1. **Samochody świata. CD Katalog**, Impresja Wydawnictwa Elektroniczne, od maja 1997 – 20 tys.
2. **Multimedialna encyklopedia powszechna PWN**, edycja 1996, grudzień 1996 – 19 tys., edycja 1998, grudzień 1997 – 3, 5 tys.
3. **Elektroniczny słownik języka polskiego PWN**, grudzień 1997 – 9 tys.
4. **Atlas 3D**, IPS C.G., 1996 – 8, 5 tys.
5. **Dyktando**, ZIP Soft, 1997 – 6, 5 tys.

Lista nie jest dziś w pełni aktualna. **Encyklopedia multimedialna** Oficyny Wydawniczej Fogra odnotowana na szóstej pozycji z 6 tys. egzemplarzy sprzedanych od listopada 1997 roku, ma już dwukrotnie większą sprzedaż, co plasowałoby ją na pozycji trzeciej. Doszło poza tym kilka nowości, m.in. świetnie sprzedające się **Dzieje PRL** (WSiP).

Włodzimierz Wypych

Cyfrowy karakter – krótka historia kodowania znaków

Dzisiaj do drukarń nie trafia żaden tekst, który wcześniej nie zostałby przetworzony elektronicznie. Tylko wspomnienia pozostały po zecerskiej kaszcie, z jej przemyślnym układem przegródek na różne czcionki. Zastąpił ją *charset*, czyli *character set*. Po polsku mówimy: „zestaw znaków", chociaż nie o znaki w ogólności tu chodzi, ale o znaki pisarskie. Ta różnica umknęła jakoś twórcom polskiej terminologii informatycznej. Nie dostrzegli, że dla oddania angielskiego *character* wystarczyło przywrócić polszczyźnie precyzyjne słowo: „karakter", które niegdyś już w niej funkcjonowało (pojawiło się np. w dziełku: „Nowy karakter polski..." Januszowskiego), ma ten sam grecki, przez łacinę zapośredniczony, źródłosłów co angielski *charakter*, i znaczy dokładnie to samo: znak pisarski właśnie – nie znak w ogóle.

Nie zamierzamy tu promować „karakteru" – na wprowadzenie go do obiegu jest już dziś chyba za późno. Pozostaniemy przy „ogólnie słusznym" znaku. Chcemy natomiast przedstawić pokrótce historię karakteru w epoce elektronicznej, ukazując problemy, jakie wiązały się z kodowaniem znaków, oraz przyjmowane rozstrzygnięcia, w tym standardy, które obowiązują w tej dziedzinie.

Kodowanie

Początek tej historii sięga telegrafu, który dzisiaj jawi się jako zwiastun zmierzchu epoki Gutenberga. W przekazie telegraficznym bowiem po raz pierwszy czcionkę zastąpił elektryczny impuls. Pierwszy raz pojawił się też problem kodowania. Harald C.M. Morse rozwiązał go w sposób naturalny, tworząc trójelementowy alfabet: kropka – kreska – pauza. Ten jednak okazał się niedogodny, gdy ludzi na obu końcach linii zastąpiły specjalne urządzenia do nadawania i odbioru komunikatów. Inżynier Anatol de Baudot, twórca dalekopisu, udoskonalił wynalazek Morse'a. Przyjął zapis binarny: sygnał – brak sygnału, i na nim oparł 5-bitowy system kodowania znaków, wykorzystywany z niewielkimi zmianami do dnia dzisiejszego.

Ciekawe, że zasady teletransmisji danych zostały opracowane, zanim pojawiły się maszyny mogące wykonywać operacje na danych, czyli komputery. Komputery nazywano początkowo „maszynami cyfrowymi". I nazwa ta, choć nieco przydługa, jest bardziej precyzyjna. Udobitnia bowiem, że znaki, którymi maszyny operują, są cyframi – znakami liczb. Wynika to z samej budowy urządzeń. Tworzone są z układów elektronicznych rozróżniających dwa stany fizyczne: sygnał/brak sygnału, i interpretujących te stany jako „0" lub „1". Każdą sekwencję zero-jedynkową o ustalonej długości (np. 8 bitów, oktet czyli → bajt) można zatem traktować jako cyfrę-zapis liczby w systemie dwójkowym, np. **10001010** to liczba 136.

Taka konwencja byłaby wystarczająca, gdyby maszyny cyfrowe miały służyć jedynie do obliczeń numerycznych, jako „kalkulatory" (lub „komputery" w wąskim znaczeniu tego słowa).

Problemy jednak, jakie stwarzało samo wprowadzanie programów i danych w nieczytelnej dla człowieka postaci cyfrowej, kazały pomyśleć o przyjęciu jakiegoś systemu kodowania liter i innych znaków, który ułatwiłby komunikację z maszyną.

To właśnie z potrzeb operatorów i programistów pierwszych maszyn cyfrowych powstały kody alfanumeryczne – obejmujące nie tylko cyfry, ale i litery. Translacja tekstów programów zapisanych w postaci alfanumerycznej na kod maszyny dała początek dziedzinie o nazwie *word processing*, czyli „przetwarzanie tekstów" (kolejna katastrofa terminologiczna polskiej informatyki), która jest jednym z korzeni elektronicznego edytorstwa.

Początkowo do translacji wykorzystywano warianty 5-bitowego kodu dalekopisowego, gdyż często właśnie dalekopisu używano do wprowadzania i wyprowadzania danych. Tworzono także nowe systemy kodowania, jak np. EBCDIC (*Exended Binary-Coded Decimal Interchange*), kod 8-bitowy stosowany w maszynach typu „main frame". Różnorodność systemów kodowania komplikowała transmisję danych między samymi systemami. Ten problem rozwiązano dopiero w 1977 roku, kiedy to ANSI (American National Standards Institute) zatwierdził kod ASCII – *The American Standard Code for Information Interchange*.

ASCII

To dobrze przemyślany 7-bitowy kod, definiujący 128 znaków (o wartościach kodowych od 0 do 127): zawiera litery (duże i małe), cyfry i znaki interpunkcji oraz różne znaki specjalne. ASCII szybko upowszechnił się, wraz z amerykańską technologią informatyczną, na całym świecie. Stał się *lingua franca* nowej epoki. Międzynarodowa Organizacja Standaryzacji – ISO, przyjęła ów fakt „do akceptującej wiadomości" i nadała amerykańskiemu systemowi kodowania status standardu międzynarodowego, jako ISO 646.

W praktycznym stosowaniu kodu ASCII występował jednak pewien poważny problem. Przyjęty repertuar znaków odpowiadał potrzebom Anglosasów („$", „#", „&"), ale nie wystarczał innym nacjom. Alfabety większości narodów europejskich (nie wspominając o reszcie świata) zawierają swoiste znaki. Polski alfabet nie jest pod tym względem szczególny (ma nawet mniej znaków swoistych niż np. alfabet francuski).

Rozszerzenie zestawu znaków ASCII okazało się dość proste. Wykorzystano ósmy bit, który, zgodnie z pierwotnymi założeniami, miał służyć do kontroli transmisji danych (tzw. bit parzystości), ale rzadko wykorzystywano go w tym celu. Do kodowania znaków można więc było przyjąć nie 7, a 8 bitów, czyli pełen bajt. Wystarczyło, by rozszerzyć repertuar o dodatkowe 128 znaków i uwzględnić nie tylko znaki swoiste kilku europejskich języków, ale też wiele symboli matematycznych i semigraficznych.

Amerykańscy producenci sprzętu komputerowego i oprogramowania zainteresowani lokalnymi rynkami w Europie szybko wykorzystali tę możliwość, tworząc „lokalizowane" warianty oparte na „rozszerzonym" ASCII. W koncernie IBM opracowano 256-elementowe zestawy znaków (*code pages* - strony kodowe), w których na pozycjach 128-255 dodano, dość zresztą mechanicznie, znaki różnych alfabetów. Strona Code Page 474 została przeznaczona dla Stanów Zjednoczonych, a Code Page 852 – dla krajów Europy wschodniej.

Strony kodowe rozprowadzano wraz z systemem DOS. W konsekwencji użytkownicy komputerów PC przez „ASCII" zwykle rozumieją „ASCII rozszerzone", czyli zestaw 256 znaków, nie pamiętając, że w istocie tylko 128 pierwszych z nich odpowiada tej normie.

