b>		<b>czas ważony</b>	<b>40,84%</b>	<b>0,6</b>
		CoverDes.exe	2 199 552	898 242	40,84%	0,6
<b>anime</b>		<b>średnie(anime)</b>			<b>83,98%</b>	
<b>small</b>		<b>średnie(small)</b>			<b>100,02%</b>	
		flaga_kuba.gif	26 836	26 819	99,94%	
		flaga_zsrr_2.gif	21 011	21 032	100,10%	
<b>medium</b>		<b>średnie(medium)</b>			<b>85,58%</b>	
		phpbb2_admin_disallowname.swf	184 259	159 735	86,69%	
		phpbb2_admin_edituser.swf	209 232	176 610	84,41%	
		phpbb2_admin_prune.swf	160 931	137 840	85,65%	
<b>big</b>		<b>średnie(big)</b>		<b>czas ważony</b>	<b>71,68%</b>	<b>1,04</b>
		better off bread.swf	3 091 456	2 393 331	77,42%	1
		class act.swf	3 837 637	2 782 106	72,50%	1
		remains to be seen.swf	4 131 201	2 691 122	65,14%	1,1
		<b>ŚREDNIA OGÓLNA</b>			<b>60,60%</b>	<b>0,92</b>

#### 4.5.5 Test archiwizera BZIP2

		plik	rozmiar	BZIP2	BZIP2 [%]	renc
<b>txt</b>		<b>średnie(txt)</b>			<b>22,69%</b>	
<b>small</b>		<b>średnie(small)</b>			<b>35,48%</b>	
		jspro.xml	10 076	3 406	33,80%	
		Ksiega13.txt	9 465	4 385	46,33%	
		Readme.htm	9 442	2 485	26,32%	
<b>medium</b>		<b>średnie(medium)</b>			<b>16,42%</b>	
		index.html	32 079	5 412	16,87%	
		Informatiker0(...)+EDVStatistik.htm	93 139	7 325	7,86%	
		menu.vim	54 760	13 436	24,54%	
<b>big</b>		<b>średnie(big)</b>		<b>czas ważony</b>	<b>16,16%</b>	<b>2,03</b>
		man_page.html	648 183	49 118	7,58%	0,2
		manual.txt	3 871 480	839 181	21,68%	1,9
		manual.html	4 447 182	854 952	19,22%	2,4
<b>sfx</b>		<b>średnie(gfx)</b>			<b>63,22%</b>	
<b>small</b>		<b>średnie(small)</b>			<b>65,01%</b>	
		ringin.wav	10 026	8 304	82,82%	
		ringout.wav	5 212	1 914	36,72%	
		start.wav	1 192	932	78,19%	
		Windows XP Dymek.wav	6 400	3 988	62,31%	
<b>medium</b>		<b>średnie(medium)</b>			<b>53,40%</b>	
		chord.wav	97 016	69 846	71,99%	
		flourish.mid	24 253	11 089	45,72%	
		onestop.mid	40 075	25 650	64,00%	

		town.mid	22 097	7 045	31,88%	
<b>big</b>		średnie(big)	czas ważony		73,92%	0,36
		Dźwięk (...) Windows XP.wav	190 208	125 446	65,95%	0,2
		tada.wav	171 100	131 109	76,63%	0,2
		Windows XP - Autostart.wav	424 644	336 285	79,19%	0,5
<b>gfx</b>		średnie(gfx)			88,77%	
	<b>small</b>	średnie(small)			110,36%	
		8eee(...)e0fc.jpg	2 820	3 214	113,97%	
		pic0downloads.png	10 894	11 373	104,40%	
		sm_workstation6.gif	3 536	3 985	112,70%	
	<b>medium</b>	średnie(medium)			83,68%	
		hol_257.jpg	44 766	44 832	100,15%	
		kpzr.jpg	99 269	99 446	100,18%	
		mizer2.gif	56 626	56 975	100,62%	
		pic-downloads.bmp	39 270	13 266	33,78%	
	<b>big</b>	średnie(big)	czas ważony		73,98%	0,41
		4.jpg	314 054	313 865	99,94%	0,2
		4.png	1 193 324	1 194 739	100,12%	0,5
		wallpaper.bmp	2 359 350	516 126	21,88%	0,4
<b>doc</b>		średnie(doc)			46,11%	
	<b>small</b>	średnie(small)			65,77%	
		m1655108.pdf	6 625	4 330	65,36%	
		m1655116.pdf	6 776	4 431	65,39%	
		m1655122.pdf	7 181	4 779	66,55%	
	<b>medium</b>	średnie(medium)			27,43%	
		doc.doc	72 192	13 751	19,05%	
		Księga 13.doc	42 496	7 759	18,26%	
		vlans.pdf	73 014	32 851	44,99%	
	<b>big</b>	średnie(big)	czas ważony		45,12%	2,39
		11.ps	2 648 952	319 180	12,05%	1,9
		Linux Kernel.pdf	1 307 314	1 018 672	77,92%	1,5
		PHP Kompendium.doc	4 771 328	2 165 952	45,40%	2,9
<b>bin</b>		średnie(bin)			44,34%	
	<b>small</b>	średnie(small)			38,13%	
		hh.exe	10 752	4 100	38,13%	
	<b>medium</b>	średnie(medium)			55,89%	
		lsb_un20.exe	119 808	66 962	55,89%	
	<b>big</b>	średnie(big)	czas ważony		39,00%	1,2
		CoverDes.exe	2 199 552	857 879	39,00%	1,2
<b>anime</b>		średnie(anime)			82,76%	
	<b>small</b>	średnie(small)			101,44%	
		flaga_kuba.gif	26 836	27 178	101,27%	
		flaga_zsrr_2.gif	21 011	21 347	101,60%	
	<b>medium</b>	średnie(medium)			80,74%	
		phpbb2_admin_disallowname.swf	184 259	149 721	81,26%	
		phpbb2_admin_edituser.swf	209 232	169 631	81,07%	
		phpbb2_admin_prune.swf	160 931	128 562	79,89%	
	<b>big</b>	średnie(big)	czas ważony		72,32%	2,64
		better off bread.swf	3 091 456	2 443 817	79,05%	2,2
		class act.swf	3 837 637	2 807 898	73,17%	2,6
		remains to be seen.swf	4 131 201	2 674 911	64,75%	3
		ŚREDNIA OGÓLNA			59,95%	2,03

## 4.5.6 Test archiwizera GZIP

kategoria	podkategoria	plik	rozmiar	GZIP	GZIP [%]	tenr
txt		średnie(txt)			25,18%	
	small	średnie(small)			36,30%	
		jspro.xml	10 076	3 372	33,47%	
		Ksiega13.txt	9 465	4 758	50,27%	
		Readme.htm	9 442	2 376	25,16%	
	medium	średnie(medium)			17,49%	
		index.html	32 079	5 727	17,85%	
		Informatiker0(...)+EDVStatistik.htm	93 139	7 947	8,53%	
	big	średnie(big)		czas ważony	21,75%	0,57
		man_page.html	648 183	68 260	10,53%	0,1
		manual.txt	3 871 480	1 102 579	28,48%	0,5
		manual.html	4 447 182	1 167 302	26,25%	0,7
	sfx		średnie(gfx)			62,72%
small		średnie(small)			63,70%	
		ringin.wav	10 026	7 840	78,20%	
		ringout.wav	5 212	2 470	47,39%	
		start.wav	1 192	830	69,63%	
		Windows XP Dymek.wav	6 400	3 813	59,58%	
medium		średnie(medium)			50,91%	
		chord.wav	97 016	74 273	76,56%	
		flourish.mid	24 253	9 484	39,10%	
		onestop.mid	40 075	24 701	61,64%	
big		średnie(big)		czas ważony	77,17%	0,10
		Dźwięk (...) Windows XP.wav	190 208	133 238	70,05%	0,1
		tada.wav	171 100	137 395	80,30%	0,1
		Windows XP - Autostart.wav	424 644	344 633	81,16%	0,1
gfx			średnie(gfx)			86,76%
	small	średnie(small)			100,86%	
		8eee(...).e0fc.jpg	2 820	2 846	100,92%	
		pic0downloads.png	10 894	10 937	100,39%	
	medium	średnie(medium)			84,17%	
		sm_workstation6.gif	3 536	3 581	101,27%	
		hol_257.jpg	44 766	44 857	100,20%	
		kpzr.jpg	99 269	99 109	99,84%	
	big	średnie(big)		czas ważony	76,12%	0,25
		mizer2.gif	56 626	56 164	99,18%	
		pic-downloads.bmp	39 270	14 712	37,46%	
		4.jpg	314 054	312 213	99,41%	0,1
	doc	small	średnie(doc)			45,69%
średnie(small)					59,98%	
m1655108.pdf			6 625	3 915	59,09%	
medium		m1655116.pdf	6 776	4 047	59,73%	
		m1655122.pdf	7 181	4 389	61,12%	
		średnie(medium)			29,81%	
		doc.doc	72 192	15 237	21,11%	
Księga 13.doc	42 496	8 813	20,74%			
vlans.pdf	73 014	34 745	47,59%			

