

Politechnika Śląska w Gliwicach  
Wydział Automatyki, Elektroniki i Informatyki



# Programowanie Komputerów

Temat projektu:  
Algorytm LZ77

---

Autor	Pan Krystian Krzemień
Prowadzący	Dr inż. Adam Gudyś
Rok akademicki	2017/2018
Kierunek	Teleinformatyka
Rodzaj studiów	SSI
Semestr	3
Termin laboratorium	Wtorek, 8:45-11:45
Grupa	2
Termin oddania sprawozdania	25-01-2018
Data oddania sprawozdania	25-01-2018

---

# 1 Treść zadania

Zaimplementować algorytm kompresji LZ77, kompresji oraz dekompresji, parametry są przekazywane z linii poleceń, program wykorzystuje polimorfizm, kontenery i algorytmy STL, strumienie oraz wyjątki.

## 2 Analiza zadania

Algorytm kompresji LZ77 nie posiada stałego słownika, słownik jest ruchomy, o stałym, ale dowolnym rozmiarze. Rozmiar bufora słownikowego można regulować przełącznikiem.

### 2.1 Struktury danych

Klasa Algorithm jest klasą ogólną która zawiera nazwy plików i parametrów, oraz metody do otwarcia i zapisu plików. Klasy Compression oraz Decompression dziedziczą publicznie po klasie Algorithm. Metoda wirtualna LZ77 przyjmuje postać algorytmu kompresji w klasie Compression, a dekompresji w klasie Decompression.

Struktura Parametry przyjmuje parametry wywołania programu.

### 2.2 Algorytmy

Po przyjęciu i sprawdzeniu parametrów z linii poleceń program uruchamia konstruktor odpowiedniej klasy, Compression lub Decompression. W konstruktorze następuje otwarcie pliku wejściowego i odczytanie danych do stringa, wywołanie metody LZ77 oraz zapis do pliku wyjściowego.

Algorytm kompresji LZ77 opiera się na dopasowaniu jak najdłuższego ciągu bufora w słowniku, za to odpowiada metoda GetMatch która zwraca pozycję i długość pod ciągu. Wszystkie dane wyjściowe są zapisywane systematycznie do stringa wyjściowego, który na końcu programu zostaje zapisany do pliku.

Dekompresja algorytmem LZ77 jest o wiele szybsza od kompresji, dekompresja wymaga tych samych parametrów okna co zostały użyte przy kompresji.

Algorytm ten ewoluował na przestrzeni lat, jego udoskonaloną wersją jest np. LZ78

### 3 Specyfikacja zewnętrzna

Program jest uruchamiany z linii poleceń. Jako argumenty należy podać: nazwę pliku wejściowego, nazwę pliku wyjściowego, tryb oraz wielkość słownika. Lista przełączników:

- i -> po nim należy podać ścieżkę pliku wejściowego
- o -> po nim należy podać ścieżkę pliku wyjściowego
- k -> tryb kompresji
- d -> tryb dekompresji
- s -> po nim należy podać wielkość słownika

Przykładowe użycie:

```
Projekt.exe -k in.jpg -o out -k -s 15
```

Warunki użycia:

Podanie trybu oraz ścieżki do plików.

Przełączniki mogą być podane w dowolnej kolejności.

### 4 Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem programowania obiektowego.

#### 4.1 Typy zdefiniowane w programie

W programie zdefiniowano następujące klasy i struktury:

---

```
1 class Algorithm
```

---

Klasa Algorithm jest klasą ogólną która zawiera nazwy plików i parametrów, oraz metody do otwarcia i zapisu plików z których korzystają klasy Compression i Decompression. Algorithm zawiera funkcję czysto wirtualną LZ77.

---

```
1 class Compression : protected Algorithm
```

---

Klasa ta jest reprezentuje kompresję, wszystkie operacje na plikach są wywoływane w konstruktorze. Posiada metodę GetMatch operującą na stringu szukającą najlepszego dopasowania.

---

<sup>1</sup> **class** Decompression : **protected** Algorithm

---

Klasa ta jest reprezentuje dekompresję, wszystkie operację na plikach są wywoływane w konstruktorze.

## 6 Wnioski i uwagi

- Udało się zrealizować algorytm LZ77, realizacja jego jest nieoptymalna z czego sobie zdaję sprawę. Według mnie bazowanie na stringu jest główną przyczyną niskiej wydajności kompresji oraz niezastosowania użycia wielowątkowości.
- Największym problem realizacji było wybranie struktury danych na której została oparta mechanika działania.
- Zauważyłem, że pliki większe niż 500kB są kompresowane w czasie który jest daleki od zadawalającego. Pliki 10kB są kompresowane w około sekundę, plik 50 razy większy nie jest kompresowany 50 razy dłużej, czas kompresji nie jest wprost proporcjonalny do wielkości pliku.