ISO 8859

Twórczość producentów w zakresie kodowania próbowała powściągnąć ISO, podejmując własne prace nad 8-bitowym systemem kodowania. Zakończono je w 1987 roku. Wówczas to – w dziesiątą rocznicę ustalenia ASCII, ustanowiono standard międzynarodowy oznaczony symbolem ISO 8859. Norma opiera się na rozwiązaniu podobnym do przyjętego przez autorów *stron kodowych*. Definiuje 10 zestawów znaków (tzw. parts – części), różniących się „nadbudowanymi" nad ASCII znakami narodowymi i specjalnymi. Pierwsza część (o nazwie Latin-1) obejmuje języki zachodnioeuropejskie, część druga (Latin-2) – wschodnioeuropejskie (słowiańskie, albański, węgierski, rumuński), pozostałe obejmują znaki m.in. greckie, hebrajskie, arabskie, tureckie, esperanto i grażdankę.

Norma ta nie miała większego znaczenia dla amerykańskiego przemysłu informatycznego. W każdym razie – dla firm aktywnie działających w Europie wschodniej. W czerwcu 1990 roku na konferencji w Budapeszcie kilkanaście z nich, łącznie z Microsoftem, zadeklarowało wschodnim Europejczykom, że będą nadal stosować Code Page 852. W zestawie tym uwzględniono znaki kilku alfabetów Europy środkowej (m.in. polskiego, czeskiego, węgierskiego), ale pominięto alfabety również tu używane (niemiecki, francuski), jakby w przekonaniu, że społeczeństwa tej części świata wymieniają informacje głównie między sobą i np. Polacy potrzebują na co dzień raczej znaków czeskich i węgierskich niż niemieckich i francuskich.

Polskie litery

W drugiej połowie lat osiemdziesiątych, gdy komputery typu PC zaczęły upowszechniać się także w Polsce, sposób kodowania polskich znaków stał się dla ich użytkowników problemem zgoła zasadniczym. Jego znaczenia nie docenili tylko odpowiedzialni za krajową informatykę (zapewne z braku czasu – dobijano właśnie historycznego kontraktu z Bull'em).

Problem był tym bardziej poważny, że dotykał poziomu sprzętu. Ówczesne PC-ty były powszechnie wyposażone w monitory Hercules, które w trybie tekstowym zawierały generator znaków niemodyfikowalny na drodze programowej, gdyż umieszczony w ROM (czyli w pamięci stałej). Podobne generatory miała też większość drukarek. Zmiana znaków wymagała wymiany (przeprogramowania) „kostki". Irytację właścicieli PC-tów powiększała świadomość, że podobne kłopoty ominęły użytkowników prostszych, 8-bitowych komputerów, których system operacyjny (CP/M) był wraz z wzorami znaków ładowany z dyskietki.

Wobec braku stanowiska „czynników oficjalnych", sprawą kodowania polskich znaków zajęli się *nolens volens* sami bezpośrednio zainteresowani obywatele. Jednoczył ich sprzeciw wobec niepraktycznego rozwiązania CP 852, które, jak ujął to Mariusz Dec, „zamyka nas we wschodnioeuropejskim getcie". Poza tym jednak zdecydowanie się różnili. Z obywatelskiej inicjatywy powstało mnóstwo „patentów" na rozmieszczenie polskich znaków narodowych w górnej połowie rozszerzonego ASCII. Rozwiązania te, dziś już tylko historyczne, można znaleźć m.in. na Polskiej Stronie Ogonkowej.

Około 1990 roku szczególną popularność uzyskał tzw. kod Mazovii, wykorzystany w pierwszym polskim PC-cie opracowanym w Instytucie Maszyn Matematycznych. Miał on rzeczywiście kilka zalet: zachowywał znaki niemieckie i francuskie (bez jednej litery), omijał kody „gorące", mające specjalne funkcje w niektórych programach, a jednocześnie zachowywał sporo znaków semigrafiki wykorzystywanych do tworzenia interfejsu programów DOS-owych. Rozwiązanie to zaczęli respektować dystrybutorzy drukarek i kart graficznych sprzedawanych wówczas w kraju.

Urzędy odpowiedzialne za standaryzację w Polsce nie spieszyły się z zajęciem stanowiska. Dopiero w marcu 1991 Polski Komitet Normalizacji, Miar i Jakości opublikował normę **PN-91/T-42115** stanowiącą polską wersję normy ISO 8859-2 (jej części 2.): „Przetwarzanie informacji. Zestaw znaków graficznych w jednobajtowym kodzie 8-bitowym. Alfabet łaciński nr 2".

Przyjęcie tego rozwiązania wywołało zdecydowany sprzeciw wśród użytkowników PC. Normę uznano za „martwą", uchwaloną bez oglądania się na praktykę i „standardy faktyczne" (tj. kod Mazovii). Redakcja PCkuriera (12/91) wyrażała dość powszechną opinię:

1. *Uważamy, że fakt ustalenia normy nastąpił co najmniej o 5 lat za późno.*
2. *Uważamy, że ustalając normę tak późno, jedynym wyjściem było uznanie przez PKNMiJ albo obowiązującego de facto standardu Mazovii, albo uzgodnionego przez kilka wielkich zachodnich firm Code Page 852.*
3. *Wybór nie używanego przez nikogo standardu, zamiast porządkować sprawy, niepotrzebnie wprowadził dodatkowy zamęt.*
4. *Nowa norma najprawdopodobniej pozostanie jeszcze jednym dokumentem tylko na papierze, a szkoda.*

Norma została w praktyce zlekceważona i wydawało się, że nikt nigdy nie będzie jej respektował.

Sprawa kodowania polskich znaków straciła zresztą w tym czasie na ostrości. Rozpowszechniał się właśnie nowy typ kart graficznych i monitorów (VGA, SVGA), który umożliwiał zmianę zestawów znaków w sposób programowy. Jednocześnie zaczął w Polsce zdobywać popularność system Windows 3.1. z zupełnie odmiennym zestawem znaków. Microsoft bowiem, nie czując się skrępowany deklaracjami odnośnie do CP 852, przyjął niespodziewanie całkiem nowy zestaw (*East European Windows Character Set*), który do dziś funkcjonuje po nazwą *Windows CP 1250*. Wprowadzenie nowego rozwiązania uzasadniono krótko... koniecznością techniczną. Rosnąca liczba użytkowników Windows, dla których był on pierwszym środowiskiem pracy, przyjęła to wyjaśnienie do wiadomości ze spokojem. Niewielu wiedziało o składanych wcześniej przez Microsoft deklaracjach.

Norma ISO 8859, odrzucona przez pracujących w DOS, niepotrzebna tym, którzy przesiedli się na Windows, pozostałaby rzeczywiście normą „martwą", gdyby nie środowisko Unixa i... Internet, który ze środowiskiem tym jest genetycznie związany.

Internet

Normy ISO 8859 konsekwentnie przestrzegali producenci systemów unixowych. Jako międzynarodowy standard, była też respektowana przez autorów RFC (dokumentów definiujących normy internetowe), którzy problem kodowania znaków widzieli nie przez pryzmat „lokalizacji" produktów, ale w perspektywie „internacjonalizacji", czyli tworzenia technicznych podstaw wymiany informacji dla międzynarodowej społeczności.

ISO 8859 Latin-2 była standardem w systemach unixowych na polskich uczelniach. Podobnie jak w innych krajach, właśnie w środowiskach akademickich najwcześniej zaczęto korzystać z Internetu i pierwsze polskie strony WWW były kodowane zgodnie z tym standardem. Kiedy jednak krąg internautów zaczął się powiększać, liczebną przewagę wśród nich uzyskali użytkownicy Windows, dla których „naturalnym" sposobem kodowania polskich znaków był oczywiście CP 1250. Na listach dyskusyjnych w Internecie wręcz żądano uznania tego kodowania za normę. Pod „żądaniem mas" ugięli się nawet niektórzy polscy *providerzy* Internetu (np. Polska OnLine). W polskiej domenie internetowej zapanował bałagan. I utrzymuje się do dziś. Część stron WWW jest kodowana w ISO Latin-2, część w CP 1250. Minimaliści w ogóle rezygnują z polskich liter. Maksymaliści dają odwiedzającym ich strony możliwość wyboru standardu kodowania. To ostatnie rozwiązanie można by uznać za najbardziej rozsądne, gdyby... no właśnie – gdyby w ogóle było potrzebne. Dziś bowiem problem kodowania znaków na stronach WWW został rozwiązany. I to zupełnie inaczej niż w domorosłych patentach oraz za sprawą znacznie ogólniejszych kwestii niż problem „polskich znaków".