<b>big</b>	<b>średnie(big)</b>		<b>czas ważony</b>	<b>47,28%</b>	<b>0,47</b>
	11.ps	2 648 952	451 561	17,05%	0,2
	Linux Kernel.pdf	1 307 314	1 040 510	79,59%	0,2
	PHP Kompendium.doc	4 771 328	2 157 258	45,21%	0,7
<b>bin</b>	<b>średnie(bin)</b>			<b>43,65%</b>	
	<b>small</b>	<b>średnie(small)</b>		<b>34,56%</b>	
<b>medium</b>	hh.exe	10 752	3 716	34,56%	
	lsb_un20.exe	119 808	66 789	55,75%	
<b>big</b>	<b>średnie(big)</b>		<b>czas ważony</b>	<b>40,64%</b>	<b>0,3</b>
	CoverDes.exe	2 199 552	893 872	40,64%	0,3
<b>anime</b>	<b>średnie(anime)</b>			<b>83,37%</b>	
	<b>small</b>	<b>średnie(small)</b>		<b>99,43%</b>	
<b>medium</b>	flaga_kuba.gif	26 836	26 685	99,44%	
	flaga_zsrr_2.gif	21 011	20 890	99,42%	
<b>big</b>	<b>średnie(big)</b>		<b>czas ważony</b>	<b>71,35%</b>	<b>0,47</b>
	better off bread.swf	3 091 456	2 387 771	77,24%	0,4
	class act.swf	3 837 637	2 774 848	72,31%	0,4
	remains to be seen.swf	4 131 201	2 665 439	64,52%	0,6
<b>ŚREDNIA OGÓLNA</b>				<b>59,87%</b>	<b>0,45</b>

## 4.5.7 Test archiwizera LZH

kategoria	podkategoria	plik	rozmiar	LZH	LZH[%]	tenr
<b>txt</b>	<b>średnie(txt)</b>				<b>25,78%</b>	
	<b>small</b>	<b>średnie(small)</b>			<b>36,58%</b>	
<b>medium</b>	jspro.xml	10 076	3 398	33,72%		
	Ksiega13.txt	9 465	4 784	50,54%		
	Readme.htm	9 442	2 404	25,46%		
	<b>średnie(medium)</b>				<b>17,94%</b>	
<b>big</b>	index.html	32 079	5 952	18,55%		
	Informatiker0(...)+EDVStatistik.htm	93 139	7 897	8,48%		
	menu.vim	54 760	14 673	26,80%		
	<b>średnie(big)</b>		<b>czas ważony</b>	<b>22,82%</b>	<b>2,62</b>	
<b>medium</b>	man_page.html	648 183	69 616	10,74%	0,2	
	manual.txt	3 871 480	1 160 555	29,98%	2,6	
	manual.html	4 447 182	1 233 552	27,74%	3	
	<b>średnie(gfx)</b>				<b>73,61%</b>	
<b>small</b>	<b>średnie(small)</b>				<b>69,21%</b>	
	ringin.wav	10 026	10 060	100,34%		
<b>medium</b>	ringout.wav	5 212	2 474	47,47%		
	start.wav	1 192	827	69,38%		
	Windows XP Dymek.wav	6 400	3 817	59,64%		
	<b>średnie(medium)</b>				<b>58,20%</b>	
<b>medium</b>	chord.wav	97 016	97 049	100,03%		
	flourish.mid	24 253	9 577	39,49%		
	onestop.mid	40 075	26 758	66,77%		
	town.mid	22 097	5 854	26,49%		

	<b>big</b>	<b>średnie(big)</b>	<b>czas ważony</b>		<b>100,02%</b>	<b>0,10</b>
		Dźwięk (...) Windows XP.wav	190 208	190 271	100,03%	0,1
		tada.wav	171 100	171 132	100,02%	0,1
		Windows XP - Autostart.wav	424 644	424 694	100,01%	0,1
<b>gfx</b>		<b>średnie(gfx)</b>			<b>87,07%</b>	
	<b>small</b>	<b>średnie(small)</b>			<b>101,24%</b>	
		8eee(...)e0fc.jpg	2 820	2 880	102,13%	
		pic0downloads.png	10 894	10 935	100,38%	
		sm_workstation6.gif	3 536	3 579	101,22%	
	<b>medium</b>	<b>średnie(medium)</b>			<b>84,51%</b>	
		hol_257.jpg	44 766	44 801	100,08%	
		kpzr.jpg	99 269	99 301	100,03%	
		mizer2.gif	56 626	56 660	100,06%	
		pic-downloads.bmp	39 270	14 879	37,89%	
	<b>big</b>	<b>średnie(big)</b>	<b>czas ważony</b>		<b>76,31%</b>	<b>0,89</b>
		4.jpg	314 054	314 083	100,01%	0,1
		4.png	1 193 324	1 193 353	100,00%	0,1
		wallpaper.bmp	2 359 350	682 130	28,91%	1,4
<b>doc</b>		<b>średnie(doc)</b>			<b>48,16%</b>	
	<b>small</b>	<b>średnie(small)</b>			<b>60,09%</b>	
		m1655108.pdf	6 625	3 926	59,26%	
		m1655116.pdf	6 776	4 053	59,81%	
		m1655122.pdf	7 181	4 394	61,19%	
	<b>medium</b>	<b>średnie(medium)</b>			<b>29,88%</b>	
		doc.doc	72 192	15 692	21,74%	
		Księga 13.doc	42 496	8 806	20,72%	
		vlans.pdf	73 014	34 457	47,19%	
	<b>big</b>	<b>średnie(big)</b>	<b>czas ważony</b>		<b>54,50%</b>	<b>1,70</b>
		11.ps	2 648 952	464 441	17,53%	2,5
		Linux Kernel.pdf	1 307 314	1 307 354	100,00%	0,1
		PHP Kompendium.doc	4 771 328	2 193 478	45,97%	1,7
<b>bin</b>		<b>średnie(bin)</b>			<b>43,99%</b>	
	<b>small</b>	<b>średnie(small)</b>			<b>34,78%</b>	
		hh.exe	10 752	3 740	34,78%	
	<b>medium</b>	<b>średnie(medium)</b>			<b>55,76%</b>	
		lsb_un20.exe	119 808	66 799	55,76%	
	<b>big</b>	<b>średnie(big)</b>	<b>czas ważony</b>		<b>41,43%</b>	<b>1,5</b>
		CoverDes.exe	2 199 552	911 190	41,43%	1,5
<b>anime</b>		<b>średnie(anime)</b>			<b>84,38%</b>	
	<b>small</b>	<b>średnie(small)</b>			<b>100,17%</b>	
		flaga_kuba.gif	26 836	26 874	100,14%	
		flaga_zsrr_2.gif	21 011	21 051	100,19%	
	<b>medium</b>	<b>średnie(medium)</b>			<b>86,10%</b>	
		phpbb2_admin_disallowname.swf	184 259	160 605	87,16%	
		phpbb2_admin_edituser.swf	209 232	176 907	84,55%	
		phpbb2_admin_prune.swf	160 931	139 327	86,58%	
	<b>big</b>	<b>średnie(big)</b>	<b>czas ważony</b>		<b>72,13%</b>	<b>1,70</b>
		better off bread.swf	3 091 456	2 402 357	77,71%	1,2
		class act.swf	3 837 637	2 808 023	73,17%	2
		remains to be seen.swf	4 131 201	2 705 903	65,50%	1,8
		<b>ŚREDNIA OGÓLNA</b>			<b>63,06%</b>	<b>1,75</b>