Polska Strona Ogonkowa

Przed laty kilku entuzjastów stosowania polskich liter w Internecie utworzyło na serwerze WWW krakowskiej Akademii Górniczo-Hutniczej „stronę ogonkową" poświęconą kwestii polskich znaków (**http://www.agh.edu.pl/ogonki**). Zebrano tam informacje, programy i fonty przydatne użytkownikom Internetu chcącym czytać i pisać po polsku.

Rozwiązanie nazywa się **MIME** – *Multipurpose Internet Mail Extension* (RFC 2045) i polega na tym, że dokument tekstowy (list, posting, dokument HTML), poza właściwą sobie zawartością, przekazuje informację, jaki zestaw znaków został w nim użyty. Dzięki temu program odbierający tekst (np. przeglądarka internetowa) może udostępnić go w zestawie oryginalnym, niezależnie od tego, z jakiego zestawu znaków odbiorca tekstu korzysta na co dzień i co ma jako „default" wpisane w ustawienia systemowe. MIME nie określa więc kodowania znaków, a jedynie definiuje sposób zapisu informacji o użytym w dokumencie zestawie znaków. Regulacja nie ogranicza się tylko do poczty (jak sugerowałaby nazwa) – jest dziś powszechnie respektowana również przez serwery WWW.

W połowie 1997 roku obaj główni producenci oprogramowania internetowego, firma Netscape i Microsoft, wypuścili na rynek nowe wersje swoich przeglądarek obsługujące MIME. Nie ma dziś żadnych merytorycznych powodów, aby również na polskich stronach WWW (i w poczcie) wykorzystywać inne kodowanie niż ISO 8859 Latin-2. Problem „polskich znaków" w Internecie powinien był więc zostać ostatecznie rozwiązany. No cóż... twórcy polskich stron są chyba odporni na „światowe nowinki". A może to przejaw dezorientacji wywołanej niekonsekwentną postawą Microsoftu, który wprawdzie przyjął standard ISO dla swojego oprogramowania internetowego, ale jednocze-

śnie utrzymuje CP 1250 w zakresie oprogramowania biurowego. Wielu użytkowników gubi się w tej dwoistości kodowania.

Co należy zrobić, aby polskie strony były po polsku:
1. W każdym dokumencie HTML należy wyraźnie zadeklarować zestaw znaków ISO 8859-2. Deklaracja ta powinna mieć postać:

```
<META HTTP-EQUIV="Content-type"
CONTENT="text/html; charset=ISO-8859-2">
```

i znajdować się w nagłówku dokumentu, między znacznikami <head>...</head>. Jej brak spowoduje, że przyjmowany będzie zestaw domyślny – zachodnioeuropejski ISO 8859-1, w którym nie ma polskich znaków.

2. Deklaracja powinna być prawdziwa: w przygotowanym dokumencie należy faktycznie zastosować zadeklarowane kodowanie. Samo dodanie deklaracji nie wystarczy, jeśli dokument będzie w innym niż zadeklarowy zestawie znaków.

3. Należy odpowiednio skonfigurować przeglądarkę, wybierając spośród opcji język (kodowanie) Latin-2. Użytkownicy straszych wersji Windows powinni sami zaopatrzyć się w fonty odpowiadające zestawowi znaków ISO 8858-2.

ISO 8859 jest jednym z fundamentów Internetu, jako sieci globalnej. Nie jest jednak uniwersalnym rozwiązaniem, gdyż nie obejmuje wszystkich znaków, którymi na co dzień posługują się użytkownicy Internetu w skali globu. Nie uwzględnia np. znaków ideograficznych, sylabarycznych i podobnych, stosowanych na Dalekim Wschodzie i w wielu innych częściach świata. Pomija także znaki wykorzystywane w różnych dziedzinach nauki i techniki.

A potrzeba stworzenia uniwersalnego systemu kodowania znaków była odczuwana, zanim jeszcze „eksplodował" Internet.

Unicode

W 1983 roku ISO podjęła ambitną próbę opracowania nowego standardu kodowania znaków (oznaczonego symbolem 10646), obejmującego znaki (pisownie) wszystkich używanych na świecie języków. Dla tak wielkiego repertuaru znaków konieczne było oczywiście przyjęcie większej podstawy kodowania niż tylko jeden bajt (8-bitów). Wydawało się, że kod 16-bitowy, dający możliwość kodowania 65 536 znaków, rozwiąże problem całkowicie (a nawet pozwoli uwzględnić egipskie hieroglify, by egiptolodzy mogli w swych rozprawach cytować złożone z nich inskrypcje).

W praktyce zadanie okazało się jednak bardzo trudne. Szczególny problem stwarzały ideogramy chińskie i znaki pochodne stosowane w Japonii, Korei i na Tajwanie (który zachował własną odmianę chińszczyzny). Otóż wiele znaków niewiele różni się między sobą, ale różnice te mają ogromne znaczenie dla poczucia tożsamości posługujących się nimi narodów. Propozycja, aby przyjąć wspólną, kanoniczną pisownię, została zdecydowanie odrzucona przez Japonię i Koreę. Zachowanie zaś pisowni każdego z tych języków okazało się niemożliwe, gdyż łączna liczba znaków przekraczałaby zakres 2-bajtowego kodowania. Rozwiązaniem byłoby przyjęcie kodowania 4-bajtowego, ale nie zgodziłaby się na to ta część ludzkości, która nie używa na co dzień ideogramów. Okazało się, że za narodowe partykularności większość krajów rozprezentowanych w ISO nie jest gotowa zapłacić więcej niż dwa bajty. Inicjatywa ISO znalazła się więc w impasie.

W nowy sposób kwestię uniwersalnego systemu 2-bajtowego kodowania, dla wszystkich *printable human letters*, podjęli w 1987 roku Joe Becker i Lee Collins z Xerox Palo Alto Research Center i Mark Davis z firmy Apple. W odróżnieniu od założeń ISO przyjęli rozsądną zasadę, że znak występujący w kilku pisowniach narodowych będzie miał ten sam kod.

Tak właśnie powstał Unicode. Termin ukuł Becker, jako zwięzłe określenie *„unique, universal, and uniform character encoding"*. Do inicjatywy wkrótce dołączyli przedstawiciele innych firm (m.in. Microsoft, Sun, Adobe, a następnie IBM i Novell). W styczniu 1991 powstało The Unicode Consortium, organizacja *non profit* stawiająca sobie za zadanie promocję tego systemu kodowania. W tym samym roku opublikowana została specyfikacja **Unicode 1.0**.

Poparcie, jakie Unicode zyskał wśród największych producentów oprogramowania, sprawiło, że stanowisko wobec inicjatywy musiała zająć ISO. Dość szybko znaleziono salomonowe rozwiązanie, odpowiadające zarówno aktualnym potrzebom w zakresie międzynarodowej (międzykulturowej) komunikacji, jak i możliwościom technicznym. Unicode został uznany po prostu za 2-bajtowy podsystem wcześniej proponowanego przez ISO 4-bajtowego standardu.

W 1992 roku ISO przyjęła nowy standard ISO/IEC 10646 oparty na 4-bajtowym systemie kodowania, nazwany „formą kanoniczną" (oznaczony sybolem UCS-4) i obejmujący tzw. Basic Multilingual Plane (BMP) – 2-bajtowy system (oznaczony jako UCS-2), identyczny z Unicodem w wersji 1.1.

System wielobajtowy ułatwia internacjonalizację oprogramowania i technologii informacyjnych, umożliwia przygotowanie tekstów wielojęzycznych (zawierających np. cytaty w innym języku). Oczywiście, przeciętni użytkownicy korzystający dotąd z „rozszerzonego" 8-bitowego ASCII mogą go uznawać za utrudnienie. 32-bitowy system kodowania znaków powoduje wzrost objętości danych i wydłużenie czasu transmisji. Znak „a" o kodzie 97, czyli „0061" w systemie szesnastkowym, ma w UCS-4 postać „0000 0000 0000 0061". Nie znaczy to jednak, że przy czterobajtowym systemie kodowania objętość zwykłego tekstu ASCII zwiększy się czterokrotnie. Jeśli bowiem trzy pierwsze (bardziej znaczące bity) są zerowe, wystarczy ten fakt w pewien sposób oznaczyć i zamiast 32 bitów wykorzystać jedynie 16. Taki „skompresowany" kod jest oszczędny i zarazem uniwersalny, choć do korzystania z niego potrzeba odpowiedniego oprogramowania.