## 4.5.8 Test archiwizera RAR

kategoria	podkategoria	plik	rozmiar	RAR	RAR [%]	temp	
txt		średnie(txt)			20,05%		
	small	średnie(small)			31,41%		
		jspro.xml	10 076	2 926	29,04%		
		Ksiega13.txt	9 465	4 064	42,94%		
		Readme.htm	9 442	2 101	22,25%		
	medium	średnie(medium)			14,62%		
		index.html	32 079	4 768	14,86%		
		Informatiker0(...)+EDVStatistik.htm	93 139	7 118	7,64%		
		menu.vim	54 760	11 692	21,35%		
	big	średnie(big)		czas ważony	14,11%	3,43	
		man_page.html	648 183	57 179	8,82%	0,4	
		manual.txt	3 871 480	692 295	17,88%	4,2	
		manual.html	4 447 182	695 020	15,63%	3,2	
sfx		średnie(gfx)			53,51%		
	small	średnie(small)			66,17%		
		ringin.wav	10 026	8 242	82,21%		
		ringout.wav	5 212	2 658	51,00%		
		start.wav	1 192	895	75,08%		
		Windows XP Dymek.wav	6 400	3 610	56,41%		
	medium	średnie(medium)			42,08%		
		chord.wav	97 016	46 129	47,55%		
		flourish.mid	24 253	9 135	37,67%		
		onestop.mid	40 075	23 446	58,51%		
		town.mid	22 097	5 435	24,60%		
	big	średnie(big)		czas ważony	51,87%	0,18	
		Dźwięk (...) Windows XP.wav	190 208	78 794	41,43%	0,2	
		tada.wav	171 100	106 316	62,14%	0,1	
		Windows XP - Autostart.wav	424 644	220 966	52,04%	0,2	
	gfx		średnie(gfx)			86,48%	
		small	średnie(small)			102,20%	
8eee(...).e0fc.jpg			2 820	2 910	103,19%		
pic0downloads.png			10 894	11 026	101,21%		
sm_workstation6.gif			3 536	3 614	102,21%		
medium		średnie(medium)			84,11%		
		hol_257.jpg	44 766	44 614	99,66%		
		kpzr.jpg	99 269	99 336	100,07%		
		mizer2.gif	56 626	56 277	99,38%		
		pic-downloads.bmp	39 270	14 658	37,33%		
big		średnie(big)		czas ważony	73,93%	0,99	
		4.jpg	314 054	311 696	99,25%	0,2	
		4.png	1 193 324	1 193 388	100,01%	0,2	
	wallpaper.bmp	2 359 350	531 988	22,55%	1,5		
doc		średnie(doc)			43,70%		
	small	średnie(small)			60,49%		
		m1655108.pdf	6 625	3 952	59,65%		
		m1655116.pdf	6 776	4 078	60,18%		
		m1655122.pdf	7 181	4 426	61,63%		
	medium	średnie(medium)			26,59%		
		doc.doc	72 192	12 758	17,67%		
		Księga 13.doc	42 496	6 863	16,15%		
	vlangs.pdf	73 014	33 556	45,96%			



<b>big</b>	<b>średnie(big)</b>	<b>czas ważony</b>		<b>44,00%</b>	<b>2,12</b>
	11.ps	2 648 952	344 219	12,99%	3
	Linux Kernel.pdf	1 307 314	1 020 438	78,06%	1,5
	PHP Kompendium.doc	4 771 328	1 954 467	40,96%	1,8
<b>bin</b>	<b>średnie(bin)</b>				<b>37,67%</b>
	<b>small</b>	<b>średnie(small)</b>			<b>32,41%</b>
<b>medium</b>	hh.exe		10 752	3 485	32,41%
	<b>średnie(medium)</b>				<b>48,46%</b>
<b>big</b>	lsb_un20.exe		119 808	58 064	48,46%
	<b>średnie(big)</b>	<b>czas ważony</b>		<b>32,12%</b>	<b>2</b>
<b>anime</b>	CoverDes.exe		2 199 552	706 499	32,12%
	<b>średnie(anime)</b>				<b>77,52%</b>
	<b>small</b>	<b>średnie(small)</b>			<b>99,93%</b>
	flaga_kuba.gif		26 836	26 801	99,87%
<b>medium</b>	flaga_zsrr_2.gif		21 011	21 007	99,98%
	<b>średnie(medium)</b>				<b>73,80%</b>
	phpbb2_admin_disallowname.swf		184 259	137 397	74,57%
	phpbb2_admin_edituser.swf		209 232	155 116	74,14%
<b>big</b>	phpbb2_admin_prune.swf		160 931	117 013	72,71%
	<b>średnie(big)</b>	<b>czas ważony</b>		<b>66,30%</b>	<b>4,02</b>
	better off bread.swf		3 091 456	2 294 027	74,21%
	class act.swf		3 837 637	2 567 979	66,92%
	remains to be seen.swf		4 131 201	2 387 288	57,79%
	<b>ŚREDNIA OGÓLNA</b>			<b>55,21%</b>	<b>2,82</b>

## 4.5.9 Test archiwizera ZIP

kategoria	podkategoria	plik	rozmiar	ZIP	ZIP [%]	time
<b>txt</b>	<b>średnie(txt)</b>				<b>25,43%</b>	
	<b>small</b>	<b>średnie(small)</b>			<b>37,20%</b>	
	jspro.xml		10 076	3 458	34,32%	
	Ksiega13.txt		9 465	4 849	51,23%	
<b>medium</b>	Readme.htm		9 442	2 461	26,06%	
	<b>średnie(medium)</b>				<b>17,54%</b>	
	index.html		32 079	5 795	18,06%	
	Informatiker0(...)+EDVStatistik.htm		93 139	7 954	8,54%	
<b>big</b>	menu.vim		54 760	14 253	26,03%	
	<b>średnie(big)</b>	<b>czas ważony</b>		<b>21,55%</b>	<b>1,16</b>	
	man_page.html		648 183	66 386	10,24%	0,2
	manual.txt		3 871 480	1 096 425	28,32%	1,5
<b>sfx</b>	manual.html		4 447 182	1 160 128	26,09%	1
	<b>średnie(sfx)</b>				<b>63,64%</b>	
	<b>small</b>	<b>średnie(small)</b>			<b>66,50%</b>	
	ringin.wav		10 026	7 927	79,06%	
<b>medium</b>	ringout.wav		5 212	2 553	48,98%	
	start.wav		1 192	916	76,85%	
	Windows XP Dymek.wav		6 400	3 912	61,13%	
	<b>średnie(medium)</b>				<b>50,85%</b>	
<b>big</b>	chord.wav		97 016	73 543	75,81%	
	flourish.mid		24 253	9 565	39,44%	
	onestop.mid		40 075	24 640	61,48%	
	town.mid		22 097	5 898	26,69%	
<b>średnie(big)</b>	<b>czas ważony</b>		<b>76,88%</b>	<b>0,16</b>		

		Dźwięk (...) Windows XP.wav	190 208	132 269	69,54%	0,2
		tada.wav	171 100	136 912	80,02%	0,1
		Windows XP - Autostart.wav	424 644	344 310	81,08%	0,2
<b>gfx</b>		<b>średnie(gfx)</b>			<b>87,47%</b>	
	<b>small</b>	<b>średnie(small)</b>			<b>103,33%</b>	
		8eee(...)e0fc.jpg	2 820	2 959	104,93%	
		pic0downloads.png	10 894	11 026	101,21%	
		sm_workstation6.gif	3 536	3 672	103,85%	
	<b>medium</b>	<b>średnie(medium)</b>			<b>84,09%</b>	
		hol_257.jpg	44 766	44 551	99,52%	
		kpzr.jpg	99 269	99 276	100,01%	
		mizer2.gif	56 626	56 303	99,43%	
		pic-downloads.bmp	39 270	14 682	37,39%	
	<b>big</b>	<b>średnie(big)</b>		<b>czas ważony</b>	<b>76,11%</b>	<b>0,50</b>
		4.jpg	314 054	312 356	99,46%	0,1
		4.png	1 193 324	1 193 704	100,03%	0,4
		wallpaper.bmp	2 359 350	680 284	28,83%	0,6
<b>doc</b>		<b>średnie(doc)</b>			<b>45,91%</b>	
	<b>small</b>	<b>średnie(small)</b>			<b>61,15%</b>	
		m1655108.pdf	6 625	3 997	60,33%	
		m1655116.pdf	6 776	4 125	60,88%	
		m1655122.pdf	7 181	4 470	62,25%	
	<b>medium</b>	<b>średnie(medium)</b>			<b>29,57%</b>	
		doc.doc	72 192	15 187	21,04%	
		Księga 13.doc	42 496	8 845	20,81%	
		vlans.pdf	73 014	34 213	46,86%	
	<b>big</b>	<b>średnie(big)</b>		<b>czas ważony</b>	<b>47,02%</b>	<b>1,29</b>
		11.ps	2 648 952	438 562	16,56%	2,2
		Linux Kernel.pdf	1 307 314	1 039 870	79,54%	0,5
		PHP Kompendium.doc	4 771 328	2 144 819	44,95%	1
<b>bin</b>		<b>średnie(bin)</b>			<b>43,83%</b>	
	<b>small</b>	<b>średnie(small)</b>			<b>35,23%</b>	
		hh.exe	10 752	3 788	35,23%	
	<b>medium</b>	<b>średnie(medium)</b>			<b>55,75%</b>	
		lsb_un20.exe	119 808	66 792	55,75%	
	<b>big</b>	<b>średnie(big)</b>		<b>czas ważony</b>	<b>40,52%</b>	<b>1,4</b>
		CoverDes.exe	2 199 552	891 331	40,52%	1,4
<b>anime</b>		<b>średnie(anime)</b>			<b>83,36%</b>	
	<b>small</b>	<b>średnie(small)</b>			<b>99,61%</b>	
		flaga_kuba.gif	26 836	26 776	99,78%	
		flaga_zsrr_2.gif	21 011	20 893	99,44%	
	<b>medium</b>	<b>średnie(medium)</b>			<b>84,73%</b>	
		phpbb2_admin_disallowname.swf	184 259	157 662	85,57%	
		phpbb2_admin_edituser.swf	209 232	175 323	83,79%	
		phpbb2_admin_prune.swf	160 931	136 502	84,82%	
	<b>big</b>	<b>średnie(big)</b>		<b>czas ważony</b>	<b>71,17%</b>	<b>1,19</b>
		better off bread.swf	3 091 456	2 385 304	77,16%	1
		class act.swf	3 837 637	2 767 852	72,12%	1
		remains to be seen.swf	4 131 201	2 653 837	64,24%	1,5
		<b>ŚREDNIA OGÓLNA</b>			<b>60,31%</b>	<b>1,10</b>

#### 4.5.10 Test archiwizacji danych tekstowych

	Plik	rozmiar	7Z	7Z [%]	ACE	ACE [%]	ARJ	ARJ [%]	BZIP2	BZIP2 [%]	GZIP	GZIP [%]	LZH	LZH [%]	RAR	RAR [%]	ZIP	ZIP [%]
	średnie(txt)			19,62%		23,87%		25,80%		22,69%		25,18%		25,78%		20,05%		25,43%
<b>small</b>	średnie(small)			31,64%		37,21%		37,52%		35,48%		36,30%		36,58%		31,41%		37,20%
	jspro.xml	10 076	2 961	29,39%	3 441	34,15%	3 486	34,60%	3 406	33,80%	3 372	33,47%	3 398	33,72%	2 926	29,04%	3 458	34,32%
	Ksiega13.txt	9 465	4 099	43,31%	4 857	51,32%	4 878	51,54%	4 385	46,33%	4 758	50,27%	4 784	50,54%	4 064	42,94%	4 849	51,23%
	Readme.htm	9 442	2 098	22,22%	2 470	26,16%	2 496	26,44%	2 485	26,32%	2 376	25,16%	2 404	25,46%	2 101	22,25%	2 461	26,06%
<b>medium</b>	średnie(medium)			14,13%		16,99%		17,85%		16,42%		17,49%		17,94%		14,62%		17,54%
	index.html	32 079	4 631	14,44%	5 722	17,84%	5 939	18,51%	5 412	16,87%	5 727	17,85%	5 952	18,55%	4 768	14,86%	5 795	18,06%
	Informatiker.htm	93 139	5 920	6,36%	7 469	8,02%	7 995	8,58%	7 325	7,86%	7 947	8,53%	7 897	8,48%	7 118	7,64%	7 954	8,54%
	menu.vim	54 760	11 830	21,60%	13 756	25,12%	14 480	26,44%	13 436	24,54%	14 286	26,09%	14 673	26,80%	11 692	21,35%	14 253	26,03%
<b>big</b>	średnie(big)			13,09%		17,41%		22,04%		16,16%		21,75%		22,82%		14,11%		21,55%
	Man_page.html	648 183	43 327	6,68%	57 861	8,93%	67 844	10,47%	49 118	7,58%	68 260	10,53%	69 616	10,74%	57 179	8,82%	66 386	10,24%
		3 871					1 121				1 102		1 160				1 096	
	manual.txt	480	670 890	17,33%	880 894	22,75%	335	28,96%	839 181	21,68%	579	28,48%	555	29,98%	692 295	17,88%	425	28,32%
		4 447					1 187				1 167		1 233				1 160	
	manual.html	182	678 082	15,25%	913 543	20,54%	482	26,70%	854 952	19,22%	302	26,25%	552	27,74%	695 020	15,63%	128	26,09%

#### 4.5.11 Test archiwizacji plików audio

	Plik	rozmiar	7Z	7Z [%]	ACE	ACE [%]	ARJ	ARJ [%]	BZIP2	BZIP2 [%]	GZIP	GZIP [%]	LZH	LZH [%]	RAR	RAR [%]	ZIP	ZIP [%]
	średnie(gfx)			56,71%		52,75%		63,83%		63,22%		62,72%		73,61%		53,51%		63,64%
<b>small</b>	średnie(small)			61,08%		60,91%		66,77%		65,01%		63,70%		69,21%		66,17%		66,50%
	ringin.wav	10 026	7 785	77,65%	7 718	76,98%	7 957	79,36%	8 304	82,82%	7 840	78,20%	10 060	100,34%	8 242	82,21%	7 927	79,06%
	ringout.wav	5 212	1 919	36,82%	2 423	46,49%	2 570	49,31%	1 914	36,72%	2 470	47,39%	2 474	47,47%	2 658	51,00%	2 553	48,98%
	start.wav	1 192	890	74,66%	901	75,59%	920	77,18%	932	78,19%	830	69,63%	827	69,38%	895	75,08%	916	76,85%
	Windows XP Dymek.wav	6 400	3 531	55,17%	2 854	44,59%	3 919	61,23%	3 988	62,31%	3 813	59,58%	3 817	59,64%	3 610	56,41%	3 912	61,13%
<b>medium</b>	średnie(medium)			43,99%		42,94%		51,14%		53,40%		50,91%		58,20%		42,08%		50,85%
	chord.wav	97 016	64 094	66,07%	49 201	50,71%	73 588	75,85%	69 846	71,99%	74 273	76,56%	97 049	100,03%	46 129	47,55%	73 543	75,81%

	flourish.mid	24 253	8 606	35,48%	9 076	37,42%	9 667	39,86%	11 089	45,72%	9 484	39,10%	9 577	39,49%	9 135	37,67%	9 565	39,44%
	onestop.mid	40 075	20 862	52,06%	23 175	57,83%	24 832	61,96%	25 650	64,00%	24 701	61,64%	26 758	66,77%	23 446	58,51%	24 640	61,48%
	town.mid	22 097	4 942	22,37%	5 696	25,78%	5 939	26,88%	7 045	31,88%	5 821	26,34%	5 854	26,49%	5 435	24,60%	5 898	26,69%
<b>big</b>	<b>średnie(big)</b>			<b>67,85%</b>		<b>54,96%</b>		<b>76,85%</b>		<b>73,92%</b>		<b>77,17%</b>		<b>100,02%</b>		<b>51,87%</b>		<b>76,88%</b>
	Dźwięk (...).wav	190 208	108 864	57,23%	90 435	47,55%	132 092	69,45%	125 446	65,95%	133 238	70,05%	190 271	100,03%	78 794	41,43%	132 269	69,54%
	tada.wav	171 100	126 794	74,11%	104 016	60,79%	136 843	79,98%	131 109	76,63%	137 395	80,30%	171 132	100,02%	106 316	62,14%	136 912	80,02%
	Windows XP (...).wav	424 644	306 648	72,21%	240 114	56,54%	344 470	81,12%	336 285	79,19%	344 633	81,16%	424 694	100,01%	220 966	52,04%	344 310	81,08%

#### 4.5.12 Test archiwizacji plików graficznych

	plik	rozmiar	7Z	7Z [%]	ACE	ACE [%]	ARJ	ARJ [%]	BZIP2	BZIP2 [%]	GZIP	GZIP [%]	LZH	LZH [%]	RAR	RAR [%]	ZIP	ZIP [%]
	<b>średnie(gfx)</b>			<b>86,27%</b>		<b>85,36%</b>		<b>87,61%</b>		<b>88,77%</b>		<b>86,76%</b>		<b>87,07%</b>		<b>86,48%</b>		<b>87,47%</b>
<b>small</b>	<b>średnie(small)</b>			<b>102,60%</b>		<b>102,46%</b>		<b>103,42%</b>		<b>110,36%</b>		<b>100,86%</b>		<b>101,24%</b>		<b>102,20%</b>		<b>103,33%</b>
	8e(...)e0fc.jpg	2 820	2 912	103,26%	2 916	103,40%	2 956	104,82%	3 214	113,97%	2 846	100,92%	2 880	102,13%	2 910	103,19%	2 959	104,93%
	pic0do(...).png	10 894	11 108	101,96%	10 999	100,96%	11 036	101,30%	11 373	104,40%	10 937	100,39%	10 935	100,38%	11 026	101,21%	11 026	101,21%
	sm_workstation6.gif	3 536	3 627	102,57%	3 643	103,03%	3 682	104,13%	3 985	112,70%	3 581	101,27%	3 579	101,22%	3 614	102,21%	3 672	103,85%
<b>medium</b>	<b>średnie(medium)</b>			<b>83,05%</b>		<b>82,09%</b>		<b>84,37%</b>		<b>83,68%</b>		<b>84,17%</b>		<b>84,51%</b>		<b>84,11%</b>		<b>84,09%</b>
	hol_257.jpg	44 766	44 845	100,18%	44 719	99,90%	44 644	99,73%	44 832	100,15%	44 857	100,20%	44 801	100,08%	44 614	99,66%	44 551	99,52%
	kpzr.jpg	99 269	99 867	100,60%	99 365	100,10%	99 291	100,02%	99 446	100,18%	99 109	99,84%	99 301	100,03%	99 336	100,07%	99 276	100,01%
	mizer2.gif	56 626	56 376	99,56%	56 494	99,77%	56 277	99,38%	56 975	100,62%	56 164	99,18%	56 660	100,06%	56 277	99,38%	56 303	99,43%
	pic-downloads.bmp	39 270	12 507	31,85%	11 225	28,58%	15 057	38,34%	13 266	33,78%	14 712	37,46%	14 879	37,89%	14 658	37,33%	14 682	37,39%
<b>big</b>	<b>średnie(big)</b>			<b>74,23%</b>		<b>72,61%</b>		<b>76,12%</b>		<b>73,98%</b>		<b>76,12%</b>		<b>76,31%</b>		<b>73,93%</b>		<b>76,11%</b>
	4.jpg	314 054	313 755	99,90%	312 373	99,46%	312 424	99,48%	313 865	99,94%	312 213	99,41%	314 083	100,01%	311 696	99,25%	312 356	99,46%
	4.png	1 193	1 195	100,15%	1 199	100,50%	1 193	100,01%	1 194	100,12%	1 193	99,99%	1 193	100,00%	1 193	100,01%	1 193	100,03%
	wallpaper.bmp	324 359	534 256	22,64%	421 149	17,85%	681 118	28,87%	739	22,8%	258	7,9%	353	10,9%	388	11,9%	704	21,7%