Do normy ISO 10646 zostały dodane takie właśnie ekonomizowane formaty zapisu 32-bitowego, określane skrótem **UTF – UCS Transformation Format**: UTF-7, UTF-8, UTF-16, które oznaczają odpowiednio format 7-, 8- i 16-bitowy. UTF-8 np. zachowuje kody ASCII (0...127), kody natomiast od 128 w górę przekodowuje na ciąg bajtów z informacją o ich liczbie. Np. znak o nazwie „circled R registered sign" – mający kod 174 (0xAE szesnastkowo) w ISO 8859-1, a w USC-2 kod „0x00AE" – w formacie UTF-8 jest zapisywany na dwóch bajtach (16-bitach), jako „0xC2AE".

Standard ISO 10646 rzeczywiście rozwiązuje, i to w sposób radykalny, dotyczasowe problemy kodowania. Trzeba jednak będzie jeszcze nieco czasu, zanim się spopularyzuje. Jego pełna implementacja jest możliwa na poziomie systemów operacyjnych, a nie tylko na poziomie oprogramowania użytkowego. W każdym razie nowe wersje systemów operacyjnych, i te spod znaku Unixa, i Windows, będą – zgodnie z zapowiedziami producentów – uwzględniać nowy standard. Tu tkwi szansa na ostateczne rozwiązanie kwestii polskich znaków.

Tradycyjna kaszta drukarska zawierała czcionki znaków, które stanowiły „alfabet kulturowy" pewnej nacji. Teksty wielojęzyczne lub z symboliką naukową były niegdyś składane w wyspecjalizowanych drukarniach, w których tworzono zestawy czcionek dla określonej publikacji. Nie było to łatwe i nie dawało się szybko. Wprawdzie i dziś żadane studio DTP nie dysponuje zestawami fontów obejmującymi wszystkie znaki tabeli Unicodu, ale przygotowując wielojęzyczną publikację, nie trzeba już tworzyć ich od początku. Ktoś gdzieś na świecie już stworzył glify, potrzebne znaki zostały już przez kogoś opracowane graficznie. Aby je uzyskać, wystarczy podać kod znaków i określić cechy fontu. Jest to jeden z pożytków wieletnich wysiłków zmierzających do stworzenia uniwersalnego zestawu „karakterów" dla całej piśmiennej ludzkości.

Słownik

bajt (ang. *byte*) – jednostka danych (adresowalna w pamięci maszyny), składająca się z → bitów i wchodząca w skład większych jednostek, takich jak np. słowo (word). Termin wprowadził Werner Buchholz w 1956 podczas opracowywania architektury komputera Stretch (IBM). W 1962 roku określił bajt, jako: *a group of bits used to encode a character, or the number of bits transmitted in parallel to and from input-output units.*

Sekwencja bitów tworząca jeden bajt jest uzależniona od architektury systemu komputerowego. We współczesnych komputerach za bajt uznaje się 8 bitów (tzw. oktet). W dawniejszych systemach bajt mógł liczyć także 6, 7 lub 9 bitów. 8-bitową organizację pamięci przyjęto m.in. w IBM System/360, który, szeroko stosowany w latach sześćdziesiątych, upowszechnił rozumienie bajtu jako 8 bitów.

Symbol: **B** (w odróżnieniu od **b** oznaczającego bit).

Jednostki większe są oznaczane przedrostkami takimi jak w systemie metrycznym miar: kilo-, mega-, tetrabajt. W systemach cyfrowych bardziej poręcznym mnożnikiem jest liczba 1024, czyli dwa do potęgi dziesiątej, bliska metrycznemu tysiącowi. Dla podkreślenia różnicy symbole tych jednostek zapisywane są dużą literą.

Kilobajt (KB) to 2^{10}, czyli 1024 bajty, megabajt – 1024 kilobajty (ponad milion bajtów), tetrabajt – 1024 kilobajty (ponad miliard bajtów).

bit (od ang. *BInary digiT*) – najmniejsza jednostka informacji (ilość informacji, jaką daje wskazanie jednego z dwóch możliwych stanów) – urządzenie (pamięć), które może przyjmować jedynie dwa odróżnialne stany interpretowane jako: „0" i „1", „tak" i „nie", „prawda" i „fałsz". Za twórcę tego terminu uważany jest John Tukey, który zaczął go używać w 1949, jako skrótu bardziej poręcznego niż „bigit" (binary digit) i „binit" (binary unit).

ISO – nazwa Międzynarodowej Organizacji Standaryzacji, która opracowuje i ustanawia standardy międzynarodowe. Organizacja powołana w 1947 roku, zrzesza instytucje normalizacyjne z różnych krajów świata, ma siedzibę w Genewie. „ISO", wbrew pozorom, nie jest skrótem, ale nazwą własną, pochodzącą od greckiego słowa „iso" („równy"). W zakresie technologii informacyjnej ISO współpracuje z International Electrotechnical Commission (IEC). Wspólny komitet ISO/IES opracował mormę ISO 10646. Zob. http://www.iso.ch/.

kodowanie (ang. *coding*) – w ogólnym sensie: przyporządkowanie → *znakom* jednego zbioru znaków innego zbioru, w szczególności zaś: przyporządkowanie jednoznaczne, czyli funkcja w sensie algebraicznym, przypisująca każdemu z elementów zbioru kodowanego dokładnie jeden element innego zbioru nazywanego → kodem.

kod (ang. *cod*) – zbiór znaków, który reprezentuje (zastępuje) inne znaki (→ znak), wynik → kodowania.

Kodem podstawowym, narzuconym przez samą techniczną naturę urządzeń cyfrowych, jest kod dwójkowy, umożliwiający zapisanie informacji w postaci sekwencji → bitów o określonej długości, stanowiącej podstawę kodu. Znak zapisany w ten sposób może być interpretowany jako liczba całkowita zapisana w systemie dwójkowym.

Sekwencja 8 bitów (→ bajt) daje 256 kombinacji zerojedynkowych (od 0 do 255), którym można przyporządkować jednoznacznie elementy pewnego zbioru symboli, np. cyfr, liter itd. Dla stworzenia kodu wystarcza symbole te kolejno ponumerować. W praktyce przyporządkowanie określa się tak, aby spełniało pewne warunki techniczne. Praktyczniej jest np. kod o wartości 0 traktować jako „brak danych" niż kod oznaczający cyfrę „zero".

Czasem kod oznaczany jest parą liczb określających współrzędne tabeli, czyli tzw. pozycje tablicy kodowej.

Słowo „kod" używane jest też jako konkretna wartość (liczba), która odpowiada pewnemu → znakowi w danym systemie informacyjnym.

Zestawienie chronologiczne niektórych norm kodowania znaków

1977: **ANSI ASCII**, 7 bitów – 128 znaków

1983: IBM **Code Page**, 8 bitów, ASCII + 128 znaków

1987: **ISO-8859-x**, 8 bitów, ASCII + 128 znaków w 10 zestawach regionalnych

1985-1990: polskie firmy, m.in.: Mazovia, Cyfronet, CSK, DHN, Elwro-Junior, IEA-Świerk, 8 bitów, ASCII + 128 znaków

1991: **Windows CP-1250**, 8 bitów, tj. ASCII + 128 znaków

1991: **Unicode 1.0**, 16 bitów, 65536 znaków

1991: **Polska Norma PN-91/T-42115** tożsama z ISO-8859-2

1992: **ISO/IEC-10646**: **UCS-4**, **USC-2**, 32 bity, 16 bitów (tożsama z Unicode 1.1), 7, 8, 16 bitów.

zestaw znaków (ang. *character set*)
(1) Zbiór → *znaków*, jaki można wykorzystać w ramach danego systemu informacyjnego (systemu operacyjnego, oprogramowania) do organizowania, sterowania i reprezentowania informacji. Zestaw obejmuje zrówno → *znaki drukowalne*, jak i → *znaki sterujące*.
(2) Termin „zestaw znaków" używany jest na ogół w bardziej szczególnym znaczeniu, odpowiadającym ang. *coded charset* („kodowany zestaw znaków"), czyli jako → kod – przypisanie znakom wartości numerycznej.
Rozróżnianie obu powyższych znaczeń jest nader istotne. Ten sam zbiór znaków może być bowiem odmiennie kodowany (różnić się uporządkowaniem znaków). W konsekwencji, tym samym symbolom (identyfikowalnym graficznie) mogą odpowiadać różne wartości kodowe.