#### 4.5.13 Test archiwizacji dokumentów

	plik	rozmiar	7Z	7Z [%]	ACE	ACE [%]	ARJ	ARJ [%]	BZIP2	BZIP2 [%]	GZIP	GZIP [%]	LZH	LZH[%]	RAR	RAR [%]	ZIP	ZIP [%]
	średnie(doc)			41,87%		44,69%		46,19%		46,11%		45,69%		48,16%		43,70%		45,91%
<b>small</b>	średnie(small)			57,48%		60,59%		61,46%		65,77%		59,98%		60,09%		60,49%		61,15%
	m1655108.pdf	6 625	3 750	56,60%	3 956	59,71%	4 020	60,68%	4 330	65,36%	3 915	59,09%	3 926	59,26%	3 952	59,65%	3 997	60,33%
	m1655116.pdf	6 776	3 869	57,10%	4 084	60,27%	4 147	61,20%	4 431	65,39%	4 047	59,73%	4 053	59,81%	4 078	60,18%	4 125	60,88%
	m1655122.pdf	7 181	4 217	58,72%	4 436	61,77%	4 489	62,51%	4 779	66,55%	4 389	61,12%	4 394	61,19%	4 426	61,63%	4 470	62,25%
<b>medium</b>	średnie(medium)			25,21%		28,90%		29,72%		27,43%		29,81%		29,88%		26,59%		29,57%
	doc.doc	72 192	12 086	16,74%	15 227	21,09%	15 378	21,30%	13 751	19,05%	15 237	21,11%	15 692	21,74%	12 758	17,67%	15 187	21,04%
	Księga 13.doc	42 496	6 824	16,06%	8 609	20,26%	8 880	20,90%	7 759	18,26%	8 813	20,74%	8 806	20,72%	6 863	16,15%	8 845	20,81%
	vlans.pdf	73 014	31 267	42,82%	33 113	45,35%	34 298	46,97%	32 851	44,99%	34 745	47,59%	34 457	47,19%	33 556	45,96%	34 213	46,86%
<b>big</b>	średnie(big)			42,92%		44,58%		47,39%		45,12%		47,28%		54,50%		44,00%		47,02%
	11.ps	2 648																
		952	278 169	10,50%	351 433	13,27%	450 441	17,00%	319 180	12,05%	451 561	17,05%	464 441	17,53%	344 219	12,99%	438 562	16,56%
		1 307	1 010		1 022		1 043		1 018		1 040		1 307		1 020		1 039	
	Linux Kernel.pdf	314	472	77,29%	764	78,23%	431	79,81%	672	77,92%	510	79,59%	354	100,00%	438	78,06%	870	79,54%
	PHP	4 771	1 954		2 015		2 163		2 165		2 157		2 193		1 954		2 144	
	Kompendium.doc	328	912	40,97%	284	42,24%	373	45,34%	952	45,40%	258	45,21%	478	45,97%	467	40,96%	819	44,95%

#### 4.5.14 Test archiwizacji plików wykonywalnych

	plik	rozmiar	7Z	7Z [%]	ACE	ACE [%]	ARJ	ARJ [%]	BZIP2	BZIP2 [%]	GZIP	GZIP [%]	LZH	LZH [%]	RAR	RAR [%]	ZIP	ZIP [%]
	średnie(bin)			35,38%		38,20%		44,04%		44,34%		43,65%		43,99%		37,67%		43,83%
<b>small</b>	średnie(small)			31,59%		33,43%		35,54%		38,13%		34,56%		34,78%		32,41%		35,23%
	hh.exe	10 752	3 397	31,59%	3 594	33,43%	3 821	35,54%	4 100	38,13%	3 716	34,56%	3 740	34,78%	3 485	32,41%	3 788	35,23%
<b>medium</b>	średnie(medium)			46,07%		48,43%		55,74%		55,89%		55,75%		55,76%		48,46%		55,75%
	lsb_un20.exe	119 808	55 194	46,07%	58 024	48,43%	66 779	55,74%	66 962	55,89%	66 789	55,75%	66 799	55,76%	58 064	48,46%	66 792	55,75%
<b>big</b>	średnie(big)			28,47%		32,73%		40,84%		39,00%		40,64%		41,43%		32,12%		40,52%
	CoverDes.exe	2 199 552	626 322	28,47%	719 872	32,73%	898 242	40,84%	857 879	39,00%	893 872	40,64%	911 190	41,43%	706 499	32,12%	891 331	40,52%

#### 4.5.15 Test archiwizacji sekwencji animowanych

	plik	rozmiar	7Z	7Z [%]	ACE	ACE [%]	ARJ	ARJ [%]	BZIP2	BZIP2 [%]	GZIP	GZIP [%]	LZH	LZH [%]	RAR	RAR [%]	ZIP	ZIP [%]
	średnie(anime)			77,66%		77,51%		83,98%		82,76%		83,37%		84,38%		77,52%		83,36%

<b>small</b>	<b>średnie(small)</b>			<b>100,38%</b>	<b>100,15%</b>		<b>100,02%</b>		<b>101,44%</b>		<b>99,43%</b>		<b>100,17%</b>		<b>99,93%</b>		<b>99,61%</b>	
	flaga_kuba.gif	26 836	26 916	100,30%	26 854	100,07%	26 819	99,94%	27 178	101,27%	26 685	99,44%	26 874	100,14%	26 801	99,87%	26 776	99,78%
	flaga_zsrr_2.gif	21 011	21 106	100,45%	21 060	100,23%	21 032	100,10%	21 347	101,60%	20 890	99,42%	21 051	100,19%	21 007	99,98%	20 893	99,44%
<b>medium</b>	<b>średnie(medium)</b>			<b>73,13%</b>	<b>73,81%</b>		<b>85,58%</b>		<b>80,74%</b>		<b>84,67%</b>		<b>86,10%</b>		<b>73,80%</b>		<b>84,73%</b>	
	phpbb2_(...)name.swf	184 259	135 952	73,78%	137 273	74,50%	159 735	86,69%	149 721	81,26%	157 441	85,45%	160 605	87,16%	137 397	74,57%	157 662	85,57%
	phpbb2_admin_edituser.swf	209 232	153 912	73,56%	155 285	74,22%	176 610	84,41%	169 631	81,07%	175 282	83,77%	176 907	84,55%	155 116	74,14%	175 323	83,79%
	phpbb2_admin_prune.swf	160 931	115 943	72,05%	117 010	72,71%	137 840	85,65%	128 562	79,89%	136 455	84,79%	139 327	86,58%	117 013	72,71%	136 502	84,82%
<b>big</b>	<b>średnie(big)</b>			<b>67,05%</b>	<b>66,12%</b>		<b>71,68%</b>		<b>72,32%</b>		<b>71,35%</b>		<b>72,13%</b>		<b>66,30%</b>		<b>71,17%</b>	
	better off bread.swf	3 091	2 520		2 295		2 393		2 443		2 387		2 402		2 294		2 385	
		456	526	81,53%	640	74,26%	331	77,42%	817	79,05%	771	77,24%	357	77,71%	027	74,21%	304	77,16%
		3 837	2 490		2 560		2 782		2 807		2 774		2 808		2 567		2 767	
	class act.swf	637	320	64,89%	365	66,72%	106	72,50%	898	73,17%	848	72,31%	023	73,17%	979	66,92%	852	72,12%
		4 131	2 261		2 370		2 691		2 674		2 665		2 705		2 387		2 653	
	remains to be seen.swf	201	247	54,74%	310	57,38%	122	65,14%	911	64,75%	439	64,52%	903	65,50%	288	57,79%	837	64,24%

## 5 Interpretacja wyników

Przeprowadzone testy pokazały dość spore zróżnicowanie efektywności kompresji porównywanych metod, zarówno pod względem objętościowym, jak i czasowym. Różne wyniki uzyskano również przy kompresji zbiorów danych różnych typów. Pewną uwagę należy zwrócić na parametry czasowe rozważanych metod. W niektórych wypadkach rozbieżności okazały się duże, co zmusza do zastanowienia się nad wpływem samego narzędzia na wynik testu. Testy metod ARJ, GZIP i BZIP2 przeprowadzono za pomocą narzędzi działających w trybie konsolowym (dostępne inne narzędzia oferowały co prawda kompresję do tych formatów, aczkolwiek proces ten powodował utworzenie pliku o rozszerzeniu tar.bz2 (tar.gz), co oznaczało, że finalna kompresja odbywała się na archiwum TAR, którego utworzenie również wymaga czasu i zaburzałoby to pomiary czasu, jak i rozmiaru (szczególnie dla danych z kategorii „small” – archiwum TAR wprowadza też swoje nagłówki)). Reszta formatów była testowana narzędziami „okienkowymi”, co implikowało pewne (aczkolwiek minimalne) narzuty czasowe na bardzo dynamiczną „ekspozycję” przebiegu kompresji. Ponieważ jednak za priorytet uznano stopień upakowania, a wpływ struktury aplikacji szacuje się na niewielki, zatem problem ten jedynie zasygnalizowano, nie przywiązując do niego krytycznej wagi. W testach zanotowano również pewne niespodzianki – pliki w formacie Macromedia Flash (\*.swf), pomimo wewnętrznego zastosowania kompresji Sorenson, wykazały pewną podatność na kompresję. Ogólnie (na podstawie uogólnionych wyników – niebieskie pola na dole tabeli 4.5.2 – 4.5.9), nie można jednoznacznie stwierdzić zależności między efektywnością upakowania a czasem jej trwania. Wynika to też w niewielkim stopniu z zasygnalizowanego wcześniej problemu. Jednak dużo większy wpływ ma tutaj zróżnicowanie charakterystyki strukturalnej, oraz informacyjnej różnych typów danych. Jednak, na podstawie analiz ogólnych stopni upakowania można wyróżnić grupę trzech wiodących metod kompresji – 7ZIP, ACE i RAR. Wszystkie trzy charakteryzują się podobnym upakowaniem ok. 55%, i wszystkie były testowane narzędziami okienkowymi, co pozwala w miarę dokładnie wnioskować o ich charakterystyce czasowej. Osiągnięte dobre rezultaty można wytłumaczyć z jednej strony innowacyjnym podejściem do problemu kompresji (format 7Zip), jak i użyciem specyficznych zorientowanych na typ danych metod wspomagających kompresję (formaty ACE i RAR). Zwłaszcza, że w ogólnym stopniu kompresji mają znaczny udział duże zbiory (powyżej 1MB), a są to przeważnie pliki, których kompresja jest tymi specjalnymi metodami wspomagana (przez metody RAR i 7Zip). Ponadto, wiodące metody w swoim zrębie wykorzystują metodę LZ77 lub jej modyfikacje, co pozwala wysnuć (aczkolwiek raczej ostrożny) wniosek na temat uniwersalności zastosowania tej metody w kompresji danych w ogóle (jeśli nawet nie ona sama, to jej modyfikacje okazują się być dobre do kompresji danych bez względu na ich rodzaj).

### 5.1 Efektywność poszczególnych metod kompresji

Jedną z oczywistych różnic w wynikach osiąganych przez różne archiwizery są zastosowane w nich algorytmy kompresji. Należy tu zauważyć, że, mimo iż większość metod bazuje na metodach typu LZ\* (LZ77, LZ78, LZW), oraz ich modyfikacjach, to osiągają one różne wyniki zarówno pod względem stopnia upakowania, jak i czasu trwania procesu kompresji.

## Metody oparte o algorytmy LZ77

Algorytm kompresji LZ77 i jego modyfikacje stanowiły największy procent metod reprezentowanych w teście. Wykorzystywane one są przez archiwizery RAR, ARJ (modyfikacja LZSS), ACE (inna nieznaną z nazwy modyfikacja LZ77), 7-Zip (algorytm LZMA), oraz ZIP i LZH (modyfikacja o nazwie LZH). Należy stwierdzić, że wśród tej grupy znalazł się najlepszy z ogólnego punktu widzenia archiwizery RAR (stopień upakowania 55,21%, średni ważony czas kompresji 2,82s), oraz inna wiodąca metoda ACE (odpowiednio: 55,71%; 3,46s). Metody te bardzo dobrze radziły sobie z kompresją tekstów, zwłaszcza o dużej objętości (ACE – 17,41%/5,53s; RAR – 14,11%/3,43s) ustępując jednak innowacyjnej metodzie 7-Zip. Przy tak dobranej rozmiarze tekstu, mając na uwadze otrzymane wyniki, można stwierdzić, że archiwizery RAR, wydaje się efektywniej pakować dane tekstowe, niż zaproponowana przez twórców ACE archiwizery. Należy tu mieć na uwadze, że archiwizery RAR oferuje dodatkowe mechanizmy wspierające kompresję tekstu. Aby uzmysłowić siłę oraz znaczenie zastosowania tychże mechanizmów, przeprowadzono dodatkowe testy kompresji tych danych (dla tekstów „big”), już bez użycia metod wspomagających (w archiwizeryze RAR w kontrolce „Compression” zaznaczono metodę „Text compression” jako „disabled”). Wyniki pomocniczych testów przedstawia poniższa tabelka:

plik [big]	rozmiar	RAR [wspomaganie]	RAR [bez wspomaganie]	zysk [%]
man_page.html	648 183	57 179	57 650	0,82%
manual.txt	3 871 480	692 295	870 273	20,45%
manual.html	4 447 182	695 020	905 217	23,22%

Jak widać, zastosowanie metod wspomagających w archiwizeryze RAR przyczyniło się – zwłaszcza dla zbiorów o dużej wolumetrii, do pokaźnej redukcji objętości spakowanego zbioru. Kompresja „zwykła” przebiegała też zauważalnie wolniej. Archiwizery ACE nie wspiera kompresji tekstów, zatem widać, że wynik uzyskany poprzez zmodyfikowany na potrzeby ACE algorytm LZ77 okazał się lepszy niż zastosowany przez RAR surowy algorytm LZSS. Przyjrzyjmy się również wynikom osiągniętym przez innego reprezentanta metody LZSS – archiwizery ARJ. W kompresji tekstów (uwzględniając już cały zbiór testowanych plików tekstowych) okazał się on gorszy od „surowego” RAR’a.

Mając na uwadze wpływ metod wspomagających na ogólny wynik uzyskiwany przez archiwizery, dokonamy teraz zestawienia skuteczności archiwizerów ACE oraz RAR (a także porównania z pakierem ARJ), pracujących w trybie „surowym”, dla danych typu audio, plików graficznych oraz plików wykonywalnych.

kat	Plik	rozmiar	ACE	ACE#	Zysk	RAR	RAR#	zysk	ARJ
sfx	ringin.wav	10 026	7 718	7 986	3,35%	8 242	8 242	0%	7 957
	ringout.wav	5 212	2 423	2 196	-10,33%	2 658	2 225	-19,46%	2 570
	Windows XP –dymek.wav	6 400	2 854	3 926	27,30%	3 610	3 580	-0,83%	3 919
	chord.wav	97 016	49 201	74 441	33,90%	46 129	72 415	36,29%	73 588
	town.mid	22 097	5 696	5 368	-6,11%	5 435	5 435	0%	5 939
	Dźwięk(...)Windows XP.wav	190 208	90 435	132 859	31,93%	78 794	121 325	35,05%	132 092
	tada.wav	171 100	104 016	138 096	24,67%	106 316	132 221	19,59%	136 843
	Windows XP Autostart.wav	424 644	240 114	346 222	30,64%	220 966	332 415	33,52%	344 470
gfx	pic-downloads.bmp	39 270	11 225	14 505	22,61%	14 658	14 573	-0,58%	15 057
	4.jpg	314 054	312 373	313 445	0,34%	311 696	312 692	3,18%	312 424
	4.png	1 193 324	1 199 293	1 193 417	-0,49%	1 193 388	1 193 388	0%	1 193 444
	wallpaper.bmp	2 359 350	421 149	631 817	33,34%	531 988	632 261	15,85%	681 118
bin	CoverDes.exe	2 199 552	719 872	781 072	7,83%	706 499	782 456	9,70%	898 242