znak (grec. *charakter*, ang. *character*, w starej polszczyźnie – *karakter*)
(1) Podstawowy element pewnego alfabetu, pozwalający zapisać wyrażenia pewnego języka. Znaki mają reprezentację graficzną (grafemy) i także (choć niekoniecznie) dźwiękową (fonemy). Przykładami znaków są litery, cyfry, znaki interpunkcji. Ze znaków elementarnych można budować symbole złożone, np. dwuznaki (digrafy): „sz" w języku polskim, „sh" w angielskim.
(2) W informatyce znak to symbol, element kodu → *zestawu znaków*, który nie musi mieć widzialnej formy, ale spełnia pewną funkcję w procesie przetwarzania danych, umożliwia zapis danych lub operacji (programu). Dlatego rozróżnia się znaki drukowalne, znaki sterujące, znaki specjalne. Polska norma (PN-91/T-42115) definiuje znak jako „element zbioru służący do organizacji i sterowania lub przedstawiania danych".
(3) Znak jest pojęciem zasadniczym semiologii, dyscypliny naukowej wyemancypowanej z filozofii. Tak naprawdę znaki służą jedynie do oznaczania innych znaków, nie ma znaków, które by odsyłały do czegoś, co samo znakiem już nie jest. Wszelka działalność intelektu jest operacją na znakach, toczy się w kręgu interpretacji (rozumienia) i poza krąg znaczenia nie może wykroczyć ku „rzeczy samej w sobie" (szczegółowe uzasadnienie w „Krytyce czystego rozumu" Immanuela Kanta).

znak alfanumeryczny (ang. *alphanumeric character*) – znak będący literą lub cyfrą.

znak drukowalny (ang. *printable character*) – znak mający określoną reprezentację graficzną, możliwą do uzyskania na wydruku lub na monitorze.

znak specjalny (ang. *special character*) – znak inny niż alfanumeryczny, a więc znak interpunkcji lub inny symbol.

znak sterujący (ang. *control charakter*) – znak, który w danym systemie informatycznym jest wykorzystywany do sterowania procesem przetwarzania danych (lub transmisji) i z tego względu nie może być wykorzystany do zapisu danych. Taką funkcję pełni np. znak końca wiersza. W tabeli kodów ASCII przewidziano 32 znaki sterujące o kodach od 0 do 31. Spotykane czasem określenie „znak kontrolny" jest bezmyślną kalką z angielskiego, gdyż *control* znacza przede wszystkim sterowanie.

Nazwy znaków graficznych ASCII

ASCII obejmuje 128 znaków o kodach od 0 do 127. Znaki o kodach od 0 do 31 są znakami sterującymi, nie mają określonej reprezentacji graficznej. Mają natomiast swoje nazwy związane z rolą w systemach telekomunikacyjnych lub dalekopisowych przyjętych przez drukarki. Np. kod 10 (tzw. LF, *Line Feed*) oznacza wysuw papieru o jeden wiersz, kod 12 (FF, *Form Feed*) – wysunięcie papieru do początku nowej strony, kod 13 (CR, *Cariage Return*) – powrót karetki w drukarce lub maszynie do pisania. Kod 27 (*Escape*, dosłownie „ucieczka") pełni funkcję przełącznika, sygnalizuje początek sekwencji sterującej. Kody z tego zakresu są wykorzystywane w niektórych edytorach tekstu do oznaczenia formatu dokumentu. Koniec wiersza oznaczany jest przez LF (w systemach unixowych) lub za pomocą sekwencji CR LF (w DOS, Windows).

Nie wszystkie nazwy znaków są równie oczywiste, jak nazwy liter i cyfr, a niektóre przyjęte w polskiej normie (PN-93/T-42115) są co najmniej dyskusyjne. Niżej przedstawiamy wykaz wszystkich znaków spe-

cjalnych ASCII, nazw angielskich, przyjętych w polskiej normie ich odpowiedników, a w nawiasach – nasze objaśnienia lub komentarze.

32 *space*, spacja
33 ! *exclamation mark*, wykrzyknik
34 „ *double quote*, cudzysłów
35 # *hash*, znak numeru
36 $ *dollar*, znak dolara
37 % *percent*, procent
38 & *ampersand*, handlowe „i" (ligatura, czyli połączenie liter „e" i „t", tworzących łacińskie słówko „et" – „i")
39 ‘ *quote*, apostrof
40 (*open parenthesis*, lewy nawias okrągły
41) *close parenthesis*, prawy nawias okrągły
42 * *asterix*, gwiazdka
43 + *plus*
44 , *comma*, przecinek
45 - *minus*, łącznik, minus (łącznik nazywany jest dywizem)
46 . *full stop*, kropka
47 / *oblique stroke*, *slash*, kreska ułamkowa (ukośnik, skośnik)
48÷57 cyfry
58 : *colon*, dwukropek
59 ; *semicolonm*, średnik
60 < *less than*, mniejszy od (lewy nawias kątowy)
61 = *equals*, znak równości
62 > *greater than*, większe od (prawy nawias kątowy)
63 ? *question mark*, znak zapytania
64 @ *commercial at*, handlowe „przy", (norma PN-42118 zmieniła nazwę tego znaku na „handlowe po", stwierdzając, że „znak ten jest używany przy podawaniu ceny, np. *n* sztuk **po** *m* zł". Ciekawe, gdzie autorzy normy widzieli „jaja @ 2 zł". W istocie znak „@" jest skrótem angielskiego przyimka „at" (który w zwrotach tłumaczony jest jako „w", „na", „przy") wykorzystywanego nie tylko w kontekście handlowym czy komercyjnym. Dziś najczęściej spotykany w adresach poczty elektronicznej: kowalski@firma.com.pl. Żargonowo znak ten nazywamy „małpą", co nie jest zręczne, gdy adres trzeba podać słownie (kowalski małpa...). Ludzie kulturalni mówią po prostu „at" (wymawiając: „et").
65÷90 wielkie litery łacińskie
91 [*open square bracket*, lewy nawias kwadratowy
92 \ *backslash*, kreska ułamkowa odwrócona (jest to raczej opis znaku niż jego nazwa, powszechnie używa się nazwy angielskiej),
93] *close square bracket*, prawy nawias kwadratowy
94 ^ *caret*, znak akcentu cyrkumfleksowego
95 _ *underscore*, podkreślenie (wykorzystywany jako „twarda spacja")
96 ` *backquote*, znak akcentu słabego
97÷122 małe litery łacińskie
123 { *open curly bracket*, lewy nawias klamrowy
124 | *vertical bar*, kreska pionowa
125 } *close curly bracket*, prawy nawias klamrowy
126 ~ *tilde*, tylda

W prostych systemach edycyjnych (np. w systemie DOS) ograniczony repertuar znaków rozszerzano w ten sposób, że niektórym znakom przydawano kilka funkcji, np. cudzysłów (") jest tu używany jako znak cala. Najbardziej jednak obciążony jest znak „-", który może oznaczać: arytmetyczny minus, dywiz (łącznik w słowach złożonych), pauzę i myślnik (znaki rozróżniane w edytorstwie) i do tego jeszcze służy jako znak przeniesienia międzywierszowego, *hypen*).

Alfabet łaciński nr 2

(PN-91/T-42115 eqv ISO 8859-2: 1987)

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00			(sp)	0	@	P	`	p			(nbsp)	°	Ŕ	Đ	ř	đ
01			!	1	A	Q	a	q			Ą	ą	Á	Ń	á	ń
02			"	2	B	R	b	r			˘	˛	Â	Ň	â	ň
03			#	3	C	S	c	s			Ł	ł	Ă	Ó	ă	ó
04			$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
05			%	5	E	U	e	u			Ľ	ľ	Ĺ	Ő	ĺ	ő
06			&	6	F	V	f	v			Ś	ś	Ć	Ö	ć	ö
07			'	7	G	W	g	w			§	ˇ	Ç	×	ç	÷
08			(	8	H	X	h	x			¨	¸	Č	Ř	č	ř
09			)	9	I	Y	i	y			Š	š	É	Ů	é	ů
10			*	:	J	Z	j	z			Ş	ş	Ę	Ú	ę	ú
11			+	;	K	[	k	{			Ť	ť	Ë	Ű	ë	ű
12			,	<	L	\	l	|		Ź	ź	Ě	Ü	ě	ü
13			-	=	M	]	m	}			(shy)	˝	Í	Ý	í	ý
14			.	>	N	^	n	~			Ž	ž	Î	Ţ	î	ţ
15			/	?	O	_	o				Ż	ż	Ď	ß	ď	˙

Kod znaku określają współrzędne jego pozycji w tablicy (numer kolumny × 16 + numer wiersza), np. literze „A" dopowiada kod 65.
Tablica zawiera 256 pozycji. Znaki nie mające określonej normą postaci graficznej (niedrukowalne) pozostawiono puste.
Znaki na pozycjach 32, 160 i 173 zasadniczo nie mają reprezentacji graficznej. Jest to: spacja (*sp* – od ang. *space*), spacja nieprzenaszalna (*nbsp – no-break space*) i łącznik nietrwały (*shy – soft hyphen*, czyli znak przeniesienia).
W zakresie pozycji 0-127 (pierwsze 8 kolumn) tabela odpowiada standardowi ASCII, w zakresie 128-255 obejmuje zaś znaki alfabetów różnych nacji wschodniej Europy, w tym także znaki polskie.