## LEGENDA:

**achiwizer** – wynik uzyskany przez użycie archiwizera z aktywnym wspomaganie kompresji określonych typów danych

**archiwizer#** -wynik uzyskany przez użycie archiwizera bez aktywnego wspomaganie kompresji określonych typów danych

**zysk** – wielkość 1 – (archiwizer/archiwizer#)

Jak widać, dla wiodących metod bazujących na algorytmie LZ77, zastosowaniu wspomaganie dla poszczególnych typów danych powoduje zróżnicowanie wyników rozpatrywanych pakierów zarówno w jedną, jak i w drugą stronę (świadczą o tym ujemne wartości zysku przy zastosowaniu mechanizmów wspomaganie kompresji). Analizując wyniki zamieszczone w powyższej tabelce można wywnioskować, że wiodące metody bazujące na odmianie algorytmu LZ77 nie różnią się tak bardzo przy kompresji w trybie „surowym” – różnice w objętości spakowanych zbiorów wynoszą przeważnie kilka kilobajtów (wyjątek stanowi testowany plik wykonywalny, oraz plik bitmapowy „wallpaper.bmp”). W ostatecznym rozrachunku wnioskujemy, że formaty RAR oraz ACE zawdzięczają swoją wysoką efektywność specyficznym dla nich mechanizmom wspomagającym zmniejszanie objętości poszczególnych typów danych. Okupione jest to jednak wydłużeniem czasu kompresji. Tu na pierwsze miejsce wysuwa się ARJ, nie oferujący tak wyrafinowanych dodatków. Mimo, iż spakowane przez niego zbiory są trochę większe, może śmiało rywalizować z ACE i RAR'em w kompresji plików graficznych oraz dokumentów – szczególnie, gdy zależy nam na czasie.

Przyjrzyjmy się z kolei innym reprezentantom algorytmów wywodzących się z LZ77-formatowi LZH. Osiągnął on ogólnie najgorszy wynik ze wszystkich testowanych archiwizerów (63,06%), jednak okazał się ogólnie szybszy od metod RAR i ACE (średni czas ważony 1,75s), lecz nie prześcignął on ARJ'a (prze grywa z nim w każdej kategorii zarówno pod względem wolumetrii zbiorów, jak i czasu trwania kompresji). Dlatego używanie go do kompresji jakichkolwiek danych nie jest wskazane, gdyż istnieją znacznie lepsze i szybsze (oraz popularniejsze) metody.

Przeanalizujemy teraz innowacyjny format 7-Zip. W ogólnym rozrachunku otrzymał on drugie miejsce pod względem kompresji (średnio spakowane przez niego pliki zajmowały 55,35% rozmiaru początkowego). Jednak średni czas kompresji okazał się być najdłuższy ze wszystkich, choć przy przetwarzaniu poszczególnych typów danych zdarzały mu się odstępstwa, np. pliki audio pakował szybciej niż pakery ACE, ARJ oraz RAR. Bezapelacyjnie najlepszy okazał się jednak przy przetwarzaniu danych tekstowych, co zawdzięcza wykorzystaniu metod z rodziny PPM. Spakowane nim pliki tekstowe zajmowały średnio 19,62% rozmiaru oryginalnego. Dla przykładu najlepszy RAR osiągnął 20,05%, co pozwala wnioskować, że stosowane przez niego algorytmy wspomaganie kompresji tekstu bazują również na rozwiązaniu PPM. 7-Zip okazał się także najlepszy w zakresie kompresji plików wykonywalnych (dla których to dostarcza metod wspomagających BCJ i BCJ2), i tylko minimalnie (o setne części procenta) gorszy od RAR'a w pakowaniu sekwencji animowanych. Archiwizer 7-Zip okazał się najlepszy pod względem kompresji dokumentów (wszystkich typów występujących w teście), w kompresji grafiki zajął drugie miejsce (po ACE), będąc minimalnie lepszym od RAR'a, a sekwencję animowane pakował praktycznie na równi ze swoimi konkurentami (różnice w zakresie setnych części procenta). Jeśli chodzi o pliki audio, to 7-Zip najlepiej ze wszystkich kompresował pliki \*.mid (reprezentujące tony), gorzej natomiast (w porównaniu z ACE i RAR) radził sobie w kompresji plików \*.wav

(zawierające reprezentację fal dźwiękowych). Wynika to ze wspomnianego wcześniej faktu wspomaganego przez konkurencję pakowania plików audio odrębnymi metodami. Gdyby porównać go z konkurencją pracującą w trybie surowym, wysunąłby się on na pierwsze miejsce. Jako, że 7-Zip jest rozwiązaniem raczej nowym, i w dalszym ciągu rozwijanym, zasługuje na szczególne zainteresowanie, gdyż wyniki przezeń osiągnięte są bardzo przyzwoite, aczkolwiek uzyskany czas działania zdaje się zaprzeczać zapewnieniom twórców, że kompresuje on dane do 20 razy szybciej niż inne metody. Gdyby nie fakt, że konkurenci wspierają dodatkowo kompresję niektórych formatów danych, zająłby on zapewne pierwsze miejsce w teście, co wskazuje na trafny kierunek opracowania algorytmu LZMA (oraz zastosowania metod PPMd), obrany przez autorów.

Kolejnym archiwizytem, bazującym na metodzie LZ77 jest GZIP. Okazał się on być najszybszym archiwizytem w teście (średni czas ważony 0,45s), za to średnia wydajność kompresji na poziomie 59,87% nie pozwala zaliczyć go do metod dobrych (w porównaniu z innymi). Przyczyn tego można upatrywać w wykorzystaniu surowej i mało modyfikowanej metody LZ77 (algorytm deflate), oraz w podziale strumienia wejściowego na bloki, co ogranicza możliwość uwzględnienia charakterystyki słownikowej większej ilości danych. A ponieważ do każdego bloku jest dodawana opisująca go informacja (np. drzewo Huffmana), efektywna wydajność kompresji jeszcze spada. Warto zaznaczyć tutaj, że GZIP pakuje pliki tekstowe minimalnie lepiej od ogólnie wydajniejszego ARJ'a, przy zaoszczędzeniu czasu zużytego na ten proces. Minimalnie wygrywa z ARJ'em również w kompresji dźwięku, zwłaszcza średnich plików \*.wav, a także plików z grafiką, plików binarnych, oraz sekwencji animowanych.

Innym reprezentantem metod opartych o algorytm deflate w naszym teście był archiwizyter ZIP. Paker ten oferuje dosyć szybką kompresję (średni czas ważony 1,10s), choć nie tak szybką jak omawiany wcześniej program GZIP. Niestety, osiągnięta przez niego wydajność (na poziomie 60,31%), plasuje go raczej na końcu listy w rankingu (gorszy był tylko LZH). Osiągnął on także minimalnie gorszą efektywność niż GZIP. Przyczyny tej różnicy wydają się być proste – format pliku ZIP, z racji chociażby możliwości kompresji katalogów oraz podziału skompresowanego zbioru na porcje, musi udostępniać więcej metadanych, zawartych w nagłówkach. Do takiego wniosku skłaniają obserwowane różnice w rozmiarach plików skompresowanych metodą GZIP oraz ZIP- w większości przypadków wynosiły one ok. 50 – 100 bajtów, na korzyść GZIP'a (dane dla plików „small”). Jednak w kompresji dokumentów, oraz sekwencji animowanych ZIP i jego „brat” GZIP są porównywalne- czasem z lekką przewagą na korzyść ZIP'a (różnice ok. 1%).

## **Metoda oparta za BWT**

Jedynym wyłamującym się z ogólnie stosowanych odmian algorytmu LZ77 archiwizytem był open-source'owy BZIP2. Jądem tej metody jest transformata Burrowsa – Wheelera, wspomaganą na wyjściu algorytmem Move – to – Front. Końcowym etapem jest zastosowanie kodowania Huffmana. Takie podejście dało (w teorii) lepszą efektywność od innego powszechnie stosowanego otwartego pakera GZIP, jednak, zgodnie z regułą, proces archiwizacji metodą BZIP2 zajmuje więcej czasu. I rzeczywiście, średni czas ważony wynosi 2,03s i jest kilkakrotnie dłuższy niż ten sam parametr dla GZIP'a. BZIP2 Okazał się lepszy od GZIP'a w kompresji tekstów (średnio 22,69% vs. 25,18% osiągnięte przez GZIP'a). Co godne odnotowania, dla zestawu danych testowych, kompresja tekstu przebiegła lepiej zarówno pod względem objętościowym, jak i czasowym, niż dla archiwizera ACE. Niestety, dla pozostałych zbiorów danych użytych w teście, nie znaleziono potwierdzenia dla

powszechnej opinii o lepszej efektywności objętościowej i gorszej wydajności czasowej typowej dla metody BZIP2, w porównaniu do archiwizera GZIP. Należy to uznać za słaby punkt testu.

## **5.2 Podatność różnych typów danych na kompresję**

Zgodnie z naszymi oczekiwaniami danymi najbardziej podatnymi na kompresję są pliki zawierające czysty tekst. Ze statystycznego dużego takiego pliku, po zastosowaniu kompresji, zostało tylko ok. 17 - 18% rozmiaru oryginalnego (przy użyciu pakera 7-Zip w trybie PPMd, lub RAR'a ze wspomaganie). Przeciętna metoda surowa, oparta o algorytm serii LZ\*, oferuje stopień upakowania na poziomie od 22% (ACE, RAR) do 30% (LZH).