Podstawowe formaty graficzne

Juliusz Donajski
Julo@ddg.art.pl

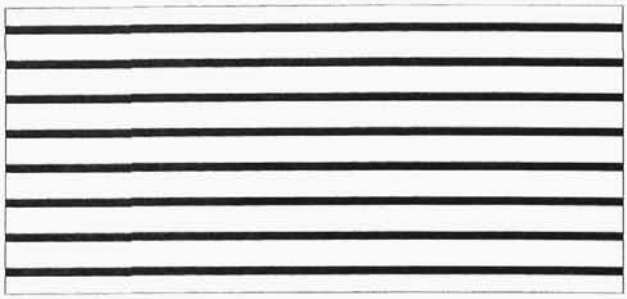
W cyfrowym świecie istnieje olbrzymia liczba formatów graficznych, które możemy wykorzystywać w zależności od potrzeb i przeznaczenia prac graficznych. Obok znajduje się krótki opis podstawowych formatów grafiki komputerowej oraz tabela pokazująca, jakie komputery i systemy operacyjne potrafią je obsługiwać. Podstawowe formaty graficzne wykorzystywane w prezentacjach stron World Wide Web to Graphic Interchange Format (GIF) i Joint Photographic Engineering Group (JPEG).

GIF – Graphic Interchange Format

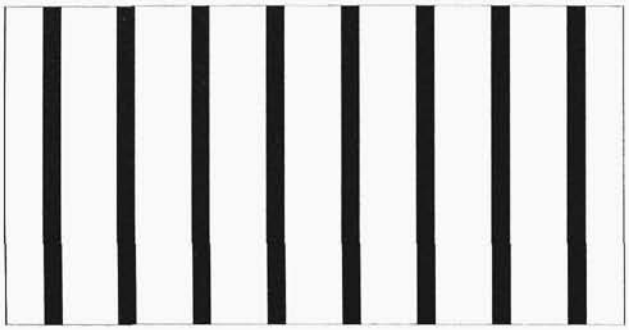
Jest to chyba najbardziej uniwersalny format. Obraz w nim zapisany może być czarno-biały, w odcieniach szarości lub kolorowy, natomiast maksymalna „ilość koloru", którą w tym formacie możemy zapisać wynosi 256 (kolorów lub odcieni szarości) w jednym pliku. Dużym plusem tego formatu jest możliwość tworzenia relatywnie małych plików, które świetnie nadają się do „transportu" w World Wide Web. Istnieje kilka odmian GIF. Najczęściej spotykane to: GIF87A oraz GIF89A. Pomimo iż dla obu odmian stosujemy tę samą nazwę, to jedynie GIF 89A ma właściwości, dzięki którym zyskał wielką popularność wśród twórców stron WWW. Są to: możliwość uczynienia jednego z kolorów przezroczystym (transparency) lub zapisywania i wyświetlania sekwencji obrazów, czyli animacji (Animated GIF). Format ten jest rozpoznawany i wyświetlany przez wszystkie przeglądarki WWW.

GIF – metoda kompresji

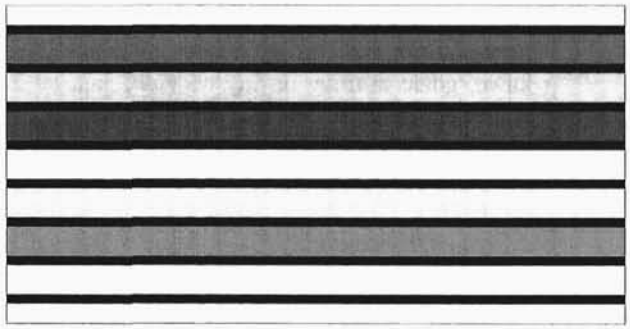
Format został opracowany w drugiej połowie lat osiemdziesiątych na zamówienie CompuServe z przeznaczeniem do „transportu" w sieciach komputerowych, w których funkcjonują różne platformy systemowe. Jak widać w tabeli obok, wszystkie platformy i systemy operacyjne potrafią rozpoznawać i zapisywać pliki w tym formacie. Stosowany tu schemat kompresji nazywany jest schematem (metodą) LZW. Nazwa pochodzi od pierwszych liter nazwisk twórców metody, czyli Lempel-Ziv i Welch. Właścicielem kodu źródłowego metody jest firma Unisys. Sama metoda należy do nie wywołujących strat w obrazie z powodu kompresji danych, a jej algorytm kompresji jest zbliżony do algorytmów stosowanych w zapisie plików BMP czy TIFF i zwanych RLE. Kompresja polega na zapisywaniu horyzontalnych zmian zachodzących w obrazie. Powiedzmy, że mamy do czynienia z plikiem składającym się z kilku jednokolorowych poziomych linii o szerokości 100 pixeli i wysokości jednego pixela każda. Zamiast zapisywać dane o każdym ze stu pixeli w linii, zapisuje się informację o całym 100-pixelowym bloku. Im mniej więc zmian kolorów mamy w obrazie, tym lepszą możemy osiągnąć kompresję pliku. Najlepiej widać to na przykładach obok.



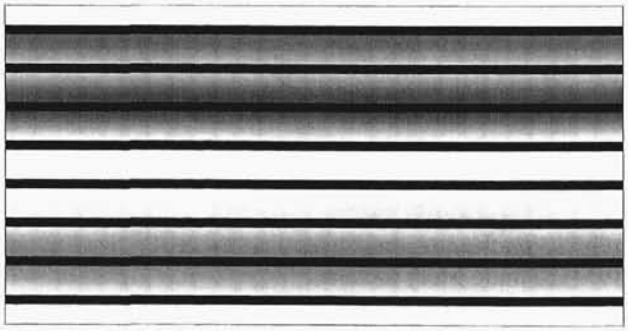
Rys. 1a: Przykład zasad kompresji danych w formacie GIF – 4.860 kB



Rys. 1b: Przykład zasad kompresji danych w formacie GIF – 8.395 kB



Rys. 1c: Przykład zasad kompresji danych w formacie GIF – 7.143 kB



Rys. 1d: Przykład zasad kompresji danych w formacie GIF – 11.016 kB

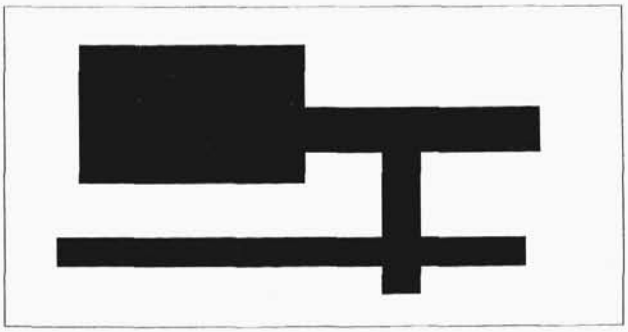
Widzimy, że ten sam dwukolorowy obraz z poziomymi liniami jest znaczenie mniejszym plikiem niż w wypadku linii pionowych. Podobnie ma się rzecz z liczbą kolorów: plik kilkukolorowy jest mniejszy od pliku zawierającego wszystkie 256 kolorów. Wyraźnie daje się dostrzec, że GIF znakomicie nadaje się do wykorzystania przy obrazach z blokami kolorów, typu loga, ikony, proste ilustracje oraz (uwaga!) czarno-białe fotografie. Nie znaczy to, że nie należy stosować tego formatu w wypadku bardziej skomplikowanych grafik lub fotografii kolorowych, trzeba jednak robić to ostrożnie i ze świadomością strat i zmian, które wówczas zachodzą przy konwersji 24-bitowego koloru do 8 bitów, bo tylko tyle w formacie można zapisać.

GIF przepleciony (Interlaced)

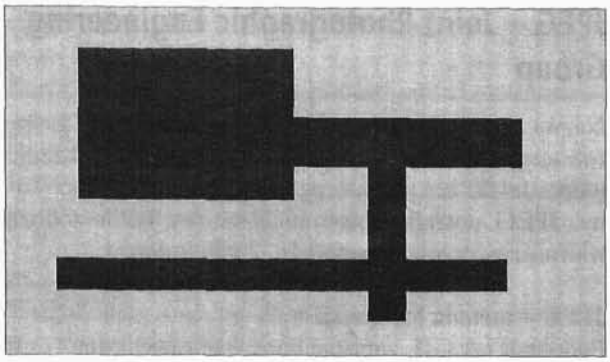
Jest to wykorzystanie jednej z cech formatu GIF89A, polegającej na wyświetlaniu uproszczonego podglądu całego pliku i następnie wyostrzaniu go do zapisanego maksimum. Jest to możliwe dzięki zastosowaniu opisanej wyżej metody kompresji polegającej na zapisywaniu informacji w poziomych jednopixelowych blokach. Kiedy przeglądarka wyświetla tego typu plik, najpierw otrzymuje informacje o jego szerokości i wysokości – w postaci jednopixelowych pasków, które może od razu wyświetlić. Paski te są wyświetlane w miarę odczytywania inforacji z pliku wg wzoru: 1, 6, 11,... następnie 2, 7, 12,... itd. Pliki wykorzystujące tę metodę są trochę większe od standardowych. W niektórych sytuacjach warto ją jednak zastosować. Należy przestrzegać zasady NIEŁĄCZENIA tej metody z przezroczystością w jednym pliku ponieważ może to prowadzić do unieczytelnienia obrazu (co czasami widać na stronach WWW).

GIF przezroczysty

Plik ten, otwarty w którymś z popularnych programów graficznych, wygląda jak „zwykły" GIF. W trakcie wyświetlania go przez przeglądarkę WWW widzimy natomiast, że zdefiniowany kolor jest przezroczysty. To kolejna cecha formatu GIF89A. Niestety GIF – w przeciwieństwie do takich formatów, jak TGA lub PICT – potrafi zapisać przezroczystość tylko jednobitową. Nadzieją dla poszukujących szerszych możliwości jest format PNG, w którym można zapisać 8 bitów przezroczystości. Obok widzimy przykład działania przezroczystości w formacie GIF.



Rys. 2a: GIF z wybranym obszarem przezroczystości



Rys. 2b: GIF bez wybranego obszaru przezroczystości

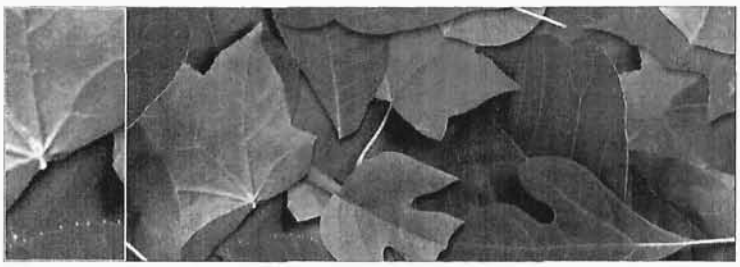
GIF animowany

Jest to kolejna możliwość wykorzystania cech formatu GIF89A. W pojedynczym pliku można zapisać sekwencję oddzielnych bloków mieszczących się w tych samych rozmiarach szerokości i wysokości. Sekwencja tych bloków-obrazów jest następnie wyświetlana przez przeglądarkę WWW w postaci animacji. Tworząc taki plik, możemy określić parametry typu: opóźnienie wyświetlania, prędkość wyświetlania kolejnych bloków-klatek, oraz wykorzystać opisywane wyżej: przezroczystość i interlacing. Możemy stworzyć animację wyświetlaną ciągle w pętli lub określić, ile razy ma być powtórzona. Obecnie wszystkie popularne przeglądarki WWW rozpoznają i poprawnie wyświetlają animowane GIF-y. Starsze przeglądarki – jeżeli jeszcze ktoś je wykorzystuje – ignorują sekwencje i wyświetlają pierwszy lub ostatni „kadr" animacji. Tworząc animacje tego typu, należy przestrzegać kilku reguł. Jedną z nich jest troska o prędkość transmisji danych: im większych bowiem rozmiarów (szerokość x wysokość) jest obraz, tym większy jest sam plik, im więcej klatek animacji – tym większa ilość danych i tym większa „waga" pliku. Jednocześnie wyświetlanie tego typu animacji pochłania dotatkowe moce procesora. W przypadku komputerów z silnymi procesorami i dobrymi kartami graficznymi użytkownik tego nie zauważa, jeśli jednak przyjrzymy się wyświetlaniu animowanego GIF-a na starszym komputerze, to od razu zauważymy, że animacja wyświetlana jest wolniej. Inną – często lekceważoną – regułą jest ograniczenie formatu GIF do zapisu jedynie 256 kolorów. Jeżeli dla wszystkich „klatek" zapisywanej sekwencji nie stworzymy wspólnej palety 256 kolorów, to może się okazać, że tworzymy animowany GIF z elementów zawierających ich łącznie więcej niż 256. Każdy kolor powyżej 256. jest wówczas maskowany do czerni lub „sprowadzany" do jednego z wcześniej zapisanych 256 kolorów. Efekty widzimy czasem na stronach WWW. Często też możemy obserwować skutki niewłaściwego ustawienia przezroczystości w animowanym GIF-ie. Błędy te najczęściej są konsekwencją przekroczenia 256 kolorów przy tworzeniu sekwencji obrazów.

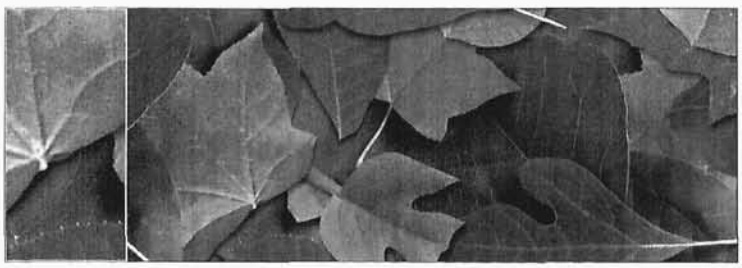
JPEG – Joint Photographic Engineering Group

Nazwa formatu powstała przez przyjęcie nazwy grupy roboczej opracowującej założenia standardu. Wskazuje jednocześnie cel, dla którego format został opracowany. JPEG potrafi zapisywać 24-bitowy kolor z dużą wiernością, przenosi ponad 16, 7 mln kolorów.

JPEG – metoda kompresji
Podobnie jak GIF, zapisuje dane o obrazie, kompresując je. Metoda kompresji jest jednak inna.

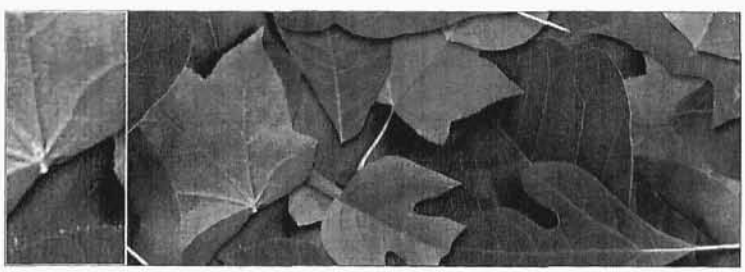


Rys. 3a: Przykład kompresji danych w formacie JPEG – niska kompresja



Rys. 3b: Przykład kompresji danych w formacie JPEG – średnia kompresja

Rys. 3c: Przykład kompresji danych w formacie JPEG – wysoka kompresja



Rys. 3d: Przykład kompresji danych w formacie JPEG – maksymalna kompresja

Kompresja stosowana w tym formacie jest bardzo efektywna (widać to na przykładach). W praktyce natomiast czas wyświetlania obrazów zapisanych w tym formacie jest trochę dłuższy niż w wypadku plików typu GIF, ponieważ metoda wymaga, przy każdorazowym wyświetlaniu – dekompresji danych. Generalnie rzecz biorąc, kompresja polega tu na poszukiwaniu obszarów o zbliżonych kolorach, uśrednianiu ich i „ujednolicaniu" powstałych w ten sposób obszarów. Im bliższą zeru kompresję stosujemy, tym więcej jest „różnych" obszarów i tym większy rozmiar pliku. W wypadku natomiast jednolitych kolorów metoda zastosowana w tym formacie sprawdza się słabo. Trzeba też pamiętać, że algorytm kompresji JPEG powoduje w obrazie straty, których nie daje się odwrócić. Podstawową różnicą – w stosunku do fromatu GIF – jest możliwość wyboru skali kompresji. Dzięki temu można znaleźć najlepszy punkt wypośrodkowujący pomiędzy potrzebą jak najwierniejszego odtworzenia detali i kolorów a wielkością pliku, czyli prędkością jego transmisji w WWW.