Nieco trudniej kompresuje się pliki tekstowe zawierające w sobie formatowanie. Spowodowane jest to większym zróżnicowaniem symboli wejściowych. Dane te spowodowały duży rozrzut wyników kompresji wśród testowanych metod (od 15% do 27%).

Typem plików sprawiającym najwięcej problemów są obrazy. W ich wypadku czasem plik wynikowy był większy niż plik wejściowy. Wynika to z obecności kompresji w plikach formatu JPEG lub GIF. Działanie archiwizera ogranicza się w tym momencie do uzupełniania pliku wynikowego o swoje nagłówki. Z tego typu danych udało się obciąć tylko ok. 14% oryginalnego rozmiaru. Bardzo dobrej kompresji uległy natomiast pliki zapisane w formacie BMP zawierające dane albo nieskompresowane albo skompresowane za pomocą prostej metody RLE- tu udało się uzyskać efektywność na poziomie porównywalnym z kompresją czystego tekstu.

Kolejnymi typami plików, bardziej podatnych na kompresję, są pliki audio. Średni stopień upakowania tychże zbiorów oscylował w granicach 52 – 64 procent. Jednak podanie średniego prawdopodobnego poziomu wydajności jest tutaj nieuprawnione- kompresja plików audio dla jednego archiwizera charakteryzowała się znacznym rozrzutem efektywności, przy kompresji tego samego formatu danych (\*.wav).

Jak już wcześniej wspomniano, dużą niespodzianką okazały się pliki animowane. Spodziewano się minimalnej efektywności pakownia tychże zbiorów, z powodu wymyślnych metod kompresji już dla nich zastosowanych, głównie na potrzeby ich transmisji przez Internet (pliki \*.swf z kompresją Sorenson). Okazało się jednak, że z takich plików można „uszczać” nawet 33% ich rozmiaru początkowego- wyniki takie osiągnięto archiwizerni 7-Zip, oraz ACE i RAR ze wspomaganie. Ukazuje to raz jeszcze siłę algorytmu LZMA, zastosowaną w 7-Zip – osiągnęła ona wydajność porównywalną z wiodącymi metodami konkurencyjnymi, przy zastosowanych dodatkowych mechanizmach polepszających kompresję.

Pliki wykonywalne należą do kategorii tych, które w wyniku działania archiwizera udaje się upakować do rozmiaru będącego średnio około 40% rozmiaru oryginalnego. Jest to wynik zbliżony do tego, jaki uzyskały pliki zawierające dokumenty sformatowane.

Zaobserwowane zależności potwierdzają opinie, że popularne archiwizery głównie stosuje się do kompresowania tekstów lub programów. Te typy danych były najbardziej podane na działanie pakerów, które poddaliśmy naszemu testowi.

### **5.3 Zależność między typem danych a metodami kompresji**

Przeprowadzone testy wykazały również pewne zróżnicowanie między zastosowaną do kompresji metodą, a typem kompresowanych danych. Mimo ogólnej tendencji, że im lepszy archiwizator, tym lepiej pakuje on różne typy danych, na podstawie danych testowych udało się zauważyć pewne zależności pozwalające na sformułowanie pewnych wskazówek odnośnie zastosowania poszczególnych archiwizatorów.

#### **Dane tekstowe**

Tu bezapelacyjnym liderem okazał się 7-Zip. Zawdzięcza on to zastosowaniu metody probabilistycznej PPMd, która okazuje się bardzo dobrze redukować ilość miejsca zajmowanego przez informację zawartą w tekście (do ok. 13% dla dużych zbiorów). Prawie tak samo dobry okazał się RAR (duże pliki tekstowe zmniejszał średnio o ok. 85%), dzięki zastosowaniu specjalnych mechanizmów polepszających kompresję tekstów, które, jak ostrożnie wywnioskowano wcześniej, mogą również bazować na rodzinie PPM. Również bardzo przyzwoite rezultaty osiąga kombinacja algorytmów Burrowsa – Wheelera i Move – To – Front, zaimplementowana w archiwizatorze BZIP2 (średni stopień upakowania długich tekstów wyniósł 16,16%). Pozostałe metody, bazujące na algorytmach opartych o LZ77, redukowały duże zbiory tekstowe średnio do ok. 22% ich pierwotnej objętości. Pliki średnie (o rozmiarze od 10 do 100 kilobajtów) są pakowane do średnio 16 – 17% ich początkowego rozmiaru przez prawie wszystkie metody (wyjątkiem są 7-Zip i RAR, które osiągnęły wynik ok. 14%), a pliki małe (do 10K) są redukowane do średnio 35 – 37% objętości (znów za wyjątkiem 7 – Zip i RAR – tu osiągnięto ok. 31%).

#### **Pliki audio**

Tu najlepszymi rezultatami (ok. 53%) pochwalić się mogą metody oferujące specjalne wspomaganie kompresji dźwięku – ACE oraz RAR. Tuż za nimi jest 7-Zip (56,71%), pozostałe archiwizery osiągnęły wynik powyżej 60%.

#### **Grafika**

W przypadku tych danych, głównie ze względu na powszechnie stosowaną w plikach graficznych kompresję, wszystkie archiwizery osiągnęły stopień upakowania oscylujący w okolicach ok. 85%. Wspomnianym wcześniej wyjątkiem są bitmapy, które ze względu na stosunkowo niski stopień przetworzenia dadzą się skompresować do poziomu poniżej 40% rozmiaru oryginalnego, przy czym najkorzystniej jest użyć wtedy pakera ACE.

#### **Dokumenty**

Statystycznie najlepszym archiwizorem kompresującym pliki dokumentów okazał się 7-Zip. Średni stopień upakowania przetworzonych przez niego danych wyniósł 41,87%. Minimalnie gorszy okazał się RAR (43,7%). Dokumenty przechowujące tekst w postaci sformatowanej (\*.ps, \*.doc) wykazywały daleko lepsze stopnie upakowania, natomiast nie zaleca się używania metody BZIP2 do kompresji plików \*.pdf – rezultaty uzyskane dla średniej wielkości dokumentów tego rodzaju okazały się być najgorsze ze wszystkich uzyskanych w teście.

## Pliki binarne

W tej kategorii najlepszym archiwizерem okazał się być 7-Zip (średnio 35,38%). Drugie miejsce zajął RAR (37,67%), a potem ACE (38,20%). Pozostałe pakery uzyskały wynik powyżej 40%.

## Sekwencje animowane

Tu wiodącymi archiwizерami okazały się być trzy dominujące w całym teście programy – 7-Zip, ACE oraz RAR. Ich średnie współczynniki upakowania wynosiły ok. 77%. Pozostałe metody legitymują się rezultatami powyżej 80%, przy czym wypada zauważyć, że najlepszym w stawce gorszych archiwizерów okazał się BZIP2 (82,76%), który przy plikach animowanych z podkategoria „big” okazał się porównywalny z resztą niewiodących pakерów (najefektywniejsze metody 7-Zip, ACE, RAR mogły „pochwalić” się oszczędnością 5% oryginalnego rozmiaru pliku w porównaniu do pozostałych metod).

## 6 Podsumowanie

Zadaniem niniejszego testu było porównanie popularnych i powszechnie stosowanych programów archiwizujących dane, oraz omówienie różnic między nimi, oraz algorytmów i mechanizmów, będących podstawą ich działania. Pomimo ograniczonego zbioru danych testowych, oraz zauważonej „pięty Achillesowej” testu w postaci odbiegających od standardowo przyjętych opinii wyników archiwizera BZIP2, udało się wykazać zróżnicowanie testowanych metod zarówno pod względem ich zdolności do redukcji rozmiaru zbiorów wynikowych, jak i czasu trwania przetwarzania danych wejściowych. Testy potwierdziły siłę nowego rozwiązania 7-Zip, oraz wpływ dodatkowych oferowanych przez różne metody mechanizmów wspomagających kompresję na efektywność objętościową tego procesu.

## Bibliografia

<http://www.datacompression.info> – Witryna na temat kompresji danych

<http://pl.wikipedia.org/wiki/> - Wolna encyklopedia dostępna w sieci Internet

<http://www.gzip.org/zlib/feldspar.html> - informacje nt. algorytmu deflate

<http://www.rasip.fer.hr/research/compress/algorithms/fund/lz/> - informacje o algorytmach LZ77, LZ78

<http://www.7zip.org> – oficjalna strona formatu 7-Zip

<http://www.computer-dictionary-online.org/LZH%20compression.htm?q=LZH%20compression> – krótką, aczkolwiek poglądową nt. algorytmu LZH

<http://michael.dipperstein.com/lzss/> - strona poświęcona aspektom algorytmu LZSS