JPEG przepleciony
Stosunkowo nowym rozwiązaniem jest – zaproponowany przez firmę Netscape i przyjęty przez rynek – standard zapisu i wyświetlania tego formatu z przeplotem (podobnie jak to ma miejsce w formacie GIF). Pro-JPEG (Progressive JPEG File Format) oferuje znacznie większą skalę kompresji danych i większe możliwości wyboru zakresu kompresji. Tak zapisane pliki mają wszystkie cechy wspomniane wcześniej, przeglądarki natomiast wyświetlają je prawie identycznie jak pliki GIF z przeplotem.
Nawigując, lub, jak twierdzą niektórzy sceptycy – przedzierając się przez morze stron WWW, najczęściej spotykamy grafikę zapisaną w jednym z omawianych tu formatów: GIF lub JPEG. Coraz częściej jednak dowiadujemy się, że aby obejrzeć jakąś stronę zgodnie z intencją jej autora, musimy zainstalować w swoim komputerze odpowiednie rozszerzenia-wtyczki (plug-ins). Wprawdzie dodatki te nie doprowadziły do jakiejś rewolucji w publikacjach on-line, jednak stały się już trwałym elementem medium i w najbliższym czasie trzeba będzie przyjrzeć się im bliżej.

Słownik podstawowych rastrowych formatów graficznych

BMP – Windows Bitmap i OS/2 File Format
Format ten jest wykorzystywany głównie w systemach operacyjnych Windows i OS/2. Każdy plik BMP może być wykorzystany jako wallpaper w systemie Microsoft Windows. BMP może występować jako:
1-bitowa mapa – plik czarno-biały,
4-bitowa mapa – plik 16-kolorowy,
8-bitowa mapa – plik 256-kolorowy,
24-bitowa mapa – plik zawierający ponad 16, 7 mln kolorów

GIF – Graphics Interchange Format
Bardzo popularny i powszechnie stosowany format pliku graficznego rozpoznawany przez wszystkie chyba systemy operacyjne. Maksymalna głębia koloru to 8 bitów, czyli 256 kolorów.

HPCD – znany jako PhotoCD
Format ten zapisuje 4 różnej skali obrazy i daje możliwość ich wyświetlania. Głębia koloru to 24 lub 32 bity.

ILBM – Interleaved Bitmap File Format
Format powszechnie używany w komputerach Amiga. Istnieje wiele odmian tego formatu, m.in.:
- HAM – 12 bitów na pixel, maksymalnie 4 096 kolorów,
- HAM8 – 16 bitów na pixel, maksymalnie 256 000 kolorów,
- IFF24 – 24-bitowe rozszerzenie standardu IFF.

JPEG – Joint Photographic Engineering Group
Nazwa formatu przeniesiona z nazwy grupy roboczej dyskutującej założenia standardu. Format (faktycznie oznaczony skrótem „jfif") potrafi zapisać 24-bitową paletę kolorów i kompresować dane z różnymi poziomami strat jakości.

PCX – Zsoft Paintbrush File Format
Format opracowany na początku lat 80., gdy istniały jedynie karty graficzne CGA i Hercules. Później, aby sprostać wymogom nowszych monitorów – modyfikowany. Obecnie zapisuje 24-bitowe dane RGB – nie ma jednak żadnych mechanizmów kompresji.

PICT
Format opracowany na platformę Macintosh, jako format zapisu plików wektorowych. Obecne implementacje obsługują dane rastrowe i wektorowe, 24-bitową głębię koloru oraz różne poziomy kompresji.

PNG – Portable Network Graphics
Portable Network Graphics to jedna z ciekawszych propozycji, jakie w ostatnim czasie pojawiły się w wyniku poszukiwań metod przesyłania obrazów graficznych w Sieci.
Format GIF jest ograniczony do 256 kolorów. Mimo że w wielu wypadkach to wystarcza, jednak dalece nie zaspokaja wszystkich potrzeb. (Dodatkowo, gdzieś na początku 1995 roku Unisys i CompuServe ogłosiły nagle, że zamierzają wprowadzić opłaty licencyjne za wykorzystywanie – opatentowanej przez Unisys-a – metody kompresji LZW).
Drugi powszechny w Internecie format, czyli JPEG, wprowadza nieodwracalne straty koloru przy kompresji plików lub zachowuje bardzo duże rozmiary, zmuszając Sieć do „transportu" olbrzymiej ilości bitów.
Połączenie szybkości transferu plików z możliwością przesyłania prawdziwego koloru (true color) legło u podstaw tworzenia nowego formatu. Przyjrzyjmy się temu:
Lepsza kompresja niż w wypadku GIF-a, czyli mniej danych do transferu.
Kompresja nie powodująca żadnych strat.
Możliwość zapisu koloru w dowolnej palecie: od 1- do 8-bitowego zapisu, tak jak ma to miejsce w formacie GIF, poprzez 16-bitową skalę szarości, aż do 8- i 16-bitowego zapisu na każdy kanał, czyli do 24- i 48-bitowego zapisu prawdziwego koloru (true color).
Pełne pokrycie maskowania kolorów, czyli możliwość tworzenia do ośmiu przezroczystych (transparency) kolorów w jednym pliku, w miejsce dzisiejszego „włącz-wyłącz" jeden kolor w GIF-ie.
Korekcja gamma i kontrola jasności ponad platformami sprzętowymi.
Interlacing, czyli progresywne wyświetlanie obrazu.
Jest to format, który ma szansę stać się podstawowym formatem zapisu obrazu w publikacjach on-line.

PS – Postscript File Format
Format, który może zawierać informacje o obrazie rastrowym lub/i wektorowym.

XBM – X BitMap File Format
Natywny format systemu X Windows stosowany m.in. do zapisywania ikon wykorzystywanych w tych systemach. Format ten różni się zasadniczo od popularnych formatów graficznych typu BMP, GIF, JPEG itp., ponieważ pliki odczytywane są za pomocą kompilatora C, a nie wyświetlane przez typowy program graficzny. Dane plików XBM zapisywane są w nagłówkach (pliki.h). Format został stworzony przez X Consortium, jako część systemu X Windows.

XPM – PixelMap File Format
Format zbudowany na bazie kodów ASCII i biblioteki C, która definiuje przechowywanie danych o obrazie. XPM nie jest wprawdzie zatwierdzonym standardem X Consortium, ale w praktyce specyfikacje CDE dla X Windows mówią, że ikony ekranowe muszą być zapisane w formacie XBM lub XPM. Większość popularnych programów graficznych nie potrafi obsłużyć tego formatu bez bibliotek XPM.

RAS – Sun Raster file format
Format własny firmy SUN Corporation zapisujący dane w postaci nieskompresowanej. Głębia koloru to 24 bity.

TGA – Truevision Targa Image
Kolejny format zapisu mapy bitowej. Ma możliwość zapisu szerokiego spektrum map bitowych: od czarno-białych do RGB. Jest to jeden z popularniejszych formatów zapisu kolorowych obrazów w 24- i 32-bitowych plikach. Obsługuje szeroką gamę metod kompresji danych.

TIFF – Tagged Image File Format
Format ten został stworzony przez firmy Aldus i Microsoft. Właścicielem kodu źródłowego była firma Aldus. Po przejęciu przez Adobe Systems, prawo własności przeszło na tę drugą firmę.
Podstawowym założeniem było stworzenie formatu, który byłby zapisywany (skanowanie lub rysowanie) przez programy graficzne, ale dawałby się wprowadzać do innych programów przeznaczonych do publikacji drukowanych na papierze, a także dawałby się przenosić między platformami. Obecnie istnieje około 50 odmian tego formatu.
TIFF jest formatem mapy bitowej wykorzystującym m.in. sposoby kompresji RLE (Run-Length algorithm), LZW (niemal identyczny jak w formacie GIF) czy też algorytmy JPEG. Istnieje też możliwość zapisu pliku bez kompresji. Jest to format 24-bitowy, co oznacza, że zapisuje ponad 16, 7 mln kolorów.